The 19th International Conference
on Computer Graphics and Vision

GraphiCon'2009

October 5-9, 2009
Moscow, Russia

Conference Proceedings

19-я Международная Конференция
по Компьютерной Графике и Зрению

ГрафиКон'2009

5-9 октября, 2009
Москва, Россия

Труды Конференции

Московский государственный университет им. М.В. Ломоносова

Спонсорская поддержка ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009 – 2013 годы

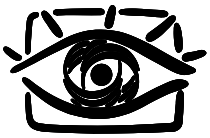
Спонсорская поддержка Российского фонда фундаментальных исследований

При партнерстве Intel corp.

Спонсорская поддержка Microsoft corp. (золотой спонсор)

Спонсорская поддержка NVIDIA corp. (серебряный спонсор)

Организовано Лабораторией компьютерной графики и мультимедиа ф-та ВМК МГУ



Организовано Факультетом вычислительной математики и кибернетики МГУ

При сотрудничестве с Eurographics Association

При сотрудничестве с Институтом прикладной математики им. М.В. Келдыша РАН

При поддержке Нижегородского фонда содействия образованию и исследованиям

Organizing Committee:

Co-Chairs: Evgeny Moiseev (Lomonosov Moscow State University)
Yuri Bayakovsky (Lomonosov Moscow State University)
Andrey Krylov (Lomonosov Moscow State University)

Conference Organizing Committee

Chair:	Alexey Ignatenko (Lomonosov Moscow State University)
Graphics and Design:	Ekaterina Chepurnova (Lomonosov Moscow State University)
Marketing and PR:	Maxim Fedyukov (Lomonosov Moscow State University)
Participants Support:	Victor Gaganov (Lomonosov Moscow State University)
Souvenirs and Conference Materials:	Andrey Ilyin (Lomonosov Moscow State University)
Conference Program:	Anton Konushin (Lomonosov Moscow State University)
Finances:	Vadim Konushin (Lomonosov Moscow State University)
Officials:	Andrey Krylov (Lomonosov Moscow State University)
Web site, Publishing and Printed Materials:	Andrey Nasonov (Lomonosov Moscow State University)
General Administration:	Maria Oseeva (Intel Corporation)
Equipment and Technical Administration:	Artem Semenov (Lomonosov Moscow State University)
Banquet and Coffee Breaks:	Ilya Tisevich (Lomonosov Moscow State University)

International Program Committee:

Co-Chairs: Yuri Boikov (The University of Western Ontario)
Victor Ostromoukhov (University of Montreal)
Anton Konushin (Lomonosov Moscow State University)

Members:

Valery Adzhiev (Great Britain)
Mark Bannatyne (USA)
Yuri Bayakovsky (Russia)
Alexander Belyaev (Great Britain)
Sergey Berezin (Russia)
Valery Bobkov (Russia)
Alexander Bondarev (Russia)
Ilya Brailovskiy (Russia)
Aleksey Chernavskiy (Russia)
Dmitry Chetverikov (Hungary)
Alexander Chipizhko (Russia)
Marc Daniel (France)
Victor Debelov (Russia)
Roman Durikovic (Japan)
Alexei Efros (USA)
Andre Gagalowicz (France)
Mikhail Iakobovski (Russia)
Andres Iglesias (Spain)
Alexey Ignatenko (Russia)
Denis Ivanov (Russia)
Stanislav Klimenko (Russia)
Vladimir Kolmogorov (Great Britain)
Andrey Krylov (Russia)
Victor Lempitsky (Great Britain)
Alexey Lukin (Russia)
Leonid Mestetskiy (Russia)
Mikhail Mikhailyuk (Russia)
Karol Myszkowski (Germany)
Pawan Kumar (USA)
Dimitri Plemenos (France)
Alexander Reshetov (Russia)
Michael Rychagov (Russia)
Vladimir Savchenko (Japan)
Alla Sheffer (Canada)
Sergey Sokolov (Russia)
Alexei Sourin (Singapore)
Daniel Thalmann (Switzerland)
Dmitriy Vatolin (Russia)
Dmitriy Vetrov (Russia)
Aleksey Voloboy (Russia)
Dmitriy Yurin (Russia)
Sergey Zheltov (Russia)
Denis Zorin (USA)

Address: GraphiCon'2009, Graphics & Media Lab, 7th floor, 701 room, Graphics & Media Lab VMiK (2-Gum corpus), Lomonosov Moscow State University, Leninskie Gory, Moscow, 119991, Russia
Tel: +7 (095) 939-0190 Fax: +7 (495) 939-0190 E-mail: gc2009@graphicon.ru
Web Site: http://gc2009.graphicon.ru

Contents

Section: Photos

Section: Image and video processing

Section: Various

Section: Визуализация

Section: Трехмерная реконструкция

Section: Биометрия

Section: Обработка геометрии

Section: Высокопроизводительная визуализация

Section: Распознавание и сегментация изображений

Section: Posters

Школа молодых ученых

Dear Participant,

We would like to welcome you at GraphiCon'2009, a major international conference on Computer Graphics, Computer Vision, Image and Video processing in Russia. Following well-established traditions, the 19th event will be hosted by Lomonosov Moscow State University on October 5-9, 2009.

This year we have a great program consisting of scientific papers, carefully selected by International Program Committee for oral and poster presentations, young scientist school and industrial exhibition. Keynote talks will be given by Prof. Craig Gotsman (Technion, Israel), Dr. Vladimir Kolmogorov (UCL, Britain), Dr. Alexey Efros (CMU, USA), Prof. Karol Myszkowski (MPI, Germany).

The International Program Committee was formed of 46 members representing 11 countries all over the world. Being top experts in the respective areas, all of them have done a tremendous job reviewing on average 7 papers out of 135 submitted. We express thanks to the committee members, who served at considerable personal sacrifice and with impressive collective wisdom. The final decision was based on at least two reviews of each manuscript and ended up with 53 works selected for oral presentation and 19 for poster presentation. Also we have decided to organize a young scientist school, where 21 papers will be presented.

Keeping traditions, this year GraphiCon has not only scientifical, but also extensive educational program consisting of 2 full-day tutorials and workshops for both undergraduate and PhD students, researchers and engineers of various companies that are interested in computer graphics topics. This year we have a number of tutorials from Intel, NVidia, Lomonosov Moscow State University and Keldysh Institute of Applied Mathematics RAS. We would like to thank lecturers for their willingness to share state-of-the-art knowledge in the respective fields.

We would also like to thank our main sponsors - Intel Corporation, NVidia and Microsoft, volunteers organized by CS MSU Graphics and Media Lab, and everyone who made this event happen.

We do hope you will enjoy the conference,

Anton Konushin, on behalf of GraphiCon'2009 Organizing Committee

Technical Section

GraphiCon'2009

October 5-9, 2009
Moscow, Russia

A priori Computation of a mesh size for Adaptive Loop Subdivision

Sandrine Lanquetin, Marc Neveu
LE2I, UMR CNRS 5158
Université de Bourgogne, BP 47870
21078 DIJON Cedex, France
{slanquet, mneveu}@u-bourgogne.fr

Abstract

Subdivision schemes provide efficient algorithms both for the design and processing of arbitrary control meshes. Theoretically, these techniques are often considered as an elegant algorithmic way to approximate a desired surface from a given surface. In practice, controlling the accuracy of control meshes with regard to the limit surface remains difficult. In this paper, from a bound of the distance between a subdivision surface and its control polyhedron, we *a priori* compute the subdivision depth according to a given accuracy. From this result, we can exactly predict the number of faces of the control polyhedron for a given accuracy during full subdivision. As full subdivision generates too many faces, we focus in this paper on adaptive subdivision. We work with the adaptive scheme which generates the fewest number of faces and explain how to predict a bound of the number of faces of the subdivided mesh because it is not possible to a priori give the final number of faces during adaptive subdivision.

Keywords: Subdivision surface, Loop scheme, polyhedral mesh, distance, adaptive subdivision, accuracy, mesh size.

1.INTRODUCTION

In computer graphics, subdivision surfaces are used to smooth meshes. These surfaces are defined by an initial coarse mesh named control polyhedron and refinement rules of subdivision. These rules describe how to move existing control points (old points) and how to create new points in order to reach the limit surface. The application of refinement rules generates a sequence of increasingly fine control meshes. The sequence of control meshes converges to a smooth surface: the limit surface. Among approximation schemes, the schemes of Loop [6] and Catmull-Clark [2] are the most popular. A particularity of approximation schemes is that control meshes approach the limit surface at each step of refinement (Figure 1). The maximum distance D at the initial level is always larger than the distance d at the next level and so on because the subdivision process 'smoothes' the mesh. In this paper, we focus on the Loop scheme.

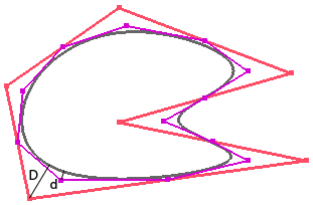


Figure 1: Distance in approximation schemes.

Each subdivision step provides a more accurate approximation of the limit surface. An upper-bound of this distance can be used as a stop criterion during the subdivision process [11][9]. However, the number of faces can be reduced by subdividing only particular faces. There are many criteria to determine faces to subdivide: faces in the viewport [3], angle between face normals [1], flatness of the surface [7], normal cones of vertices [8], curvature [12], distance [11][5]. We choose to use the distance between the control mesh and the limit surface as a geometric criterion for adaptive subdivision. This criterion, combined with local properties of subdivision, allows us to subdivide the surface only where this distance is greater than a given accuracy, saving a lot of triangles.

From a bound of the distance between a subdivision surface and its control polyhedron, the number of subdivisions to perform can be known *a priori* according to a wished accuracy. We can thus estimate the exact number of faces of the final mesh. During full subdivision, this is easy due to the definition of Loop.subdivision. Indeed, a face is split in four at each step of subdivision. When the mesh is partially subdivided, the number of faces cannot be predicted, we can only determinate an upper-bound.

The purpose of this paper is to a priori determine an upper bound of the number of triangles that will be generated during adaptive subdivision. It allows to estimate memory requirements and rendering rates, which is very usefull for applications in video games - mainly on portable playstations- or in virtual reality.

The paper is organized as follows. Section 2 reminds the reader with the necessary background that we use in our work: Loop subdivision and a bound for the distance of a subdivision surface to its control polyhedron. Section 3 explain the adaptive subdivision used with a distance criterion. In section 4, we determine the number of subdivisions to perform from an accuracy given at the initial level of subdivision and thus a bound for the number of faces of the mesh for adaptive subdivision and then in section 5, we give some practical results. Finally, directions for future work are proposed in the conclusion.

2.BACKGROUND

2.1. Loop subdivision surfaces

Subdivision surfaces are among the easiest ways to generate smooth surfaces. They preserve both the advantages of NURBS and those of polygonal meshes. We choose the Loop scheme for our results because the majority of meshes are currently triangular (triangular meshes provided by geometric modellers, triangulated meshes reconstructed from laser range images…). The Loop scheme generalizes quadratic triangular B-splines and the limit surface obtained is a quartic Box-spline. This scheme is based on splitting faces: each face of the control mesh at refinement level k is subdivided in four new triangular faces at level k + 1. This first step is illustrated into Figure 2. Consider a face: new vertices are inserted in the middle of each edge, they are named odd vertices and those of the initial face are named even vertices.

Figure 2: Left, an initial face. Right, the 4 new faces.

In the second step, all vertices are displaced by computing a weighted average of the vertex and its neighbourhood vertices. These averages can be substituted by applying different masks according to vertex properties: even, odd, valences (Figure 3). The sub-figure (a) represents the interior even vertex mask where n denotes the vertex valence and β is chosen to be [10]:

$$\beta = \begin{cases} \dfrac{3}{16} & if\ n = 3 \\ \dfrac{1}{n}\left(\dfrac{5}{8} - \left(\dfrac{3}{8} + \dfrac{1}{4}\cos\left(\dfrac{2\pi}{n}\right)\right)^2\right) & if\ n > 3 \end{cases}$$

The sub-figure (c) illustrates the interior odd vertex mask.

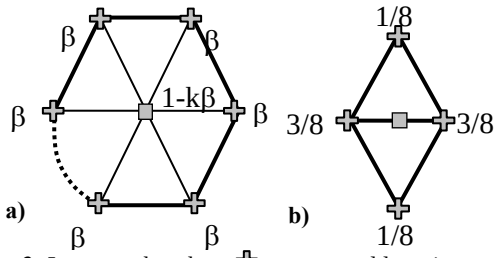


Figure 3: Loop masks where ✚ represent old vertices and ▫ the new position of an even vertex (a) and of an odd vertex (b) respectively.

2.2. Loop distance to limit surface

Loop surfaces are defined as the limit of an infinite refinement process. However, it is easy to compute exact points on the limit surface. Indeed, the images of a vertex (from the initial mesh) on the successive subdivision meshes depend only on this vertex and its neighbourhood. The following results are derived from [6].

Having the image P_0^∞ of the vertex P_0^0 as a function of P_0^0 and its neighbourhood allows us to compute the distance between P_0^0 and P_0^∞:

$$d\left(P_0^0, P_0^\infty\right) = \left\| \frac{-8.n.\beta}{8.n.\beta + 3} P_0^0 + \frac{8.\beta}{8.n.\beta + 3} \sum_{i=1}^{n} P_i^0 \right\|$$ with the valence of P_0^0 and β the Loop coefficient.

From the distance between the control mesh vertices and their image on the limit surface allows us to determine the maximum distance between the control mesh and the limit surface. Indeed, subdivision surface properties are such that the control mesh vertices are the most distant points from the limit surface because refining coefficients are all positive. If a vertex is inserted between two vertices of the control mesh, the distance between this vertex and the limit surface will inevitably be smaller (Figure 4). This property is valid for almost any surface: convex or concave, open or closed. This bound is suitable for all usual mesh, however an exotic mesh can be generated to contradict it. This is shown in [9].

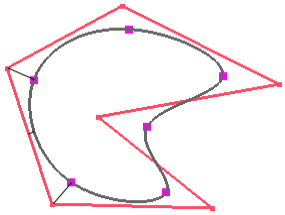


Figure 4: Distance from a midpoint to the limit surface.

Notations used for the vertices $P_{i,j}^k$ of the control mesh $\mathbf{P}^k$ at subdivision level k are described in Figure 5.

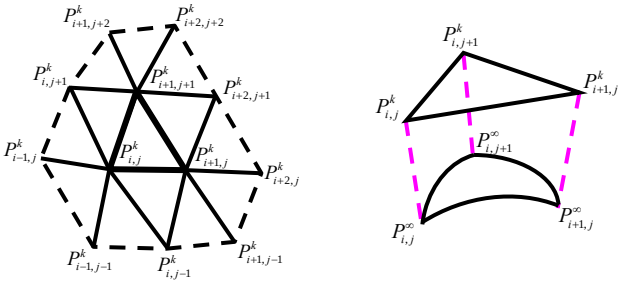


Figure 5: Notations used for the vertices of the control mesh. Left: in bold, Loop triangle at level k with its neighbourhood. Right: Loop triangle at level k with corresponding limit surface triangular patch.

In [6], the following theorem is proven:

Theorem 1. Let $S^k(u,v)$ be the polyhedral surface associated to the control mesh $\mathbf{P}^k$ at subdivision level k and $S^\infty(u,v)$ be the limit surface.

$$\left\| S^k(u,v) - S^\infty(u,v) \right\| \le M_k \le \left(\frac{5 - 8N\beta_N}{8}\right)^k M_0$$ where

$$M_k = \max_{P_{i,j}^k \in \mathbf{P}^k} \left(\left\| \frac{8n_{i,j}^k \beta_{i,j}^k}{8n_{i,j}^k \beta_{i,j}^k + 3} P_{i,j}^k - \frac{8\beta_{i,j}^k}{8n_{i,j}^k \beta_{i,j}^k + 3} \sum_{\substack{P_{i',j'}^k \\ adjacent\ to\ P_{i,j}^k}} P_{i',j'}^k \right\| \right),$$

$n_{i,j}^k$ is the valence of the vertex $P_{i,j}^k$, $\beta_{i,j}^k$ is the Loop coefficient associated to this valence and N is the maximum valence of the successive control meshes.

3.ADAPTIVE SUBDIVISION

The adaptive subdivision scheme used in this paper is described in this section. Then the chosen scheme is analyzed and compared to other schemes.

3.1. Adaptive scheme

When the same rules are applied on the whole input mesh, the number of faces quickly increases. Indeed, for Loop scheme, a face produces four faces after one subdivision, 4^2 after 2

subdivisions and 4^n after n subdivisions. A well-known method to reduce the number of faces is to subdivide only a part of the mesh.

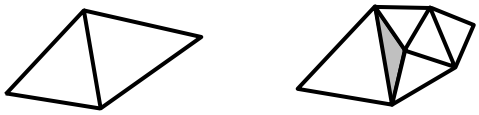


Figure 6: Left: two adjacent faces of the mesh. Right: one face is subdivided and the other is not, the crack between the faces is represented in grey.

When the surface is not entirely subdivided, cracks appear between faces with different subdivision depth as shown in Figure 6.

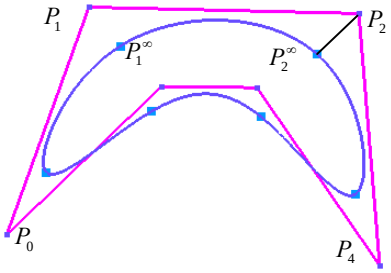


Figure 7: Upper-bound of the distance between a control vertex and the limit surface.

The area to subdivide can be selected by different ways. We choose to select it according to the accuracy of the control polyhedron from the limit surface. The adaptive subdivision used here generates a minimum number of faces. The chosen criterion selects faces according to the distance of its vertices to the limit surface. This distance cannot easily be computed, so the distance between a vertex and its image on the limit surface is used as an upper-bound (Figure 7).

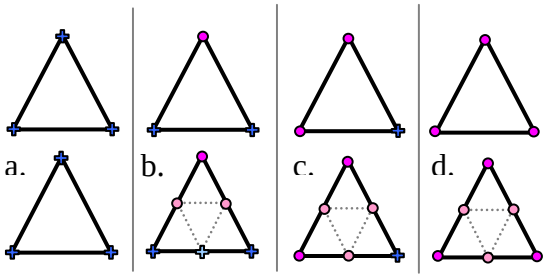


Figure 8. Different cases of face subdivision during adaptive subdivision.

In this paper, we choose to work with the first adaptive subdivision introduced in [4]. Vertices are classified into two categories: static and mobile vertices. A mobile vertex is a vertex which is displaced because the distance between this vertex and the limit surface is more than a given accuracy. Other vertices are static. The topological rules used to avoid cracks in this scheme are chosen as follows. Faces are classified into 4 categories according to the number of mobile vertices. Mobile vertices are depicted by circles in Figure 8 (top). When all vertices are static, the face is not subdivided Figure 8.a.). Figure 8.b. illustrates the case where only one vertex is mobile; only two among three new (odd) vertices are then mobile in order to avoid cracks. When there are two mobile vertices, face subdivision is almost normal except the fact that one among the old vertices is static (Figure 8.c.). Finally when all vertices are mobile, subdivision is carried out in a normal way (Figure 8.d.).

By applying this adaptive subdivision scheme, cracks are avoided because the inserted vertex is in the middle of the edge from the face to subdivide. Indeed, the middle vertex inserted on the edge between a face that is subdivided and a face that is not subdivided is not mobile any more. This scheme does not allow a correct computation of a neighborhood for all vertices for further subdivisions. Indeed faces are generated but correspondences between edges are not updated. Nevertheless, a neighborhood is not necessary because the distance for a static vertex to the limit surface does not change. We just need to save it from one level to another. For the other cases, the neighborhood is correct; the distance can thus be computed. Figure 9 illustrates a case where the new neighborhood should include both new and old vertices.

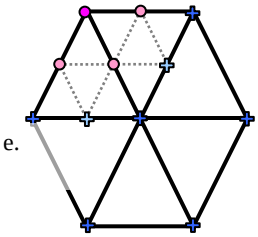


Figure 9: New neighbourhood.

3.2. Results and comparison

It is not easy to compare adaptive subdivision schemes because criteria are very different according to the required effect. For instance, comparing a view dependent scheme with ours, is not very interesting because they do not have the same goal at all.

One focuses in the part of the model which is in the view cone and the other on the accuracy of the entire model. Other criteria, such as those based on curvature or normal cone, are more similar to the above criterion of distance because they are dependent as shown in Figure 10. Indeed, for a vertex of the control mesh, the larger the angle of the normal cone the larger the distance between this control vertex and the limit surface. The distance is represented by a red line, the limit surface is represented by a blue shape and the normal cone is in grey color in Figure 10.

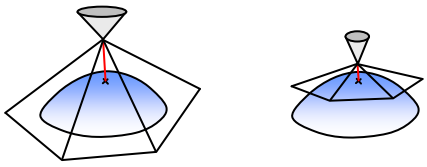


Figure 10: Relation between the size of the normal cone at a control vertex and the distance between this vertex and the limit surface.

Curvature, normal cone and distance are similar even in special cases such as that presented in Figure 11. Indeed, the distance between P_2 and the limit surface is negligible but it is the same for the curvature and the normal cone. On the other hand, the distance from P_1 or P_3 to the limit surface is large as the curvature or the normal cone.

For every adaptive scheme, topological rules are modified to avoid cracks. This modification involves that the limit surface is

different from this obtained in subdividing the whole mesh. This is not important because subdivision is stopped according to the criterion, so the limit surface is never reached.

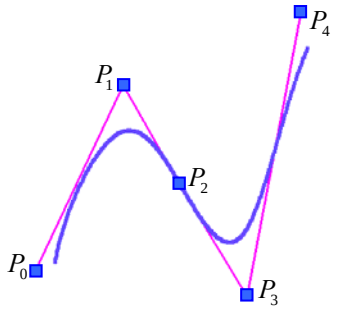


Figure 11: Similarity between distance curvature and normal cone.

The main advantage of our adaptive subdivision scheme is that the criterion is handier for the user. Moreover, for fields like computer aided geometric design, it is very significant to know the accuracy of the model. The results obtained with this distance criterion gives results similar to those obtained with other criterion such as curvature, normal cone but it is more intuitive and thus easier to use. Contrary to others adaptive scheme, the scheme introduced in this section do not generate a "good" mesh according to the mesh definition. Indeed, between subdivided and not subdivided faces, the mesh topology is particular. The odd vertex introduce and not displaced to avoid cracks (in grey in Figure 12 left) do not belong to the left face otherwise the face would not be more triangular. Thus, the mesh topology is represented in Figure 12 (right). The light grey line represents the edge of the left face which is in one part and the dark grey lines represent the edges of the right faces adjacent to the left face.

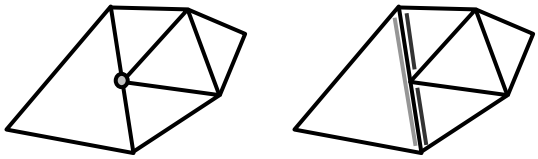


Figure 12: The generated mesh is not "good".

This kind of topology is sufficient to visualize mesh but it would be not sufficient in some particular cases. For example, in the case where the user modifies the tag of the grey vertex in Figure 12 (mobile instead of static) because he decides to change the needed accuracy or selects faces to subdivide with an other criterion. Computing intersection between two meshes can also be a problem because the data structure is not updated.

Figure 13: From left to right: respectively full subdivision, adaptive subdivision with an accuracy of 0.1, adaptive subdivision with an accuracy of 0.05.

In Figure 13, one can see on the one hand the results obtained for a given accuracy (here 0.1) and that the number of faces of the subdivided model in a non adaptive way corresponds to the number of faces which one can obtain in adaptive subdivision but for a higher accuracy (here 0.05). If the number of faces is smaller for the adaptive subdivision with a precision of 0.05, one can however note that time necessary to obtain the subdivided mesh is longer than adaptive subdivision with an accuracy of 0.1 because there is an additional level of subdivision. Table 1 gives numerical results corresponding to Figure 13 where a time value of 1 refers to the time necessary for a full subdivision.

	Plain $\varepsilon = 0.1$	Adaptive $\varepsilon = 0.1$	Adaptive $\varepsilon = 0.05$
Level of subdivision	3	3	4
Number of faces	9472	3841	7930
Minimum distance ($.10^{-4}$)	4,16	2614	15.35
Maximum distance ($.10^{-4}$)	831.95	831.95	256.01
Mean distance ($.10^{-4}$)	136.58	333.55	165.40
Time (in ratio)	1	0.1	0.33

Table 1. Comparison of the results obtained on the bunny model.

4.SIZE OF MESHES

To manage the storage of the control mesh, we need to know a priori the size of this mesh. We have already seen in a previous section that the number of faces is different according to the chosen subdivision of the mesh. Indeed, adaptive subdivision generates fewer faces than full subdivision.

4.1. Full subdivision

From [6], we have:

Corollary 1. With M_0 and N as in Theorem 1, the control mesh approximates the limit surface with the accuracy ε if the number of recursive subdivision is at least k with $k > -\ln\left(\frac{M_0}{\varepsilon}\right) \Big/ \ln\left(\frac{5}{8} - N\beta_N\right)$ where $N = \max(6, N^0)$ and N^0 is the maximum valence of the initial control mesh.

Note: The number 6 comes from the subdivision. Indeed, during the subdivision process, the valence of each inserted vertex is 6. This is develop later in the paper.

When the subdivision of the control mesh is full, it is easy to determine the exact number of faces of this mesh at a given accuracy. Indeed, Corollary 1 gives the subdivision level according to a given accuracy. From this computation, the mesh has to be subdivided k times. However, in the full subdivision, we know that a face generates 4 sub-faces at the first subdivision level, 4^2 at the second subdivision level and 4^k at the k^{th} subdivision level. The control mesh at the k^{th} subdivision level will thus have $nf_k = 4^k \times nf_0$ faces with nf_0 the number of faces of the initial control mesh.

4.2. Adaptive subdivision

When the subdivision of the control mesh is adaptive, it will be interesting to *a priori* know the size of the control mesh at a given

accuracy. Only faces the vertices of which have a distance to the limit surface greater than the given accuracy are subdivided. Thus, we cannot *a priori* exactly determine the number of faces of the control mesh for this accuracy. Nevertheless, we can upper-bound the number of faces of the control mesh without subdividing the initial control mesh.

Theorem 2. A vertex P_i^0 of valence n_{v_i} approximates the limit surface with the accuracy ε if the number of recursive subdivision is at least k_{v_i} with $k_{v_i} > -\ln\left(\frac{\left\|P_i^0 - P_i^\infty\right\|}{\varepsilon}\right) \Big/ \ln\left(\frac{5}{8} - n_i\beta_{n_i}\right)$ where $n_i = \max(6, n_{v_i})$.

Proof. For each vertex, we have:

$$\left\|P_i^k - P_i^\infty\right\| \le \left(\frac{5 - 8n_i\beta_{n_i}}{8}\right)^{k_{v_i}} \left\|P_i^0 - P_i^\infty\right\| < \varepsilon$$

The last part of the inequality can be written: $\left(\frac{5 - 8n_i\beta_{n_i}}{8}\right)^{k_{v_i}} \left\|P_i^0 - P_i^\infty\right\| < \varepsilon$

$$\Leftrightarrow k_{v_i} > -\ln\left(\frac{\left\|P_i^0 - P_i^\infty\right\|}{\varepsilon}\right) \Big/ \ln\left(1 \Big/ \frac{5}{8} - n_i\beta_{n_i}\right)$$

Corollary 2. A face f_i approximates the limit surface with the accuracy ε if the number of recursive subdivision is at least k_i with $k_i = \max_{j\in[1,2,3]}\left(k_{v_j}\right)$ where k_{v_j} are computed for each vertex P_j^0 of f_i.

Corollary 3. The total number of faces after adaptive subdivision is upper-bounded by $|F| = \sum_{i=1}^{k_i} 4^i \times |f_i|$ where $|f_i|$ is the number of faces that needs to be subdivided k_i times.

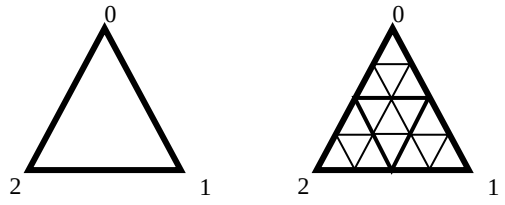


Figure 14: Left: Subdivision levels of face vertices. Right: Number of new sub-faces.

From the computation of the subdivision level k_{v_j} of each vertex of a face f_i, the subdivision level k_i of the face can be determined. We choose the subdivision level of the face to be the maximum of the subdivision level of its vertices. For instance, if the subdivision levels of the face vertices are respectively 0, 1, 2, the subdivision level of the face will be 2. In Figure 14 (on the left), subdivision levels of the vertices are represented by coefficients near the vertices.

As we wish an a priori upper bound of the mesh, the number of sub-faces created depends only on the initial subdivision level. Such as in the full subdivision, a face generates 4 sub-faces at the first subdivision level, 4^2 at the second subdivision level and 4^k at the k^{th} subdivision level. The generated sub-faces for the previous example are shown in Figure 14 (on the right). If we do not pay attention to special cases, we can have the following problem. Let the box in Figure be the initial control mesh. The subdivision level found for each initial vertex with a given accuracy $\varepsilon = 0.2$ is 2; however by successive subdivisions the mesh reaches the wished accuracy only after 3 subdivisions (Figure 15).

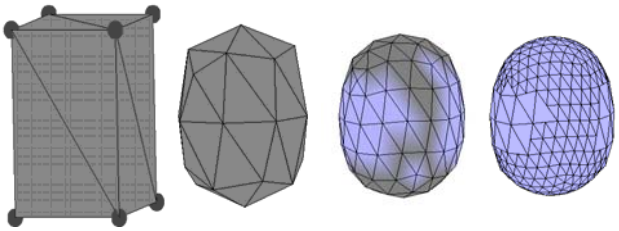


Figure 15: Subdivision depth: 2 subdivisions for each vertex and 3 subdivisions by successive refinement.

From Corollary 1, the computation of k_{v_i} uses $n_i = \max(6, n_{v_i})$. The importance of this maximum is shown in Figure 15 (a to d). Let the box in Figure 15.a. be the initial control mesh. The subdivision level found for each initial vertex with a given accuracy $\varepsilon = 0.2$ is 2; however by successive subdivisions the mesh reaches the wished accuracy only after 3 subdivisions.

The problem encountered is illustrated in Figure 16. In fact, some new vertices need one subdivision more than for the initial vertices. This problem comes from the valences of the new vertices which are always 6 whereas the initial vertices valences are 4 or 5. It can be avoided by taking 6 for the valence at the initial level when the computed valence is less than 6. For such vertices, the computation will thus be done with valence 6 and the corresponding Loop coefficient ($\beta = 1/16$).

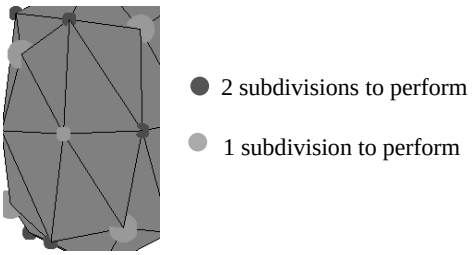


Figure 16. Focus on the problem: higher valence (6) of vertices after subdivision yields to additional subdivision

5.PRACTICAL RESULTS

In this section, results obtained on the penguin model with an accuracy $\varepsilon = 0.2$ are presented.

5.1. Full subdivision

For the full subdivision, the mesh has to be subdivided three times according to section 4.1. As the control mesh includes 500 faces at the initial level, the control mesh will have 32 000 faces at the third subdivision. Figure 17.b. illustrates the control mesh of the penguin model after full subdivision and Graph 1 represents the number of faces which have to be subdivided once, twice or more times according to the result presented in section 4.2.

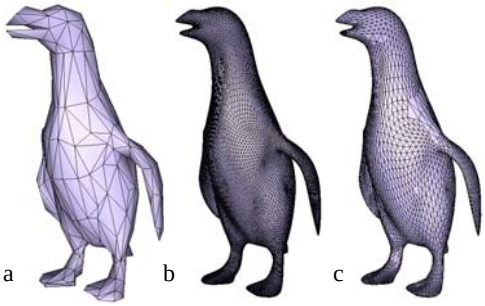
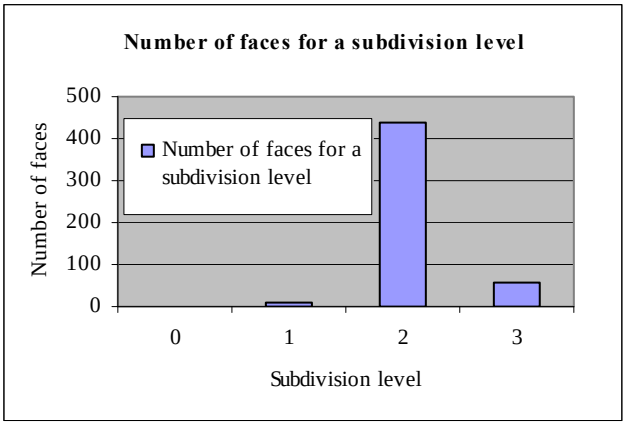


Figure 17: a. the penguin model. b. full subdivision with an accuracy $\varepsilon = 0.2$. c. adaptative subdivision with an accuracy $\varepsilon = 0.2$



Graph 1. Number of faces for successive subdivision levels in applying the formula on the penguin model with $\varepsilon = 0.2$.

5.2. Adaptive subdivision

For adaptive subdivision, we first compute the number of subdivision to perform for each vertex of the penguin mesh. The subdivision level of each face is then the maximum subdivision level of its vertices. Graph 1 shows the number of faces for each level of subdivision to perform. Subdividing each face the corresponding number of times, we can upper bound the real number of faces of the final adaptive mesh. In this case, by successive subdivisions, the adaptive mesh counts 8213 faces and our forecast is $10544 = \sum_{i=1}^{3} 4^i \left| f_i \right|$. Figure 17.c. represents the control mesh of the penguin model after adaptive subdivision. We cannot improve this upper-bound without subdividing the mesh.

5.3. Comparison on different meshes and accuracy

In this section, the number of faces are computed and estimated on various meshes and accuracies such as the penguin model with $\varepsilon = 0.1$ and $\varepsilon = 0.2$ (Figure 17), the cat model with $\varepsilon = 0.1$ and $\varepsilon = 0.2$ (Figure 18), the torus model with $\varepsilon = 0.1$ (Figure 20), the box model with $\varepsilon = 0.05$ (Figure 21) and the top model with $\varepsilon = 0.15$ (Figure 22).

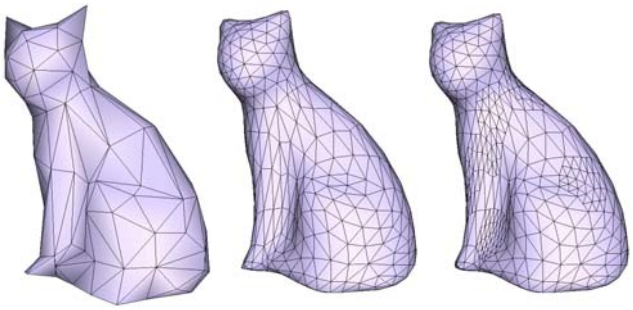


Figure 18: Successive control mesh in adaptive subdivision for the cat model with the accuracy 0.2.

Table 2 gives the subdivision depth according to the given accuracy on a regular surface (valence is 6 for every vertex), the torus. In figures 9 to 14, the dark (resp. light) faces represent the faces whose distance to the limit surface is greater (resp. less) than ε .

Accuracy ε	0.5	0.1	0.01	0.05	0.001
Subdivision depth	2	3	4	5	6

Table 2. Subdivision depth necessary for accuracy ε for the torus.

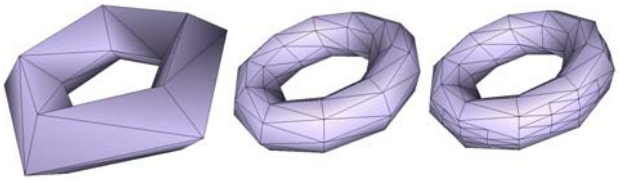


Figure 19: Subdivision of the torus ($n = 6$) with an accuracy ε of 0.5.

Figure 19 shows the number of subdivisions necessary to obtain an accuracy ε of 0.5 for the torus model which has regular valences everywhere. Figure 20 illustrates successive levels of subdivision to have an accuracy ε of 0.1. We can easily verify that subdivision level k gives a correct result whereas level $k-1$ leaves non accurate vertices (dark areas).

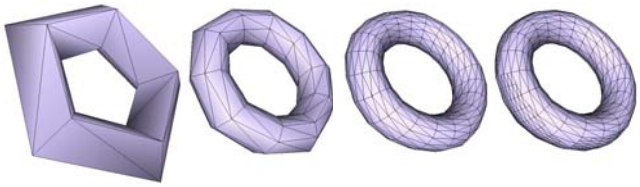


Figure 20: Subdivision of the torus ($n = 6$) with an accuracy ε of 0.1.

For a mesh with arbitrary valences, we have to pay attention to the role of valence n. n denotes the maximum valence of the mesh including the subdivision steps. When valences are strictly less than 6 on the initial control mesh, the maximum valence is 6 because all vertices inserted during the subdivision have valence 6. For instance, initial valences of the box are 4 or 5 so n = 5. For ε = 0.05, the subdivision depth k found for n = 5 is 3. But k is 4 when the maximum valence n is considered to be 6 due to subdivisions. Figure 21 shows successive subdivisions of the box. At level 3, some vertices are still further than ε from the limit surface. Indeed, subdivision depth is 4.

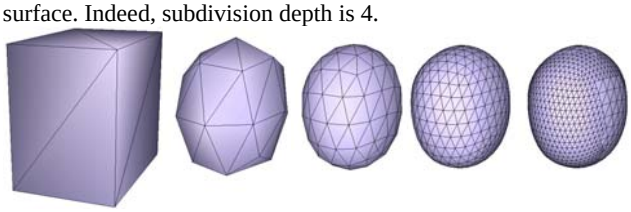


Figure 21: Box. Valence of inserted vertices must be taken into account.

In some cases, using the mean valence 6 is not sufficient. For instance, the surface in Figure 22 has a minimum valence of 4 and a maximum valence of 18. With an accuracy of 0.15, using mean valence gives a subdivision depth of 2, whereas the maximum valence gives 3, which is the true result.

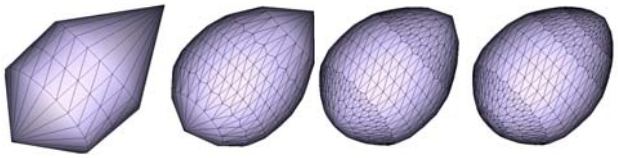


Figure 22: Spinning top. Mean valence (6) is not enough. An error remains at level 2 near the pole (vertex of valence 18).

It is then very important to choose N to be $\max(6, N^0)$ where N^0 is the maximum valence of the initial control mesh.

Table 3 gives the subdivision depth as a function of a given accuracy in the bunny model with arbitrary valences ($n \in [\![3,10]\!]$). For an accuracy ε with a difference of 2.10^{-3}, the subdivision depth increases by 1 (from 4 to 5) which represents an increased cost (memory and computation) even with adaptive subdivision.

Accuracy ε	0.5	0.1	0.05	0.012	0.01
Subdivision depth	1	2	3	4	5

Table 3. Subdivision depth necessary as a function of accuracy ε in the bunny model.

Figure 23 shows the number of subdivisions necessary to obtain an accuracy ε of 0.15 in the bunny model.

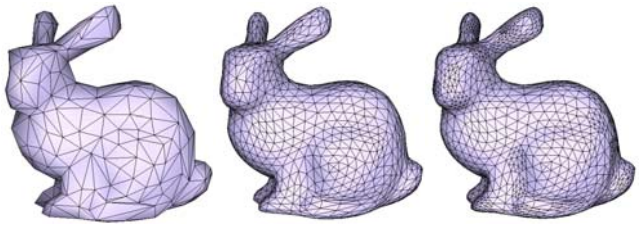
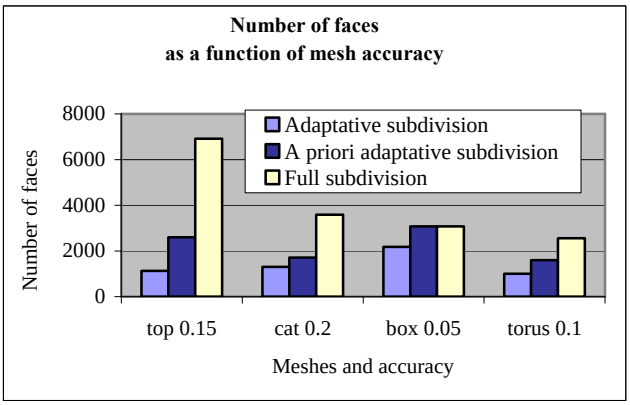
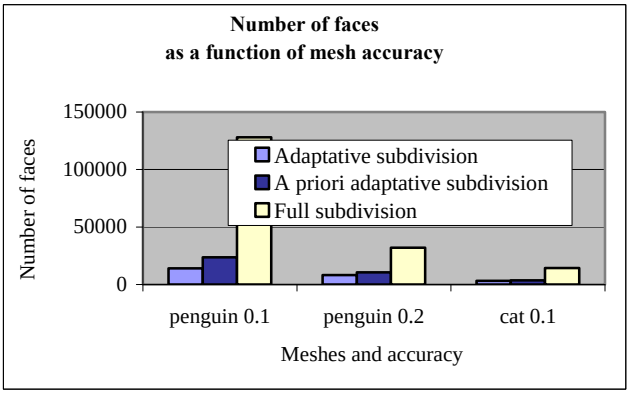


Figure 23: The bunny ($n \in [\![3,10]\!]$) with an accuracy of 0.1.



Graph 2. Number of faces found for top, cat, box and torus model according to the way of computation: successive adaptive subdivision, a priori estimated adaptive subdivision and a priori full subdivision.



Graph 3. Number of faces found for penguin and cat model according to the way of computation: successive adaptive subdivision, a priori estimated adaptive subdivision and a priori full subdivision.

It is easy to determine the number of faces of a mesh subdivided at a given accuracy without performing subdivisions. On the other hand, we do not know how to determine this number for adaptive subdivision without subdividing. So we proposed a formula in section 4.2 to upper-bound this number from the initial level. This formula gives too high a number of faces by construction but it represents the best upper-bound we can perform without subdividing. However the number of faces obtained is smaller than that of full subdivision.

The results are presented in Table 4 and Graph 2 and Graph 3. Graph 2 and 3 show for each mesh the number of faces found according to the way of computation: successive adaptive subdivision, a priori estimated adaptive subdivision and a priori full subdivision. Meshes used in Graph 2 are the following: the top model with a 0.15 accuracy, the cat model with a 0.2 accuracy, the box model with a 0.05 accuracy and the torus model with a 0.1 accuracy whereas Graph 3 is applied on the penguin model with a 0.1 then 0.2 accuracy and the cat model with accuracy 0.1. These results are completed with Table 6 where the first column gathers the additional percentage of faces in the upper-bounded subdivision compared to the real adaptive subdivision and the second column gives the percentage of faces saved in the upper-bounded subdivision compared to the full subdivision.

	Additional percentage of faces in the upper-bounded subdivision compared to the real adaptive subdivision	Percentage of faces saved in the upper-bounded subdivision compared to the full subdivision
penguin 0.1	69	81
penguin 0.2	28	67
cat 0.1	15	75
top 0.15	130	62
cat 0.2	31	52
box 0.05	41	0
torus 0.1	60	38

Table 4. Comparison of the number of faces in percentage.

6.CONCLUSION

From a distance bound between a subdivision surface and its control polyhedron we can adaptively subdivide meshes. This allows us to *a priori* determine the number of times that the control mesh has to be subdivided in order to approximate the limit surface with a given accuracy. From this number of subdivisions, mesh size can be determined for full subdivision and upper-bounded for adaptive subdivision. The determination of the mesh size, without subdividing, allows us to estimate properties of computers which are used to compute the scene. Indeed, in animation movies, there can be many meshes in a scene and they can be more or less significant. From this formula and from the data structure used, we can compute the necessary memory. Thus, if it exceeds the RAM memory of the computer (personal computer, portable playstation…), memory can be added. If the memory size cannot be increased (portable playstations for instance) scenes can be edited in order to realize the necessary compromise between quality (increasing the accuracy and thus the levels of subdivision) and memory occupation. Also, one usually considers that if the memory amount exceeds 4 GB, work can be parallelized. Moreover, for real time applications, such as virtual reality, the frame rate is a crucial criterion, which depends on the number of triangles to be displayed. A priori knowledge of this number can avoid further decimations. In this paper, the forecast of the number of faces is done only for Loop subdivision. As other subdivision schemes are currently used, we need to do the same as on this scheme. The distance used cannot always be the same. Indeed, for dual approximating schemes, we cannot upper bound the distance in the same way. Moreover for interpolating schemes, vertices of control meshes are on the limit surface, so this distance will always be null.

7.REFERENCES

[1] Amresh A., Farin G. and Razdan A., *Adaptive subdivision schemes for triangular meshes*, in *Hierarchical and Geometric Methods in Scientific Visualization*, H.H. G. Farin, and B. Hamann, editors, Editor. 2003: pp. 319-327.

[2] Catmull, E. and Clark J., *Recursively generated B-spline surfaces on arbitrary topological meshes.* Computer Aided Design, 1978. **9**(6): pp. 350-355.

[3] Hoppe H., *View-Dependent Refinement of Progressive Meshes.* Proceedings of ACM SIGGRAPH 97, 1997: pp. 189-198.

[4] Lanquetin S. and Neveu M., A *new non-uniform Loop scheme*. International Conference on Computer Graphics Theory and applications (GRAPP), Setubal, Portugal, 2006: pp. 134-141.

[5] Lanquetin S. and Neveu M., *A priori computation of the number of surface subdivision levels.* Proceedings of Computer Graphics and Vision (GRAPHICON 2004), Moscow, September 6-8, 2004: pp. 87-94.

[6] Loop, C. *Smooth Subdivision Surfaces Based on Triangles*. Department of Mathematics: Master's thesis, University of Utah, 1987.

[7] Meyer M. *et al.*, *Discrete differential-geometry operators for triangulated 2-manifolds.* VisMath, 2002.

[8] Müller K. and Havemann S., *Subdivision Surface Tesselation on the Fly using a Versatile Mesh data Structure.* Eurographics'2000, 2000. **19**(3): p. 151-159.

[9] Peters J., Wu X., *The distance of a subdivision surface to its control polyhedron,* Journal of Approximation Theory (2008), doi:10.1016/j.jat.2008.10.012.

[10] Wang H., Sun H. and Qin K., *Estimating Recursion Depth for Loop Subdivision.* International Journal of CAD/CAM, 2004, **4**(1) n°453: p. 11-18.

[11] Wu X. and Peters J., *An accurate error measure for adaptative subdivision surfaces*. Shape Modeling International, may 2005: p. 51-56.

[12] Zorin D., Schröder P. and Sweldens W., *Interactive multiresolution mesh editing.* SIGGRAPH'98 Proceedings, 1998: p. 259-268.

About the author

Sandrine Lanquetin is an assistant professor at Burgundy University, Dijon, France, Laboratory of Electronics, Informatics and Image (LEII). Research on subdivision surfaces.
Contact email : sandrine.lanquetin@u-bourgogne.fr.

Marc Neveu is a full professor at Burgundy University, Dijon, France, Laboratory of Electronics, Informatics and Image (LEII). His research concerns geometric modelling with a focus on deformable surfaces, subdivision surfaces and fractal surfaces.
Contact email : marc.neveu@u-bourgogne.fr.
Home page: http://danae.u-bourgogne.fr/Equipe/Pages/neveu/

A Reverse Scheme For Quadrilateral Meshes

Yacine Boumzaid* LE2I, Universite de Bourgogne
Sandrine Lanquetin† LE2I, Universite de Bourgogne
Marc Neveu ‡ LE2I, Universite de Bourgogne

Abstract

Reverse subdivision constructs a coarse mesh of a model from a finer mesh of this same model. In [1] Lanquetin and Neveu propose a reverse mask for the Catmull-Clark scheme which consists in locally reversing Catmull-Clark original formula for even control points, but this mask can not be applied in reversing other variants such as Quad-averaging scheme of Warren and Weimer [2]. In this paper, we derive a reverse mask for Catmull-Clark. This mask is parameterized and can also be used for reversing other quad schemes as Quad-averaging scheme.

Keywords: *Reverse Subdivision, multiresolution, Catmull-Clark scheme, Reverse loop Subdivision, Reverse Catmull-Clark Subdivision.*

1. INTRODUCTION

Since their first appearance in 1978, subdivision algorithms for generating surfaces of arbitrary topology have gained widespread popularity in computer graphics and are being evaluated in engineering applications. This development was complemented by ongoing efforts to develop appropriate mathematical tools for a thorough analysis, and today, many of the fascinating properties of subdivision are well understood. Since the earliest subdivision surfaces in 1978, many subdivision schemes were proposed. Some are approximating as Catmull-Clark [3], Loop [4] and Doo-Sabin [5], and others are interpolating as Kobbelt [6], Butterfly [7]. Moreover subdivision surfaces are more and more used in CAGD, and in this field most meshes are quadrilateral, in coherence with parametric surfaces (Bezier, B-splines, NURBS). Subdivision methods produce a sequence of increasingly fine meshes. On the contrary, it can be interesting to go quickly from a mesh to a coarser one. Using a local formula for decreasing the resolution of a mesh is a crucial element for the implementation of multiresolution surfaces. It reverses the subdivision process. While formulas for subdividing meshes are local, the existence of local formulas for the respective reverse subdivision is less evident. Using a local formula to reverse the subdivision process implies that the local connectivity of a mesh is clearly identified. For instance triangular structures, quadrilateral schemes can be detected following taubin's work [8].

There are several global methods such as multiresolution methods [9], as Loop reverse subdivision [10]. Local methods for reverse subdivision are of interest because they use small neighborhoods around a vertex (with reduced topological information) and induce small systems to solve. Samavati and Bartels determined the local reverse subdivision masks for the Butterfly and Loop scheme restricted to regular vertices (valence 6) in [11]. Samavati et al. focused on the Doo-Sabin scheme for arbitrary meshes in [12]. Samavati, et al. [13] propose a local method for the Loop scheme which consists in locally reversing the formula for a given set of vertices. Lanquetin and Neveu [1] propose a reverse mask of Catmull-Clark scheme which consists in locally reversing the Catmull-Clark original formula for even control points. The Catmull-Clark formula for even control points can be chosen in different ways. In this paper, we use the simplest one, to develop a local method to reverse Catmull-Clark and Quad-averaging subdivisions. Section 2 overviews the Catmull-Clark scheme, Quad-averaging scheme and the reverse mask for Catmull-Clark found in [1]. Then we present our method for reversing Catmull-Clark and Quad-averaging schemes in Section 3. Results are shown in Section 4, illustrating different cases that can occur.

*Yacine.Boumzaid@u-bourgogne.fr
†Sandrine.Lanquetin@u-bourgogne.fr
‡Marc.Neveu@u-bourgogne.fr

2. BACKGROUND

Subdivision is a repetitive refinement process that gradually converts a given coarse mesh to finer meshes to generate a smooth surface at the limit. An arbitrary mesh M can be denoted by the pair (F, V), where F shows the faces of M, and V denotes the vertices of M. Each element $v \in V$has the spatial coordinates, (x, y, z), and each element $f \in F$ is assigned a list that includes all indices of its adjacent vertices in V. Catmull-Clark, Doo-Sabin, Butterfly and Loop subdivisions are some important cases . The input for subdivision methods is $M^0 = (F^0, V^0)$ a control mesh. In each step of a subdivision method, the mesh $M^k = (F^k, V^k)$ is converted to a new and finer mesh $M^{k+1} = (F^{k+1}, V^{k+1})$. This conversion is done through some local affine operations on V^k, together with a mapping process from the faces of F^k to those of F^{k+1}. The affine operations are usually described by masks, or matrices, that are smoothing filters. Consequently, by successively applying a subdivision method, a hierarchy $M^0, M^1, M^2, ..., M^k, \ldots$ is obtained that usually converges to a smooth surface.

2.1 The Catmull-Clark Subdivision Scheme

In computer graphics, the Catmull-Clark algorithm is used in subdivision surface modeling to create smooth surfaces. It was devised by Edwin Catmull and Jim Clark in 1978 as a generalization of bicubic uniform B-spline surfaces to arbitrary topology. Catmull-Clark surfaces are defined recursively, using the following refinement scheme: Start with a mesh of an arbitrary polyhedron $M^k = (F^k, V^k)$. All the vertices in the mesh shall be called original points. In each step of subdivision, the mesh $M^k = (F^k, V^k)$ is converted to a new and finer mesh $M^{k+1} = (F^{k+1}, V^{k+1})$. The set of new vertices V^{k+1} includes three types of vertices (Figure 1):

• A face control point (f_i^{k+1}) for an n-gon is computed as the average of the corners of the polygon:

$$f_i^{k+1} = \frac{1}{N} \sum_{i=1}^{N} v_i^k \tag{1}$$

where N is the number of corners.

• An edge control point (e_i^{k+1}) is the average of the endpoints of the edge and newly computed face control points of adjacent faces:

$$e_i^{k+1} = \frac{1}{4}(v^k + v_i^k + f_i^{k+1} + f_{i+1}^{k+1}). \tag{2}$$

• An even control point (v^{k+1}) is a weighted average of its incident

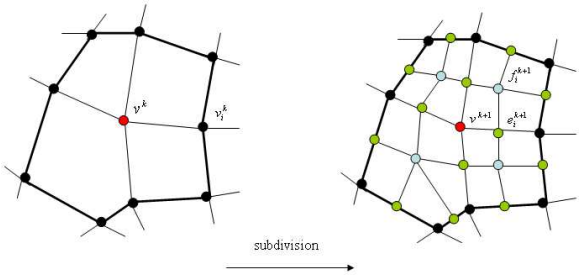


Figure 1: Catmull-Clark subdivision.

vertices of the same level and of the face points of the incident faces:

$$v^{k+1} = \frac{n-2}{n}v^k + \frac{1}{n^2}\sum_{i=1}^{n} e_i^k + \frac{1}{n^2}\sum_{i=1}^{n} f_i^{k+1} \qquad (3)$$

The new mesh will consist only of quadrilaterals, which won't in general be planar. The new mesh will generally look smoother than the old mesh. Repeated subdivision results in smoother meshes. It can be shown that the limit surface obtained by this refinement process is C^2 at ordinary vertices and C^1 everywhere else. If the rules of Catmull-Clark scheme are defined for meshes with quadrilateral faces, then in each step of subdivision, each face in F^k is replaced by four new quads that become the faces of F^{k+1} (Figure 2). In this case the masks are shown in Figure3.

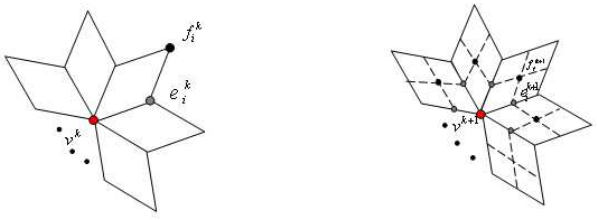


Figure 2: Situation around a vertex v^k before and after subdivision.

Let $e_1^k, e_2^k, ...e_n^k$, $f_1^k, ...f_n^k$be the set of neighbors of v^k in M^k. In addition, let v^{k+1} be the even vertex of v^k, and $e_1^{k+1}, e_2^{k+1}, ...e_n^{k+1}$, $f_1^{k+1}, ...f_n^{k+1}$ be the corresponding edge vertices and face vertices (n is the valence). The edge vertices e_i^{k+1}, face vertices f_i^{k+1} and even vertices v^{k+1} defined by equations (1), (2) and (3)become:

The edge vertex :

$$e_i^{k+1} = \frac{3}{8}v^k + \frac{3}{8}e_i^k + \frac{1}{16}f_i^k + \frac{1}{16}e_{i+1}^k + \frac{1}{16}e_{i-1}^k + \frac{1}{16}f_{i-1}^k \qquad (4)$$

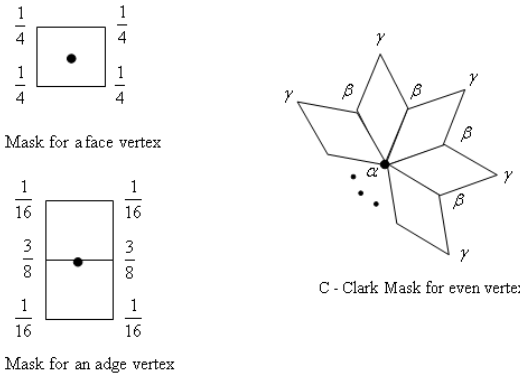


Figure 3: General case for Catmull-Clark masks: $\beta = \frac{3}{2n^2}$, $\gamma = \frac{1}{4n^2}$, $\alpha = 1 - n(\beta + \gamma)$.

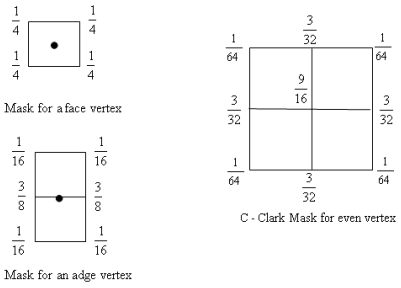


Figure 4: Catmull-Clark masks for a regular vertex

The face vertex :

$$f_i^{k+1} = \frac{1}{4}v^k + \frac{1}{4}e_i^k + \frac{1}{4}f_i^k + \frac{1}{4}e_{i+1}^k \qquad (5)$$

The position of even vertex:

$$v^{k+1} = \alpha v^k + \beta\sum_{i=1}^{n} e_i^k + \gamma\sum_{i=1}^{n} f_i^k \qquad (6)$$

With: $\alpha = 1 - n(\beta + \gamma)$, $\beta = \frac{3}{2n^2}$, $\gamma = \frac{1}{4n^2}$

2.2 Quad-Averaging scheme

The Quad-averaging scheme was described by Warren and Weimer in [2]. It can produce a smooth surface but not necessarily at extraordinary vertices. To apply these subdivision rules to an arbitrary quad mesh, we need only to generalize the vertex rule from the valence four case (Figure 4) to vertices of arbitrary valence. In the regular case, the mask can be decomposed into the sum of four sub-masks:

$$\begin{pmatrix} \frac{1}{64} & \frac{3}{32} & \frac{1}{64} \\ \frac{3}{32} & \frac{9}{16} & \frac{3}{32} \\ \frac{1}{64} & \frac{3}{32} & \frac{1}{64} \end{pmatrix} = \frac{1}{4}\Big[\begin{pmatrix} \frac{1}{16} & \frac{3}{16} & 0 \\ \frac{3}{16} & \frac{9}{16} & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & \frac{3}{16} & \frac{1}{16} \\ 0 & \frac{9}{16} & \frac{3}{16} \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ \frac{3}{16} & \frac{9}{16} & 0 \\ \frac{1}{16} & \frac{3}{16} & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{9}{16} & \frac{3}{16} \\ 0 & \frac{3}{16} & \frac{1}{16} \end{pmatrix}\Big]$$

In the case of extraordinary vertex the mask of an even vertex for this scheme is shown in Figure(5), and the position of this vertex is given by:

$$v^{k+1} = \frac{9}{16}v^k + \frac{3}{8n}\sum_{i=1}^{n} e_i^k + \frac{1}{16n}\sum_{i=1}^{n} f_i^k \tag{7}$$

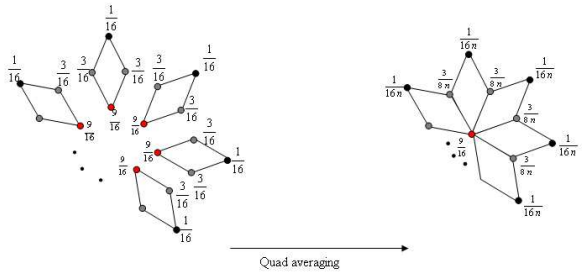


Figure 5: Quad-averaging mask

2.3 Boundary

When we encounter boundary vertices, we need to use boundary masks that are usually different from the interior mask. It is important that subdivision at any point on the boundary be independent of any point in the interior of the mesh. This permits two surfaces to be joined along a boundary curve. Therefore, cubic B-spline subdivision masks (Figure 6) for curves can be used as the boundary masks of Catmull-Clark subdivision.

• Cubic B-splines are a popular class of curves that are smooth and can be built with a simple subdivision scheme. Given a polygonal curve V^k , we denote the ith vertex of V^k by v_i^k. The edges of the polygonal curve are implicit in this representation since consecutive vertices (v_i^k and v_{i+1}^k) form an edge. The subdivision rules for cubic B-spline then have the form

$$v_{2i}^{k+1} = \frac{1}{8}v_{i-1}^k + \frac{3}{4}v_i^k + \frac{1}{8}v_{i+1}^k \tag{8}$$

$$v_{2i+1}^{k+1} = \frac{1}{2}v_i^k + \frac{1}{2}v_{i+1}^k \tag{9}$$

$$v_{2i-1}^{k+1} = \frac{1}{2}v_{i-1}^k + \frac{1}{2}v_i^k \tag{10}$$

Note that there are two rules, the number of vertices in the polygonal curve doubles during each round of subdivision.

In particular, the vertex v_i^k is repositioned to v_{2i}^{k+1} by the mask of even vertex $\{\frac{1}{8}, \frac{3}{4}, \frac{1}{8}\}$ while v_{2i+1}^{k+1} and v_{2i-1}^{k+1} are inserted at the midpoint of the edge v_i^k to v_{i+1}^k, and v_{i-1}^k to v_i^k by the mask of odd vertices $\{\frac{1}{2}, \frac{1}{2}\}$.

• The masks of even vertex and odd vertices of cubic B-splines form the subdivision mask for cubic B-splines $\{\frac{1}{8}, \frac{1}{2}, \frac{3}{4}, \frac{1}{2}, \frac{1}{8}\}$, the tensor product of this mask is the subdivision mask for Catmull-Clark in the regular case.

Let $S = \{\frac{1}{8}, \frac{1}{2}, \frac{3}{4}, \frac{1}{2}, \frac{1}{8}\}$ be the subdivision mask for cubic B-splines, this mask is given in term of generating functions by:

$$S(z) = \frac{1}{8}z^{-2} + \frac{1}{2}z^{-1} + \frac{3}{4} + \frac{1}{2}z + \frac{1}{8}z^2$$

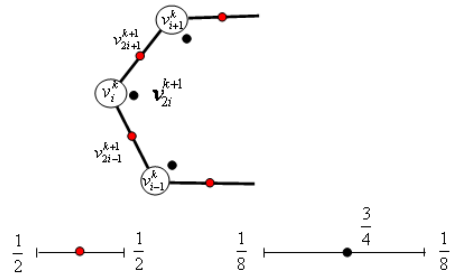


Figure 6: Cubic B-spline subdivision.

Then the bicubic B-spline surface can be expressed as the tensor product:

$S(z_1) \otimes S(z_2) = T(z_1, z_2) = Z_1 T Z_2^t$

where:

$\otimes$ is the tensor product,

$Z_1 = \begin{pmatrix} z_1^{-2} & z_1^{-1} & 1 & z_1 & z_1^2 \end{pmatrix}$,

$Z_2 = \begin{pmatrix} z_2^{-2} & z_2^{-1} & 1 & z_2 & z_2^2 \end{pmatrix}$ and

$$T = \frac{1}{64}\begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

is the subdivision masks for Catmull-Clark in the regular case from which we get three different subdivision rules, one for the even vertex of the surface, one for the edge vertices and one for the face vertices (Figure 4).

It is easy to prove that the tensor product of the mask for even vertex of cubic B-splines is the Catmull-Clark mask of even vertex in the regular case:

$$\{\frac{1}{8}, \frac{3}{8}, \frac{1}{8}\} \otimes \{\frac{1}{8}, \frac{3}{8}, \frac{1}{8}\} = \begin{pmatrix} \frac{1}{64} & \frac{3}{32} & \frac{1}{64} \\ \frac{3}{32} & \frac{9}{16} & \frac{3}{32} \\ \frac{1}{64} & \frac{3}{32} & \frac{1}{64} \end{pmatrix}$$

We will use this property of subdivision to prove that the tensor product of the reverse mask for cubic B-splines is the reverse mask for a regular vertex of Catmull-Clark.

2.4 Reverse Mask for Catmull-Clark

In reverse subdivision, we know vertices of the level $k+1$ and we want to find the position of an even vertex v^k. In [1], Lanquetin and Neveu propose a reverse mask for the Catmull-Clark scheme for all arbitrary meshes (triangular, quadrilateral), which consists in locally reversing the originals formulas of edge vertices, face vertices and even vertex of Catmull-Clark(equation (1), (2) and (3)). The position of even vertex v^k is given by:

$$v^k = \frac{n}{n-3}v^{k+1} + \frac{4}{n(3-n)}\sum_{i=1}^{n} e_i^{k+1} + \frac{1}{n(n-3)}\sum_{1=1}^{n} f_i^{k+1} \tag{11}$$

More details can be found in[1]

But with this method we can not reverse other variants of quad schemes as Quad-averaging scheme of Warren and Weimer [2].

3. REVERSE SCHEMES FOR QUADRILATERAL MESHES

In this section we use an other method to determine the reverse mask of Catmull-Clark found in [1]. With this method our mask is parameterized and can also be used for reversing Quad-averaging scheme, and we will prove that we can use some properties of subdivision for the reverse subdivision, such as tensor product and the generalization of the vertex of Quad-averaging rule from the regular case to find the mask of arbitrary case:

• To apply the reverse Quad-averaging scheme to an arbitrary quad mesh, we need only to generalize the vertex rule from the valence four to vertices of arbitrary valence.

• The tensor product of the reverse mask of cubic B-splines is the reverse mask for a regular vertex of Catmull-Clark.

3.1 General Reverse Quadrilateral scheme

For the reverse process, it is necessary to construct a mask to map V^{k+1} to V^k. For an extraordinary vertex, assume the general situation shown in Figure 7. We know v^{k+1}, e_i^{k+1} ,f_i^{k+1} and we want to find v^k by a new mask such that the following conditions are met:

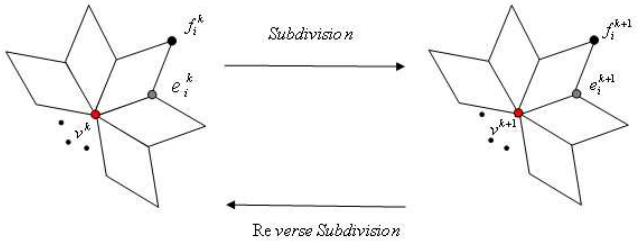


Figure 7: Situation for an extraordinary vertex

1• The operation of the new mask must be affine.

2•Weights of e_i^{k+1} in the mask must be equal. The same condition holds for weights of f_i^{k+1}. This is similar to the Catmull-Clark mask of equation (6)

3• The new mask must be a reverse of the subdivision mask i.e. the action of subdivision mask of Equation 4 , 5 and 6 on v^k and its neighbors must exactly reconstruct v^{k+1}. Condition (2) provides the diagram of Figure 8 for the reverse mask. In this diagram $\alpha^{'}$ is the weight of v^{k+1}, $\beta^{'}$ is the weight of e_i^{k+1} and $\gamma^{'}$ is the weight of f_i^{k+1} in the reverse mask. We now determine the weights $\alpha^{'}$, $\beta^{'}$ and $\gamma^{'}$ so that conditions (1) and (3) are also satisfied.

From condition (3) we have:

$v^k = \alpha^{'} v^{k+1} + \beta^{'} \sum_{i=1}^{n} e_i^{k+1} + \gamma^{'} \sum_{i=1}^{n} f_i^{k+1}$

From equations (4), (5) and (6), we obtain:

$v^k = (\alpha\alpha^{'} + \frac{3}{8}\beta^{'}n + \frac{1}{4}\gamma^{'}n)v^k + (\alpha^{'}\beta + \frac{1}{2}\beta^{'} + \frac{1}{2}\gamma^{'}) \sum_{i=1}^{n} e_i^k$

$$+(\alpha^{'}\gamma + \frac{1}{8}\beta^{'} + \frac{1}{4}\gamma^{'}) \sum_{i=1}^{n} f_i^k \qquad (12)$$

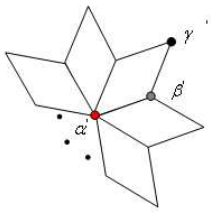


Figure 8: Situation for an extraordinary vertex.

From equation (12), we get the following system:

$$\begin{cases} \alpha\alpha' + \frac{3}{8}\beta' n + \frac{1}{4}\gamma' n = 1 \\ \alpha'\beta + \frac{1}{2}\beta' + \frac{1}{2}\gamma' = 0 \\ \alpha'\gamma + \frac{1}{8}\beta' + \frac{1}{4}\gamma' = 0 \end{cases} \qquad (13)$$

We solve equation (13) with respect to α, β, γ. As $\alpha = 1 - n(\beta + \delta)$ we get a system with respect to β and γ:

$$\begin{cases} \alpha^{'} = \frac{1}{1-2n\beta} \\ \beta^{'} = \frac{4(2\gamma-\beta)}{1-2n\beta} \\ \gamma^{'} = -\frac{2(4\gamma-\beta)}{1-2n\beta} \end{cases} \qquad (14)$$

with: $1 - 2n\beta \neq 0$.

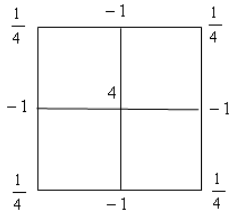


Figure 9: Reverse mask for regular vertex.

3.2 Reverse Catmull-Clark scheme

Equation (14) is a parametric formula for the reverse mask and can be applied for regular and extraordinary vertices.

In the case of regular vertex for all $(\alpha, \beta, \gamma) = (\frac{9}{16}, \frac{3}{32}, \frac{1}{64})$ we find:

$(\alpha', \beta', \gamma') = (4, -1, \frac{1}{4})$ (Figure 9).

For all $n \neq 3$ and for all $(\beta = \frac{3}{2n^2}, \gamma = \frac{1}{4n^2}$ we find:

$(\alpha', \beta', \gamma') = (\frac{n}{n-3}, \frac{4}{n(3-n)}, \frac{-1}{n(3-n)})$. Then the position of an even vertex is given by:

$v^k = \frac{n}{n-3} v^{k+1} + \frac{4}{n(3-n)} \sum_{i=1}^{n} e_i^{k+1} + \frac{1}{n(n-3)} \sum_{1=1}^{n} f_i^{k+1}$

We can see that with our method we get the same mask found in [1] equation (11) . In case of meshes with vertices of valence 3 more details can be found in [1].

3.3 Reverse mask for Quad averaging Scheme

Formula (14) is parameterized, and thus can be applied directly to Quad-averaging scheme. This implies that for all n we find:

$(\alpha', \beta', \gamma') = (4, \frac{-4}{n}, \frac{1}{n})$

We can see that in the regular case, the reverse mask can be decomposed into the sum of four sub-masks:

$$\begin{pmatrix} \frac{1}{4} & -1 & \frac{1}{4} \\ -1 & 4 & -1 \\ \frac{1}{4} & -1 & \frac{1}{4} \end{pmatrix} = \frac{1}{4}[\begin{pmatrix} 1 & -2 & 0 \\ -2 & 4 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & -2 & 1 \\ 0 & 4 & -2 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 4 & -2 \\ 0 & -2 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ -2 & 4 & 0 \\ 1 & -2 & 0 \end{pmatrix}]$$

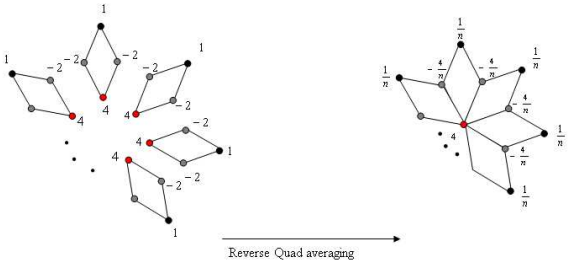


Figure 10: Reverse mask for Quad-averaging scheme .

To apply the reverse Quad-averaging mask to an arbitrary quad mesh, we only generalize the vertex rule from the valence four (Figure 9) to vertices of arbitrary valence illustrated in Figure 10.

3.4 Invariance Affine

In the case of regular vertex $(n = 4, \alpha = \frac{9}{16}, \beta = \frac{3}{32}, \gamma = \frac{1}{64})$, equation (14) gives $\alpha^{'} = 4, \beta^{'} = -1, \gamma^{'} = \frac{1}{4}$ (Figure 9). In this case the sum of the weights is one. This property is generally correct for the Catmull-Clark and Quad-averaging schemes, since

$$\alpha^{'} + n\beta^{'} + n\gamma^{'} = 1$$

this property holds for $n \neq 3$ in the case of Catmull-Clark, and for all n in the case of Quad-averaging.

3.5 Reverse Mask Of The Boundary Vertices

We have used cubic B-spline mask for boundary vertices. Therefore, we need to find a reverse mask for the cubic B-spline subdivision. In Bartels and Samavati, several masks for cubic B-spline subdivision are provided.

From equation (8), (9) and (10):

$$\begin{cases} v_{2i}^{k+1} = \frac{1}{8}v_{i-1}^{k} + \frac{3}{4}v_{i}^{k} + \frac{1}{8}v_{i+1}^{k} \\ v_{i-1}^{k} = 2v_{2i-1}^{k+1} - v_{i}^{k} \\ v_{i+1}^{k} = 2v_{2i+1}^{k+1} - v_{i}^{k} \end{cases}$$

Then:

$$v_{i}^{k} = -\frac{1}{2}v_{2i-1}^{k+1} + 2v_{2i}^{k+1} - \frac{1}{2}v_{2i+1}^{k+1}$$

This corresponds to the mask used in [13]

We can prove that as with subdivision the tensor product of the reverse mask of a cubic B-spline $\{\frac{-1}{2}, 2, \frac{-1}{2}\}$ is the reverse mask for a regular vertex of Catmull-Clark (Figure 9).

Let $M = \{-\frac{1}{2}, 2, -\frac{1}{2}\}$ be the reverse mask for cubic B-splines, we can write this mask in term of a generating fonction in the form:

$$M(z) = -\frac{1}{2}z^{-1} + 2 - \frac{1}{2}z$$

The tensor product:

$M(z_1) \otimes M(z_2) = M^{'}(z_1, z_2) = Z_1 M^{'} Z_2^t$

With:

$$Z_1 = (z_1^{-1}, 1, z_1)$$

$$Z_2 = (z_2^{-1}, 1, z_2)$$

$$M^{'} = \begin{pmatrix} \frac{1}{4} & -1 & \frac{1}{4} \\ -1 & 4 & -1 \\ \frac{1}{4} & -1 & \frac{1}{4} \end{pmatrix}$$

We can see that $M^{'}$ is the reverse mask for Catmull-Clark in the regular case (see Figure 9).

4. RESULTS

To illustrate the method described in this paper, several examples are shown. The first example consists in rebuilding an initial mesh with vertices of valence 4 (torus) subdivided three times (Figure 11). The second example consists in rebuilding an initial mesh with extraordinary vertices subdivided by the Quad-averaging scheme (Figure 12). And the third example is a mesh with boundary subdivided by Quad-averaging scheme (Figure 13). In the three examples the successive meshes can be constructed with the general method described in the previous sections.

In the case of the meshes with vertices of valence 3 subdivided by Catmull-Clark scheme shown in [1]

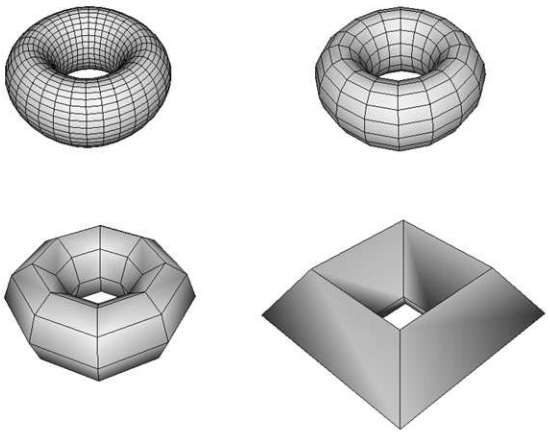


Figure 11: Reverse Catmull-Clark subdivision applied to on the torus mesh subdivided three times.

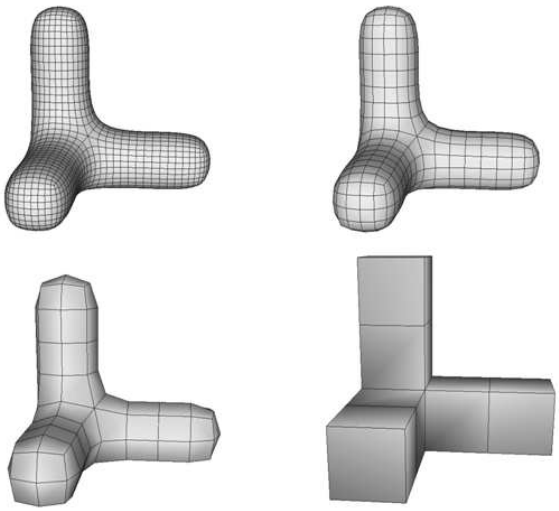


Figure 12: Reverse Quad-averaging scheme applied on a mesh with valences equal to 3, 4, 6 subdivided three times

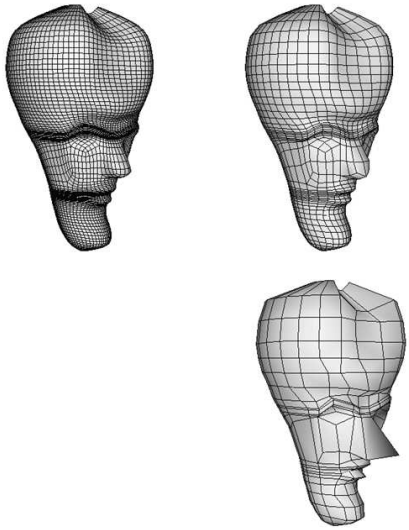


Figure 13: Reverse Quad-averaging Scheme applied on a mesh with boundary subdivided twice

5. CONCLUSION

We have used an other method to determine the reverse mask for Catmull-Clark found in[1]. With this method we have constructed a parameterized reverse mask for Catmull-Clark subdivision. The parameterization allows to reverse other quad schemes, such as the Quad-averaging scheme. we also have seen that we can use some properties of subdivision for the reverse subdivision, such as the tensor product and the quad averaging. This work proposes a general reverse scheme for quadrilateral structures that can be detected in irregular meshes using taubin's algorithms.

Future works will focus on mixed structures (i.e containing both triangular and quadrilateral faces). Tri-quad subdivision schemes have already been proposed. The problem is not solved for reverse tri-quad subdivision schemes.

ABOUT THE AUTHOR

Yacine Boumzaid is a PHD student at Burgundy University, Dijon , France, Laboratory of Electronics, Informatics and Image (LE2I). research on subdivision surfaces and reverse subdivision.contact email : Yacine.Boumzaid@u-bourgogne.fr.

Sandrine Lanquetin is an assistant professor at Burgundy University, Dijon, France, Laboratory of Electronics, Informatics and Image (LE2I). Research on subdivision surfaces. Contact email : sandrine.lanquetin@u-bourgogne.fr.

Marc Neveu is a full professor at Burgundy University , Dijon , France, Laboratory of Electronics, Informatics and Image (LE2I). His research concerns geometric modelling with a focus on deformable surfaces, subdivision surfaces and fractal surfaces. Contact email : marc.neveu@u-bourgogne.fr. Home page:http://danae.u-bourgogne.fr/Equipe/Pages/neveu/

6. REFERENCES

[1] S. Lanquetin and M. Neveu, "Reverse catmull-clark subdivision," *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'(WSCG)*, pp. 319–326, February 2, 2006.

[2] J. Warren and H. Weimer, *Subdivision Methods for Geometric design*, Morgan Kaufmann, New York, 2002.

[3] E. Catmull and J. Clark, "Recursively generated b-spline surfaces on arbitrary topological surfaces," *Computer Aided Design*, vol. 10, no. 6, pp. 350–355, November 1978.

[4] C. Loop, "Smooth subdivision surfaces based on triangles," M.S. thesis, university of Utah, 1987.

[5] D. Doo and M sabin, "Behaviour of recursive subdivision surfaces near extraordinary points.," *Computer Aided Design*, pp. 356–360, 1978.

[6] L. Kobbelt, "Interpolatory subdivision on open quadrilateral nets with arbitrary topology," pp. 409–420, 1996.

[7] N. Dyn, D. Levin, and J. A. Gregory, "A butterfly subdivision scheme for surface interpolation with tension control," *ACM Trans. Graph.*, vol. 9, no. 2, pp. 160–169, 1990.

[8] G Taubin, "Detecting and reconstructing subdivision connectivity," *The Visual Computer*, vol. 18, no. 5/6, pp. 357–367, 2002.

[9] T. DeRose. M. Lounsbery and J. Warren, "Multiresolution analysis for surfaces of arbitrary topological type," *ACM Trans. Graph.*, vol. 16, no. 1, pp. 34–73, 1997.

[10] P. Mongkolnam, A. Razdan, and G. Farin, "Reverse engineering using loop subdivision," *Computer-Aided Design and applications*, pp. 619–626, 2004.

[11] F. Samavati and R. Bartels, "Reversing subdivision rules: local linear conditions and observations on inner products," *J. Comp. and Appl. Math.*, vol. 119, no. 1-2, pp. 29–67, 2001.

[12] F. Samavati, M. A. Nezam, and R. Bartels, "Multiresolution surfaces having arbitrary topologies by a reverse doo subdivision method," *Computer Graphics Forum*, vol. 21, no. 2, pp. 121–134, 2002.

[13] H. R Smith C. Samavati, F. Pakdel and P. Prusinkiewicz, "Reverse loop subdivision," *Technical report University of Calgary*, november 4, 2003.

Similarity estimation for computerize footwear fit

Bita Turesavadkoohi [1,2,3], Raffaele de Amicis [1,2,4]
[1] Department of Information Engineering and Computer Science, University of Trento, Via Sommarive 14 I-38100, Povo, Trento, Italy
[2] Fondazione GraphiTech, Via alla Cascata, 56/c 38100 Povo, Trento, Italy
{[3] bsavadk@dit.unitn.it & [4] raffaele.de.amicis@graphitech.it}

Abstract

As consumers are becoming increasingly selective of what they wear on their feet, manufacturing encountered the problem of developing right footwear. It is widely accepted that the three-dimensional model of foot can help in good shoe fitting.
Footwear fitter have been using manual measurement for a long time, but the combination of 3D scanning systems with mathematical technique makes possible the development of systems, which can help in the selection of good footwear for a given customer. In this paper, we proposed new approach for finding footwear fit within the shoe last data base.
Our new approach is based on the efficient algorithm for cutting 3D triangle mesh to several sections toward heel and toe. Then the area of each contour is calculated and compared with area of equal section in shoe last data base for finding footwear fit.
The first step is to fill holes in triangle mesh; after solving this post-process problem, our method is applied for finding footwear fit within shoe last data base.

Keywords: *shoe last, footwear, mesh, hole.*

1. INTRODUCTION

Very few standards exist for fitting products to people. Footwear fit is a noteworthy example for consumer considerations when purchasing shoes. As a result, the footwear manufacturing industry, in order to achieve commercial success is challenged to produce the differing customer preferences through product variety.
Loose shoes (even thought function may be impaired) are not as uncomfortable as when the shoes are tight. Properly constructed footwear may provide the right pressure and force at the different locations on the foot surface, and this may result in improved comfort, fit and foot health.
Unlike any other consumer product, personalized footwear or the matching of footwear to feet is not easy if delivery of discomfort is predominantly caused by localized pressure induced by a shoe that has a design unsuitable for that particular shape of foot [1].
The design of new shoes starts with the design of the new shoe last. The shoe last is a wooden or metal model of human foot on which shoes are shaped.
Nowadays, with development of 3D acquisition devices, automatic process, producing custom-tailored footwear is reasonable, if the custom last can be automatically produced based on consumer's foot shape.
A new approach to computerize footwear fit proposed in this paper. Due to both the object complexity and scanning process, some areas of the object outer surface may never be accessible, thus obtained data typically contains missing pieces and holes. However, this deficiency is not acceptable, where the geometric models are using in design process in manufacturing. Thus certain repair must be done before taking these models into comparisation process. When the complete model is obtained, first the foot mesh is cut into several sections towards the heel and toe. Then the area of each section(avail able contour) is calculated and compared with the area of equal sections in shoe last data base.

In summary, the problem of finding the best fitting shoe within a given shoe last data base, using input data taken from the 3D foot scan of client consists of two main sub problem:

- Filling hole in triangle mesh.
- Cutting the model to several sections to search though the shoe last data base for finding footwear fit.

Figure 1 shows process presented in this paper.

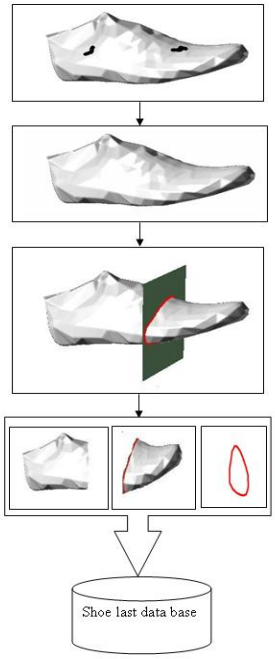


Figure 1: Steps for finding best fit from shoe last data base.

This paper is structured as follows: Section 2 reviews related work on footwear fit studies, while in Section 3 we present the filling hole in triangle mesh for building complete model. An efficient algorithm for cutting the 3D model is described in Section 4. Finally, conclusion and remarks are summarized in Section 5

2. RELATED WORK

Customer focus can influence today's business. Accurate fitting is very important factor in footwear manufacturing industry, for achieving commercial success. Mass customization starts with understanding individual customer's requirements and it finishes with fulfillment process of satisfying the target customers with

near mass production efficiency.
Properly constructed footwear improve compatibility between footwear and foot, thus contributing to fit and comfort [2], [3].
The issue of good shoe fit was posed as early as 1500 B.C. in 'Ebers Papyrus', which described a wide range of illnesses resulting from poor foot-shoe interaction [3].
Plastic tapes and Anthropometers are commonly used for obtaining measurements on people [4]. In traditional manufacturing, the device such as the Ritz Stick device [5], the Brannock device [6], the Scholl device [7], caliper and tape are always used for measurement of foot dimensions.
Depending upon the relative location of foot with shoe, the comparatively large dimensional differences can be causes either tight or loose shoe. Tight shoe will produce compression which may result in discomfort. On the other hand, when the shoe is loose, this may also lead to discomfort due to friction between the shoe and foot.
Properly constructed footwear may provide the right pressure and force at the different locations on the foot surface, and this may result in improved comfort, fit and foot health.
The design of shoe last, which represents approximate shape of human foot is the 'heart' of shoemaking because it mainly determines the foot shape, fit and comfort qualities.
Because of the complexity and the constraints imposed by the footwear manufacturing process, most importantly, the last manufacturing process, the custom footwear is expensive to produce.
In traditional, the process of foot measuring and making an accurate custom shoe last was always complicated and time consuming because the shoe maker must manually measure the specific consumer's foot and the last manufactured by last maker experience.
There are already some approaches in literature [8], [9], and [10]. The typical suggestion coming from literature is selecting a shoe last from a shoe last data base or deforming it into one that fits the scanned foot data. Authors in [11], quantify footwear fit and predict the fit-related comfort with colourcode mismatch between human foot and shoe last.
Li and Jneja [8], suggested to store front and back part of shoe last separately, to generate smooth surface between two given disjoined surfaces of the front and rear parts of the shoe last to obtain the new shoe last. This method is helpful for companies which already maintain library of last rear parts, so they need only front parts of last shapes to be designed as fashion suggests.
However, this method for custom tailored footwear designing is not very accurate because the consumers foot may change from time to time.
A meaningful way to evaluate footwear compatibility would be to determine the dimensional difference between the foot and shoe. The approach proposed in this paper is meant to cut the foot model to several section. Then the area of each section (available contour) is calculated and compared with the area of equal sections in shoe last data base. So that best fit can be obtained relatively automatically and quickly.

3. PRELIMINARIES

A *triangular mesh* is defined as a set of vertices and a set of oriented triangles that join these vertices. Two triangles are *adjacent* if they share a common edge.
A *boundary edge* is an edge adjacent to exactly one triangle. A *boundary vertex* is a vertex that is adjacent to a boundary edge. A *boundary triangle* is a triangle that own one or two boundary vertices. A *hole* is a closed cycle of boundary edges. A given hole is assumed to have no islands.
1-ring triangles of vertex are all triangles that share one common vertex. *1-ring edges* of vertex are all edges that share one common vertex and, all vertices on 1-ring edges of a vertex (except itself) are called *1-ring vertices* of the vertex.
A *vertex based topological* structure is used which records 1-rings vertices, 1-ring edges and 1-ring triangles of every vertex.
Average normal of the 1-ring triangular of the vertex is defined *vertex normal*. Figure 2 illustrated the preliminaries.

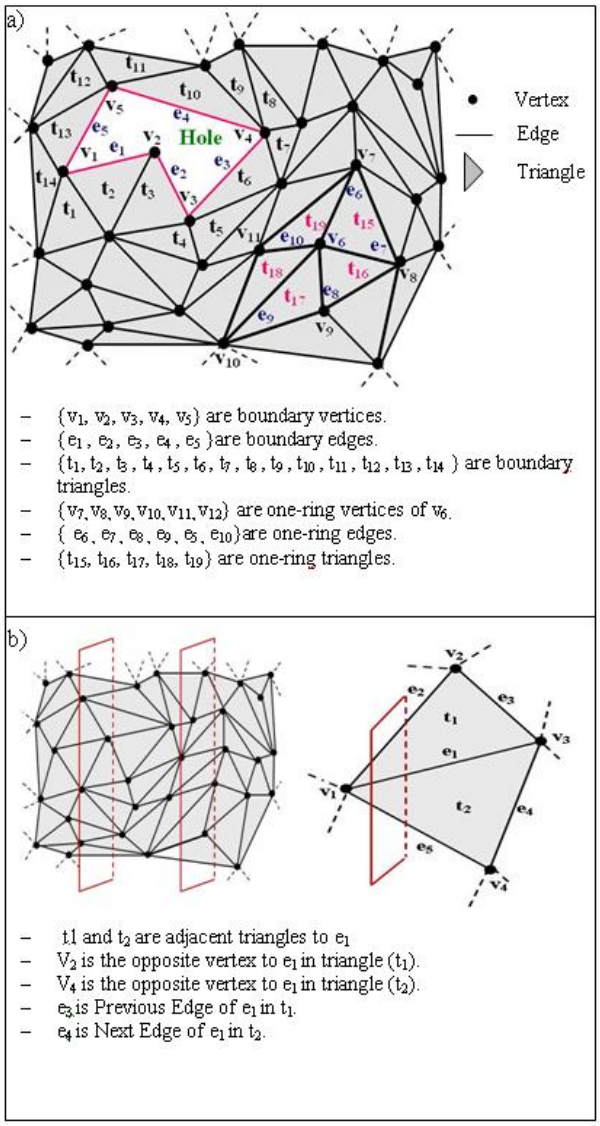


Figure 2: a) Preliminaries related to triangle mesh and hole. b) Preliminaries related to triangle mesh with intersection planes for cutting shape to several sections.

4. ACURATE 3D FOOT EXTRACTION

The development of acquisition devices that can easily and quickly acquire enormous number of surface points from a physical part have could possible automatic process of 3D models. However, several factors such as occlusion, low reflectance or even missing pieces in the original geometry can lead to incomplete data and this deficiency is not acceptable when the 3D model is taking into actual application. Thus certain repair must be done before taking these models into actual application.

In the literature we have surveyed, existing approach to fill holes in meshes can be distinguished two main categories: the geometric and non-geometric approaches.
Among the non-geometric approaches, authors in [12] detected the mesh areas that have to be filled with using volumetric representation. Davis et al [13] filled the gaps by applying volumetric process to extend a signed distance function through this volumetric representation until its zero set bridges whatever holes may be present. A similar approach has been developed by Authors in [14] for the simplification and the repairing of polygonal meshes. The advantage of this approach is working well for complex holes and drawback of current method include time-consuming and may generate incorrect topology in some case.
Considering the geometric approaches, the hole is filled in [15] with minimum area triangulation of its contour. Then the triangulation is refined so that the triangle density agrees with the density of the surrounding mesh triangles. Finally, the hole is smoothed with fairing technique based on an umbrella operator [16].
A satisfactory hole filling method should: 1. run in reasonable time. 2. be enable to patch an arbitrary holes for any model. 3. cover the missing geometry well. A hole filling process that is implemented here is quite similar to [17] and is summarized in following steps:

- Identify holes in triangle mesh. Holes can be identified automatically by looking close loop of boundary edges.
- Cover the holes with Advance Front Mesh technique.
- Modify the triangles in the initial patch mesh by estimating desirable normals instead relocating them directly.
- Rotate triangle by local rotation.
- Make algorithm more accurate by re-positioning these coordinate by solving the Poisson equation according to desirable normal and boundary vertices of the hole.
- Update the coordinate to make the smoothed patch mesh.

4.1 Hole patching

At first the hole is identified automatically by looking close loop of boundary edge. Then the Advance front mesh [18] technique is applied over the hole to generate an initial patch mesh as follows:

Step 1: The angle α_i between two adjacent boundary edges at each vertex v_i on the front is calculated.

Step 2: As Figure.3(a) depicts, when the angle α is less that or equal to 75°, we simply connect the neighboring vertices of V_1, namely V_0 and V_2 , to form a new triangle. When the angle is larger than 75°, but less than or equal to 135°, as Figure.3(b) depicts, a new vertex V_3 is inserted along the bisection line of vectors V_0 V_1 and V_1 V_2 to determine the new vertex. When the angle is larger than 135°, as Figure.3(c), two new vertices, V_3 and V_4 are inserted, equally distributed on two triple-section lines of vectors V_0 V_1 and V_1 V_2 .

Step 3: The distance between the new vertex and related boundary vertexes is calculated when the distance is less than given threshold, they should merge.

Step 4: Update the front and repeat the algorithm until the hole is patched with new triangles.

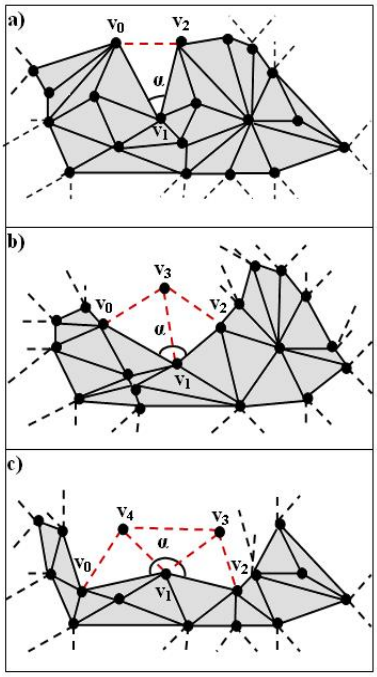


Figure 3: a) Rules for generating initial patch over the hole.

4.2 Harmonic-based desirable normal computing

The most important task of discrete harmonic function is to map a given disk-like surface $S_T \subset IR^3$ into the plane S^* which is introduced by Eck et al. [19] to computer graphic community. See Figure4.a.
Let S_T be in the form of a *triangular mesh* and V the set of vertices. If S_T has a boundary, then the boundary will be polygonal and we denote by V_B the set of vertices lying on the boundary and by V_I the set of interior vertices.
The goal is to find a suitable (polygonal) domain $S^* \subset IR^2$ and a suitable piecewise linear mapping $f : S_T \rightarrow S^*$ to minimize Dirichlet energy and it can determine by images $f(v) \in IR^2$ of vertices v. Overall, such mapping has two steps:

1- Find the boundary mapping, i.e.fix $f|_{\partial_{S_T}} = f_0$.

2- Find the piecewise linear mapping $f : S_T \longrightarrow S^*$ which minimizes the the Dirichlet energy for internal vertices.

$$E_D = \frac{1}{2}\int_{S_T} \|grad_{S_T} f\|^2 \qquad (1)$$

Subjected to the Dirichlet boundary condition $f|_{\partial_{S_T}} = f_0$.
A quadratic minimization problem and reduction to solve a linear system of equations are the main advantages of current method. Consider one triangle $T = \{v_1, v_2, v_3\}$ in the surface S_T. Referring to Figure4.b, one can show that

$$\begin{aligned} 2\int_T \|grad_T f\|^2 = &\cot\theta_3 \|f(v_1) - f(v_2)\|^2 \\ &+ \cot\theta_2 \|f(v_1) - f(v_3)\|^2 \\ &+ \cot\theta_3 \|f(v_2) - f(v_3)\|^2 \end{aligned} \qquad (2)$$

The normal equations for the minimization problem can be expressed as the following linear system of equations

$$\sum_{v_j \epsilon N_i} w_{i,j}(f(v_j) - f(v_i)) = 0 \qquad v_i \in V_I \tag{3}$$

Where

$$w_{i,j} = cot\alpha_{i,j} + cot\beta_{i,j} \tag{4}$$

and the angles are showed in Figure4.c. Here we have assumed the N_i denotes to 1-ring vertices of vertex v_i. The associated matrix is symmetric and positive and sparse, and so the linear system is uniquely solvable. The system can be solved efficiently with iterative methods such as conjugate gradient method. Note that system has to be solved three times. Once for x-, once for y- and once for z-coordinate. Now the desirable normal of all vertices in initial patch

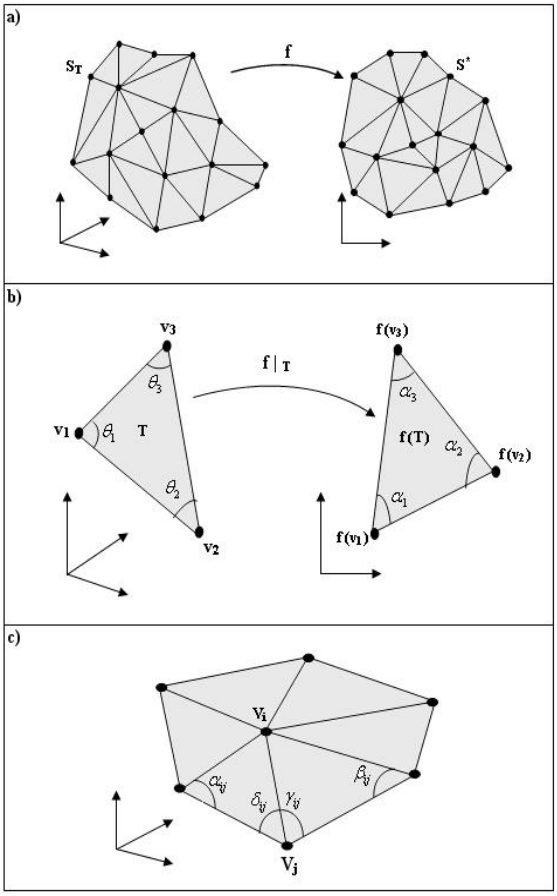


Figure 4: a) Piecewise linear mapping of a triangular mesh. b) Atomic map between a mesh triangle and the corresponding parameter triangle. c) 1-ring vertex of v_i and angles opposite to edge $v_i\ v_j$.

mesh is obtained. However, Poisson equation requires a discrete guidance field, i.e., w, defined on the triangles of the patch mesh. The guidance vector field is constructed by triangle rotation. Local rotation is applied to each triangle of initial patch mesh. Let n be the original normal of triangle and n' be the new normal of triangle that is calculated with desirable normal of vertices of triangle and c be the center of triangle. The rotation can obtained by rotating n to n' around c. See Figure5.
After rotation, the original patch mesh is torn a part and triangles are not connected any more and this torn triangles are used to construct a guidance vector filed for Poisson equation. Finally, the disconnected triangles are stitched by solving Poisson equation.

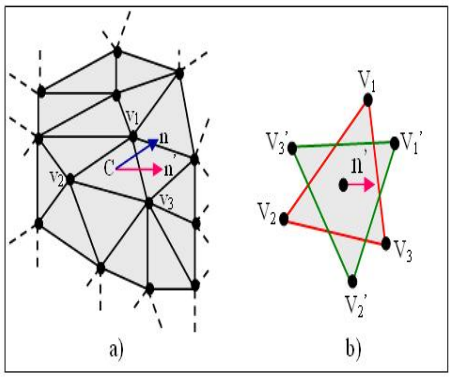


Figure 5: a) Initial patch mesh in triangle mesh, (b) A triangle red is initial patch mesh and its locally rotated is version green.

4.3 Poisson equation

In this section, we introduced the details of Poisson equation for reconstruction smooth and accurate patch mesh. We regard the mesh geometry (coordinates) as scalar function. The Poisson equation is originally appeared from [20]. The aim of this method is solving an unknown target mesh with known topology but unknown geometry (vertex coordinate). Poisson equation like Harmonic equation has to solved three time.
The Poisson equation with Dirichlet boundary condition [21], [22] is formulated as

$$\nabla^2 f = \nabla.w \quad over\ \Omega, \quad with\ f|_{\partial\Omega} = f^*|_{\partial\Omega} \tag{5}$$

where

- f is an unknown scalar function defined over interior of Ω.
- f^* is a known scalar function that provides the desirable values on the boundary $\partial\Omega$.
- $\nabla^2 = (\frac{\partial^2}{\partial x^2}, \frac{\partial^2}{\partial y^2}, \frac{\partial^2}{\partial z^2})$ is Laplacian operator.
- w is a Guidance Vector Filed and $\nabla.w = \frac{\partial w_x}{\partial x} + \frac{\partial w_y}{\partial y} + \frac{\partial w_z}{\partial z}$ is the divergence of $w = (w_x, w_y, w_z)$.

Thus it can be defined as least-squares minimization problem:

$$\min_f \int\int_\Omega |\nabla f - w|^2 \quad with \quad f|_{\partial\Omega} = f^*|_{\partial\Omega} \tag{6}$$

A discrete vector field on a triangle mesh is defined to be a piecewise constant vector function whose domain is the set of point on the mesh surface. A constant vector is defined for each triangle, and this vector is coplanar with the triangle. For discrete vector field w on a mesh , its divergence at vertex vi can be defined to be

$$(div\ w)(v_i) = \sum_{T_k \epsilon N_{(i)}} \nabla B_{ik}.w|T_k| \tag{7}$$

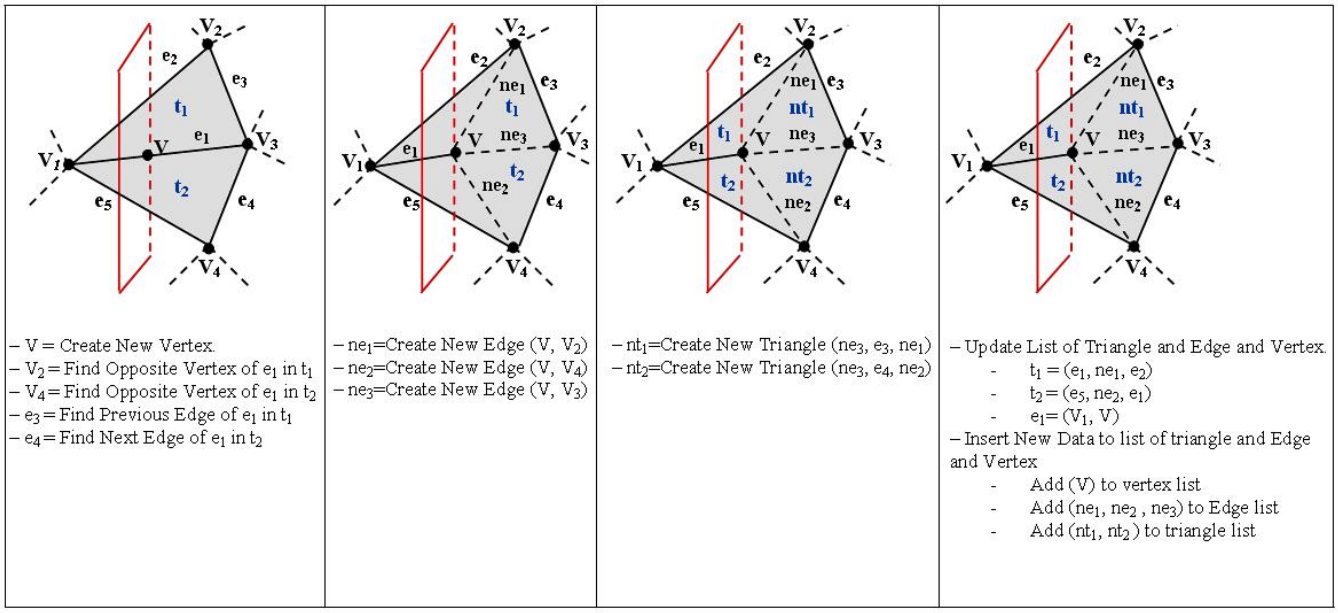


Figure 6: Steps for cutting shape to several sections

where N_i is the 1-ring vertices of v_i, $|T_k|$ is the area of triangle T_k, and ∇B_{ik} is the gradient vector of B_i within T_K.

The discrete gradient of the scalar function f on a discrete mesh is expressed as

$$\nabla f(v) = \sum_i f_i \nabla \phi_i(v) \tag{8}$$

with $\phi_i(.)$ begin the piecewise linear basis function valued 1 at vertex vi and 0 at all other vertices and f_i begins the value of f at v_i and it is one of the coordinate of v_i . The discrete Laplacian operator can determine as follow

$$\Delta f(v_i) = \frac{1}{2} \sum_{V_j \in N_i} (cot\alpha_{i,j} + cot\beta_{i,j})(f_i - fj) \tag{9}$$

where $\alpha_{i,j}$ and $\beta_{i,j}$ are the two angles opposite to edge in the two triangles sharing edge(v_i and v_j) and $N_i$is the set of the 1-ring vertexes of vertex v_i see Figure 4.c. Finally discrete Poisson equation is expressed as follows:$\nabla^2 f \equiv div(\bigtriangledown f) = \bigtriangledown w$
Discrete Poisson equation with Dirichlet boundary condition can be defined by the sparse linear system. It can be represented as the following form:

$$Ax = b \tag{10}$$

where the coefficients matrix A is determined by Eq.9 and the vector b is determined by Eq.7 and unknown vector x is the coordinate of all vertices on the patch mesh.
The smooth and accurate patch mesh is constructed as follow: *First*, Compute the gradient of each new vertex on the adjacent triangle by using Eq.8. *Next*, calculate the divergence of every boundary vertex by using Eq.7. *then*, determine the coefficient matrix A by Eq.9. vector b in this equation is determined by using divergence of all boundary vertices. *Finally*, solve the Poisson equation and obtain the new coordinate of all vertices of the patch mesh.

5. CUTTING 3D MODEL INTO SEVERAL SECTIONS

The input for the algorithm we developed is a triangle mesh file. The similarity search algorithm is based on the cutting foot triangle mesh into several sections towards the heel and the toe. Then the area of each section(available contour) is calculated and compared with the area of equal sections in shoe last data base.

5.1 Search for similarity estimation

The algorithm has the following steps summarized in Figure 6.

Step 1: Find the intersection of the cutting plane with the edge of a triangle and create new vertex.

Step 2: Choose the edge with the endpoints on the opposite sides of the intersection points.

Step 3: Find the next edge and the previous edge of the current edge.

Step 4: Build a new edge between the intersection point and the opposite vertices of the current edges triangles.

Step 5: Build a new triangle between the new edges.

Step 6: Update the triangles and vertices of the current edge.

Step 7: Add the new triangle and the edge to the list.

The output of our algorithm is set of vertices and edges related to each contour as illustrated in Figure 7. Let $V_c = \{v_1, ..., v_n\}(v_i = (x_i, y_i, z_i) \in R_3)$ be a set of "vertices" and $E_c = \{e_1, ..., e_n\}$ the set of "edges" associated to the contours.

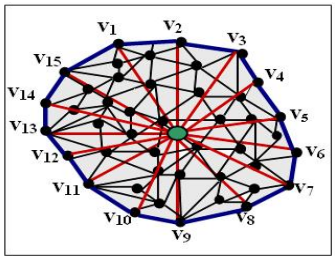


Figure 7: The illustrated contour after cutting mesh.

Step 1: : Let c_c be center of gravity of contour determined as follow:

$$c_c = \frac{\sum_{i=0}^{N} V c_i}{N} \quad (c_c = (x_i, y_i, z_i) \in R_3) \tag{11}$$

Step 2: For calculating the area of each contour. We divide the vertices of contours edge and center of gravity of contour in triangles. Let A_i be the area of each associated triangle in the contour and N be the number of triangles that associated with edges and center of gravity of contour. The area of each contour can be calculated as follow:

$$A = \sum_{i=0}^{N} A_i \tag{12}$$

6. CONCLUSION

In this paper, a new approach for evaluating footwear fit within the shoe last data base is proposed. This new approach should clearly help to improve the users comfort and it could be the starting point for mass customization approach in footwear design.
Our method is based on an efficient algorithm to cut the model to several sections toward the heel and toe for extracting contours. Then the area of each contour is calculated and compared with equal sections in shoe last data base for finding best footwear fit. However, there is a need of post-processing step for filling hole in 3D models before taking these models in comparisation step. Thus, the advance front mesh technique is used to generate initial patch mesh. Then the desirable normal based on Harmonic function is calculated for modifying of initial patch mesh. Finally accurate and smooth triangle mesh are obtained by solving the Poisson equation according to desirable normals and the boundary vertices of the hole.
As a future work and development, the specific sections of the foot will be compared to corresponding sections of shoe latst's data base, so that new shoe lasts will be designed in such a way that they fit costumers feet completely.

7. REFERENCES

[1] K.Parker, "Settling footwear complaints," *(Northants: SATRA Technology Centre)*, 1996.

[2] M.P.Cheskin, "The complete handbook of athletic footwear," *New York: Fairchild Publications*, 1987.

[3] W.A.Rossi, "The futile search for perfect shoe fit," *Journal of testing and Evaluating*, , no. 16, pp. 393–403, 1988.

[4] H.Hu, Z.Li, J.Yan, X.Wang, H.Xiao, J.Duan, and L.Zheng, "Anthropometric measurement of the chinese elderly living in the beijing area," *International Journal of Industrial Ergonomics*, vol. 37(4), no. 16, pp. 303–311, 2007.

[5] Ritz, "http://www.ritzstick.com/," .

[6] Brannock, "http://www.brannock.com/," 2006.

[7] M.R.Hawes and D.Sovak, "Quantitative morphology of the human foot in a north american population," *Ergonomics 37*, pp. 1213–1226, 1994.

[8] G.Li and A.Jneja, "A morphing-based surface blending operator for footwear cad," *Proceeding of International Mechanical Engineering Congress and Exposition*, 2004.

[9] M.Mochimaru, M.Kouchi, and M.Dohi, "Analysis of 3d human foot forms using the free form deformation method and its application in grading shoe lasts," *Ergonomics*, vol. 43, no. 9, pp. 1301–1313, 2000.

[10] K.Kimk, K.lee, and T.Hwang, "A grouping algorithm for custom-tailored products," *Journal of Materials Processing Technology*, pp. 618–625, 2002.

[11] R.S.Goonetilleke A.Luximon and K.L.Tsui, "A fit metric for footwear customization," *Proceeding of the World Congress on Mass Customization and Personalization*, 2001.

[12] B.Curless and M.levoy, "A volumetric method for building complex models from range images," *In: Processing of siggraph*, pp. 303–12, 1996.

[13] J.Davis, SR.Marschner, M.Garr, and M. Levoy, "Filling holes in complex surfaces using volumetric diffusion," *In: Processing of the first international symposium on 3D data, visualization and transmission*, 2002.

[14] FS.Nooruddin and G.Turk, "Simplification and repair of polygonal models using volumetric techniques," *IEEE Transactions on Visualization and Computer Graphics*, vol. 9(2), pp. 191–205, 2003.

[15] P.Liepa, "Filling holes in meshes," *In: Proceedings of the eurographics/ACM SIGGRAPH symposium on geometry processing*, vol. 23, pp. 200–5, 2003.

[16] L.Kobbelt, S.Campagna, J.Vorsatz, and H-P.Seidel, "Interactive multiresolution modeling on arbitrary meshes," *In: Proceedings of SIGGRAPH*, pp. 105–14, 1998.

[17] W.Zhao, S.Gao, and H.lin, "A robust hole-filling algorithm for triangular mesh," *Visual Comput*, vol. 23, pp. 987–997, 2007.

[18] L.P.George and E.Seveno, "The advancing-front mesh generation method revisited," *Int. J. Numer. Methods Eng*, vol. 37(7), pp. 36053619, 1994.

[19] M.Eck, T.D.DeRose, T. Kuchamp, H.Hoope, M.Lounsbery, and W.Stuetzle, "Multi-resolution analysis of arbitrary meshes," *In:Processing of SIGGRAPH, LosAngles, CA, USA*, vol. 9(6), no. 6-11, pp. 173–182, 1995.

[20] J.Tohline, "Origin of the poisson equation," *http://www.phys.lsu.edu/astro/$H_B$ook.current/Context/PGE/poisson.origin.text.pdf*.

[21] P.Perez, P.Gangnet, and A.Blake, "Poisson image editing," *In: Processing of SIGGRAPH, San Diego, California, USA*, pp. 313–318, ACM Press 2003.

[22] X.Shi H.Guo B.Shum Y.Yu, K.Zhou, "Mesh editing with poisson-based gradient field manipulation," *In PROCEEDINGS of Siggraph, Los Angles, California, USA*, vol. 2004, pp. 644–651, ACM Press 2003.

ABOUT THE AUTHORS

Bita Ture Savadkoohi is a Ph.D. student at the University of Trento, Italy in the GeraphiTech research division, and Her contact email is bsavadk@dit.unitn.it.

Raffaele de Amicis is the Director, he holds a MEng in Mechanical Engineering, a Ph.D. on Surface Modeling in Virtual Environments. He has been research fellow at the Industrial Applications Department of Fraunhofer Institute, Darmstadt and senior researcher at the Interactive Graphics Systems Group, University of Darmstadt. He has been involved in several EU and Industrial projects. His research interests are in CAD, virtual reality, Virtual engineering, Geovisual analytics and science and technology policy. He authored or contributed to 5 books and authored around 100 papers. His contact email is raffaele.de.amicis@graphitech.it.

Surface Reconstruction: An Improved Marching Triangle Algorithm for Scalar and Vector Implicit Field Representations

Marc Fournier
Quebec University in Montreal
CP 8888, Succ. Downtown, Montreal, Canada H3C 3P8
fournier.marc@uqam.ca

Abstract

In this paper we propose a new polygonization method based on the classic Marching Triangle algorithm. It is an improved and efficient version of the basic algorithm which produces a complete mesh without any cracks. Our method is useful in the surface reconstruction process of digitized objects. It works over the discrete distance transform of the object to produce the resulting triangle mesh. The new algorithm is also adapted to a recently introduced vector field distance transform model which is more accurate than the classic scalar field discrete distance transform of meshes. Our polygonization method is simplified and it produces better results compared to Marching Triangle basic algorithm while working on the vector field distance transform model. We use relevant error metric tools to compare results and show our new method is more accurate than Marching Cube which is the most widely used triangulation algorithm in the surface reconstruction process of digitized objects.

Keywords: *Marching Triangle, Discrete distance transform, Polygonization algorithm, Surface reconstruction, Digitized objects, Triangle mesh surface.*

1. INTRODUCTION

In the past decade a lot of improvements have been made in the field of 3D scanners to acquire and to digitize real world objects. The advancement in computer technologies have made possible the design of 3D scanners to respond to the increasing needs in many fields such as digitizing precious cultural heritage [1]. The most common output data structure produced by 3D scanners is range images. From these raw data, many operations are needed in order to produce the final mesh which represents the real object geometry. All these operations can be achieved in the scalar field distance transform (SFDT) domain which is often used because it produces good results for each step of the reconstruction procedure.

To perform surface reconstruction in the SFDT domain, one needs to convert the explicit range images, or other initial triangle meshes dataset, by computing their SFDT discrete implicit field which is defined over a regular 3D grid created inside a mesh bounding box. The cubic grid cells are called voxels and for each voxel, the closest point on the mesh surface is found and the shortest distance to the mesh is saved in the voxel. For a given surface $S \subset \Re^3$ this volume representation consist of a scalar value function $f: \Re^3 \rightarrow \Re$ such as the zero-set $f(x,y,z)=0$ defines the surface and in that case $[x,y,z] \in S$. To obtain a unique volumetric description for a given surface, this distance field is also signed according to surface normal vectors.

The surface reconstruction process begins with a mesh registration procedure [2] which is performed to express all range images in the same coordinates system. The first operation in the SFDT domain is mesh fusion [3] to integrate all initial range images into a unique representation, followed by mesh repair [4] to fill holes in the model, and then mesh smoothing [5] to remove acquisition noise introduced by the scanner and finally mesh simplification [6] to produce a more compact model without loss of details. Since the SFDT is an implicit representation, at the end of the reconstruction process a polygonization algorithm such as Marching Cube [7] or Marching Triangle [8] is needed to produce the final explicit mesh which describes the digitized object surface geometry.

In this paper we introduce a new polygonization method to triangulate the resulting mesh of the surface reconstruction process in the SFDT domain. We propose a set of improvements based on the Marching Triangle algorithm to obtain an efficient method which overcomes some crack problems in the basic version to produce a higher quality resulting mesh. We also adapt our method to the vector field distance transform (VFDT) implicit representation which allows a simplification and a more accurate result using the Marching Triangle algorithm. The remaining parts of the paper are organized as follows: Section 2 overviews related work. Section 3 presents our new improved triangulation algorithm. Section 4 describes the Marching Triangle adaptation to the VFDT representation. And in Section 5, before concluding, we show and compare our triangulation method results with previously introduced algorithms.

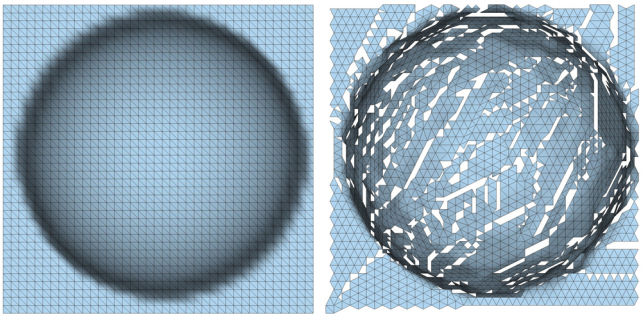
2. RELATED WORK

The most widely used algorithm to triangulate a SFDT is Marching Cube [7]. It is a volume-based approach which is very suitable to triangulate discrete implicit fields such as SFDT in the surface reconstruction process of digitized objects. The original Marching Cube method has some ambiguities and many other algorithms, based on the original one, have been proposed to improve the resulting mesh quality. For example the original algorithm has 14 cube triangulation configurations which lead to face ambiguities resolved in [9], and [10] in which the 14 basic configurations are expended in 32 different cases. A remaining cube ambiguity was then solved in [11] to guaranty the resulting mesh topology.

Methods derived from the basic Marching Cube algorithm have been proposed to also resolve the ambiguities. Cubical Marching Squares [12] which opens the cubes into six squares and Marching Tetrahedron [13] which divides the cubes into tetrahedrons are two examples of cube configurations which resolve original ambiguities. Other algorithms have been introduced to improve the resulting mesh quality. The Extended Marching Cube [14] provides a sensitivity feature to better recover sharp edges and Dual Contouring [15] focus on preserving the resulting mesh topology. More recently, the VFDT

model [16] which is a vector extension of the SFDT has been proposed to improve the implicit field accuracy and the Marching Cube algorithm has been adapted to this new representation in order to produce a higher quality resulting mesh.

Another well known algorithm to triangulate a SFDT is Marching Triangle [8]. It is a surface-based approach built on Delaunay triangulation definition. Starting from a seed triangle, a region-growing process enables the creation of triangles following the SFDT isosurface. This surface tracking process has been designed to triangulate discrete SFDT of digitized objects and it has also been applied to continuous implicit surfaces describing virtual objects with a set of equations such as parametric ones. The basic Marching Triangle algorithm leaves some part of the model un-triangulated creating cracks in the resulting mesh as shown in Figure 1.



a) Original model b) Marching Triangle result

Figure 1: Basic Marching Triangle algorithm result on the Half-Sphere model

In Figure 1a) a Half-Sphere model lying on a plane is used and its SFDT is computed. Figure 1b) shows the basic Marching Triangle algorithm result which contains cracks in the triangulated resulting mesh. Theses cracks occur because of different reasons which will be detailed in next section.

Beside the classic Marching Triangle, other methods based on a region-growing and surface tracking process have been proposed such as Gopi algorithm [17] which works over a set of unorganized points and Hartmann algorithm [18] which works on continuous implicit surfaces. In this paper we focus on improving the steps of the original Marching Triangle to directly triangulate discrete SFDT of digitized objects. The methods designed to work over point clouds are also suitable for discrete SFDT as described in the Ball-Pivoting method [19] in which a first pass generate points from the SFDT before applying the triangulation on the set of generated points. This Ball-Pivoting method is similar to Marching Triangle, it also uses the Delaunay sphere test but the constraint is applied on the sphere radius instead of the sphere center as for the original Marching Triangle algorithm. In this paper we also modify the Delaunay sphere for the triangulating test which contributes to reduce the resulting mesh cracks.

Previous methods [20, 21, 22] applied to continuous implicit surfaces have been introduced to overcome the basic Marching Triangle algorithm cracks problem. These method triangulations are adaptive to local implicit curvature to obtain variable size triangles. In this paper we also modify the basic algorithm with a variable projection distance to obtain a better adaptive result over discrete SFDT. Some of these methods [20, 21] operate in two passes. They first triangulate a resulting mesh according to the basic algorithm which contains cracks and then they introduce a crack filling algorithm to complete the resulting mesh. The other method [22] introduces a different region-growing algorithm compared to the original Marching Triangle. It is based on a hexagonal triangulation expansion pattern which is able to resolve cracks and to produce a complete resulting mesh in a single pass. In this paper we use the original Marching Triangle algorithm procedure and we propose improvements to several steps of the method. Our new triangulation works over a discrete SFDT in a single pass, as for the original algorithm, and produce a complete mesh without cracks. We do not need a second specific post-processing crack filling pass but instead we introduce an iterative process on our single pass to obtain a complete mesh.

Other algorithms have been proposed to improve the basic Marching Triangle triangulation over discrete implicit surfaces and to achieve specific goals. A topology preserving method [23] introduces a normal consistency constraint which guaranties the resulting mesh topology. An edge constrained method [24] detects discontinuities in the implicit surface and constrains triangle edges to match and better preserve these sharp features in the resulting mesh. In this paper we have a more global approach to produce a higher quality mesh by improving the entire original Marching Triangle algorithm. We also adapt our new triangulation method to the VFDT model [16] which is more accurate than the SFDT. Therefore we obtain a more globally accurate resulting mesh including a better sharp features preserving compared to the original Marching Triangle algorithm.

3. MARCHING TRIANGLE IMPROVED ALGORITHM

In this section we describe our new Marching Triangle improved algorithm. We refer to the original algorithm [8] procedure numbering to identify the steps to improve. In Subsection 3.1 we improve steps 1 and 2 of the original algorithm in finding a new vertex. In Subsection 3.2 we improve step 4 in testing a new triangle. In Subsection 3.3 we improve step 6 in considering new triangles. Finally in Subsection 3.4 we introduce an edge processing sequence to improve the overall algorithm.

3.1 Variable Projection Distance and Vertex Interpolation

The first and second steps of Marching Triangle algorithm are illustrated in Figure 2. The first step is the estimation of a new vertex position P' with a projection perpendicular to the mid-point P of the boundary edge C in the plane of the model boundary triangle ABC by a constant distance PP'. The second step is the evaluation of the nearest point to P' which is on the implicit surface. That new potential vertex is V2 in Figure 2 example.

The constant distance projection contributes to produce cracks in the mesh. In high curvature regions the projection can be further away from the isosurface and the new vertex estimation may cause either a test failure resulting in a crack beginning or a bad approximation of the surface local geometry. To correct this problem we propose a variable projection distance which is equal to $\sqrt{3}/2$ times the length of the boundary edge. This projection distance corresponds to the height of the equilateral triangle composed of the boundary edge. This improvement contributes to obtain more equilateral triangles which are less deformed and which sizes adapt gradually to local geometry curvature.

From the projection point P' on Figure 2 example, we suppose the nearest voxel which is considered on the isosurface is V2. While working with a discrete SFDT a threshold distance comparison is needed to find this nearest voxel. To consider that a voxel is on the isosurface it must contain a distance smaller than half the grid resolution. This approximation introduces a significant error in the vertex position according to the underlying implicit surface and this error can also contributes to a crack in the mesh if the new triangle test fails. To correct this problem we propose a linear vertex position interpolation between the nearest voxel found and its closest neighbor with opposite distance sign. The interpolation finds the new vertex position which corresponds to a distance equal to zero between the two distances of opposite sign. This interpolation uses the implicit surface definition to obtain a more accurate approximation of the isosurface.

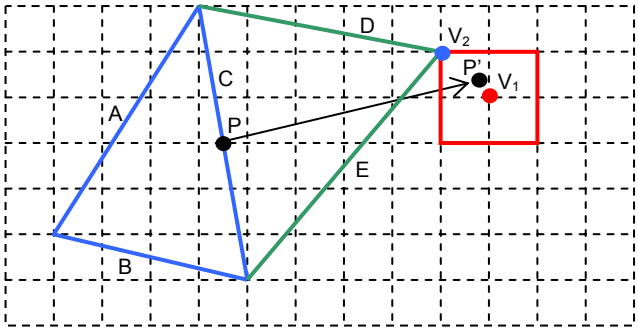
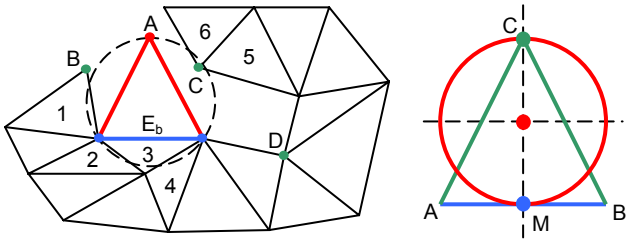


Figure 2: Marching Triangle projection and nearest vertex on the isosurface

3.2 Modified Delaunay Sphere Test

Step 4 of the original algorithm suggests testing the new potential triangle according to the Delaunay sphere which is circumscribed to the new triangle as shown in Figure 3a) in which the current boundary edge E_b forms a new triangle with the new estimated vertex A. In Figure 3a) example the test would fail because there are parts of the six numbered triangles inside the sphere. According to the original algorithm, further tests would be made with edge E_b and vertices B, C and D to consider these three new triangles. These tests would also fail because the Delaunay sphere would still contain parts of neighbor triangles. This test limits the original algorithm performances and contributes to produce cracks in the mesh. The Delaunay test sphere was designed to triangulate a set of unorganized points which is not exactly the same situation in the case of a region-growing algorithm over a SFDT.

We propose the modified sphere shown in Figure 3b) for new triangles testing. The modified sphere passes through the mid-point M of the current boundary edge and the new estimated vertex C. Its diameter is equal to the distance between these two points and its center is the mid-point between M and C. The modified sphere is smaller than the original one and it allows obtaining more successful tests which improve the algorithm and reduce the cracks in the resulting mesh. Since some parts of the new triangle are outside the sphere, we also need to test if there is no intersection between the new triangle and other triangles of the mesh. We test all triangles which have parts inside the original Delaunay sphere with the new triangle according to Moller intersection test [25] which is fast and efficient.



a) Interfering triangles with the original sphere b) Modified test sphere

Figure 3: Interfering triangles with the original Delaunay sphere and modified test sphere

3.3 New Triangles to Consider

Step 6 of the original algorithm suggests considering two new potential triangles illustrated in Figure 3a) if the first new triangle test fails at step 4. These new triangles are composed of the current boundary edge E_b and vertices B or D which are the previous and the next vertices along the boundary from the current boundary edge E_b. If the sphere test fails for these two new triangles, other new triangles should be tested to improve the algorithm. The original algorithm was upgraded in [26] with a seventh step which suggests to test another new triangle composed of the current boundary edge E_b and vertex C in reference of Figure 3a). Vertex C is the nearest boundary vertex of overlapping triangles number 5 and 6 in the sphere test. This new potential triangle contributes to reduce the cracks in the mesh compared to the original algorithm but if the test fails with this triangle other triangles should be tested to better improve the algorithm.

We propose to test not only the nearest but all boundary vertices of overlapping triangles if they exist. In some particular cases as the one shown in Figure 4 the sphere test would fail with the nearest boundary vertex leaving a part of the mesh un-triangulated and a new triangle could be added to the mesh if another boundary vertex of the overlapping triangle was considered. In Figure 4 E_b is the current boundary edge and V_1 is the triangle nearest vertex from E_b. The new potential triangle composed of E_b and V_1 would fail the test but another triangle composed of E_b and V_2 would be a better candidate even if the distance between E_b and V_2 is greater than the distance between E_b and V_1.

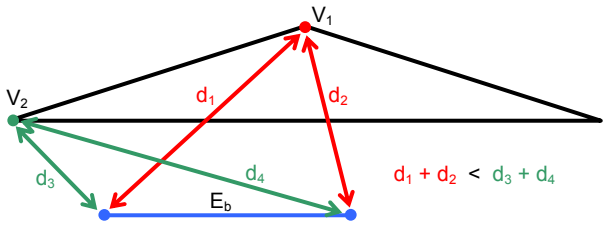
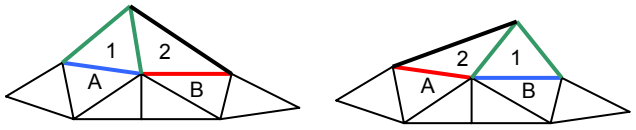


Figure 4: New triangles to consider in the triangulation process

3.4 Edge Processing Sequence and Iterative Process

The original Marching Triangle algorithm does not specify any boundary edges processing sequence. It is only defined as a single pass into the edge list to process all boundary edges with the procedure steps including the estimation of a new potential triangle and its sphere test to determine if it will be added or not to the mesh. According to the implemented data structure to triangulate the SFDT and the method to add new edges in the edge list, the edge processing sequence can be different from one implementation to another. The resulting mesh from the Marching Triangle depends on the edge processing sequence and it can be different if the sequence is changed as shown in Figure 5.



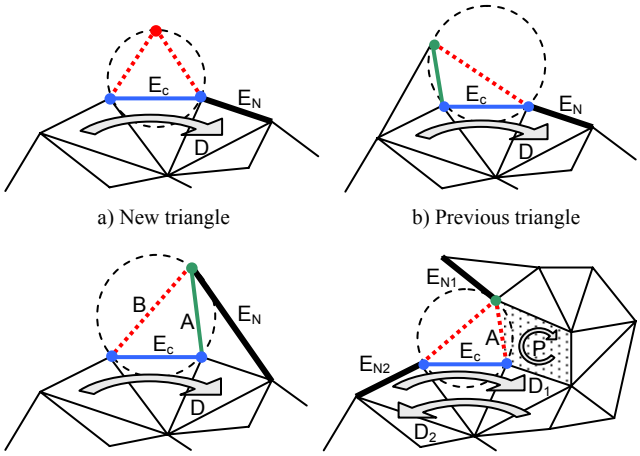
a) First possible mesh b) Second possible mesh

Figure 5: Modified mesh according to the edge processing sequence

In Figure 5 example we assume the current mesh is composed of the six bottom triangles, therefore edges A and B are boundary edges. In Figure 5a) edge A is considered first and triangle 1 is added then edge B is processed and triangle 2 is added next. In Figure 5b) edge B is considered first and triangle 1 is added first then edge A is tested and triangle 2 is also added. In Figure 5 simple example both results are different just because of an edge permutation. The results would have been more different if for example after adding triangle 1 in Figure 5a) the new boundary edges of that triangle were processed before edge B and if that same processing sequence was applied in Figure 5b) to the new boundary edges of triangle 1.

We tested different edge sequences and combinations on the original algorithm and we selected the one which optimizes the result in terms of minimizing the cracks in the mesh. We propose the following procedure to improve the resulting mesh quality. Starting from a boundary edge, an arbitrary direction is selected and the next edge to consider is the neighbor boundary edge always in that same direction around the contour of the current mesh. Newly added boundary edges are not considered immediately, they will only be considered on the next turn around. This procedure is illustrated in Figure 6 with E_c corresponding to the current boundary edge to be processed and E_N the next boundary edge to consider according to the chosen arrow direction D.

In Figure 6a) New triangle and Figure 6b) Previous triangle cases the next boundary edge to consider is straight forward. In Figure 6c) Next triangle case if the test is successful and the new triangle is added then the next boundary edge to consider jumps over edge A which is lo longer a boundary edge. The new added boundary edge B will be considered on the next turn around only. In Figure 6d) Overlapping triangle case a bridge is created in the mesh if the new triangle is added. In that case the next edge to consider is either E_{N1} or E_{N2} depending on the chosen direction D_1 or D_2 respectively. In that special case the inside contour in the region P is triangulated in priority immediately after the new triangle is added to the mesh and before pursuing with the outside contour. Starting from the new inside boundary edge A an arbitrary direction is selected and the previously described procedure is applied to the inside contour until no more triangle can be added.



a) New triangle b) Previous triangle

c) Next triangle d) Overlapping triangle

Figure 6: Next edge to consider according to the edge processing sequence

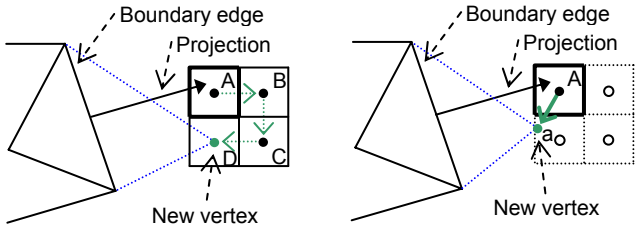
The original Marching Triangle is defined as a single pass into the edge list. Our proposed procedure is iterative and the boundary edges are tested more than once, until a complete loop is made around the mesh contour without adding any triangle. A boundary edge can be tested without adding any triangle to it at the first pass and at the second pass the test could be successful depending on the local mesh neighbourhood configuration which can be different from one pass to another. Our iterative procedure continues as long as new triangles are added. Compared to the original algorithm our procedure contributes to add more triangles and to reduce the cracks in the final resulting mesh.

4. VECTOR FIELD AND ERROR METRIC ADAPTATION

We also adapt our improved Marching Triangle algorithm as defined in Section 3 to the VFDT [16]. The discrete vector field distance transform is an extension of the SFDT. Instead of saving in each voxel only the scalar shortest distance to the surface as in the SFDT, in the VFDT a vector is saved in each voxel. This vector gives the shortest distance to the surface and in addition it also gives the orientation of the closest point on the surface. This new representation is more accurate than the classic SFDT and it is used in the surface reconstruction process of digitized objects. Marching Cube algorithm was previously adapted to the VFDT and this adaptation produces better triangulation results than other Marching Cube versions over the SFDT. Our Marching Triangle adaptation to the VFDT is simple and it has two advantages. First it saves computation time and second it also produces more accurate results compared to the scalar version over the SFDT. Figure 7 illustrates our Marching Triangle adaptation to the VFDT.

Figure 7 shows steps 1 and 2 of the Marching Triangle algorithm with a) the SFDT and b) the VFDT. Step 1 is the same for both representations; a projection is made from the boundary edge

mid-point to estimate a new vertex position. Step 2 is different and a search is needed with the SFDT to find the nearest voxel which is considered on the isosurface. In Figure 7a) example the search stopped on voxel D which is the nearest voxel from the projection point to contain a distance smaller than half the grid resolution. In the original Marching Triangle algorithm, voxel D coordinates are used as the new vertex which forms the new potential triangle to be tested. This is a coarse approximation of the isosurface. In Section 3.1 of this paper we propose a linear interpolation to obtain a better estimation of the isosurface. This is an improvement from the original algorithm but the result still is an approximation. In Figure 7b) there is no need for a search and interpolation with the VFDT. From the projection point which is in voxel A, the vector saved in that voxel is used and a simple redirection gives immediately the coordinates of the new vertex. According to the VFDT definition, this new vertex is the nearest point from the projection point which is exactly on the isosurface. With the VFDT, step 2 of the Marching Triangle is simplified and upgraded to a better result. The algorithm other steps are exactly the same as for the SFDT.



a) SFDT search b) VFDT redirection

Figure 7: Comparison of the new vertex estimation with the SFDT and the VFDT

In order to evaluate and compare our improved algorithm on the SFDT, we use the vertex to surface error metric defined in [5] to quantify the relative quality of the triangulation algorithms tested. This error metric is based on the minimal Hausdorff distance between two meshes. We start from a reference mesh and compute its SFDT. Then we triangulate the SFDT with different algorithms. The resulting meshes are compared to the reference mesh using this vertex to surface error metric which gives a scalar value of the average distance or error between two meshes. The smallest distance from each vertex of the triangulated result to the surface of the reference mesh is evaluated and a weighted average of these distances gives the associated error of the triangulation result compared to the reference mesh. Then the triangulation result errors of different algorithms from the same reference mesh can be compared together to evaluate the relative quality of each result.

The vertex to surface error metric is not suitable to evaluate and compare our adapted algorithm to the VFDT because the resulting triangulation vertices are exactly on the reference mesh surface. Using the vertex to surface error metric with our adaptation would produce an error equal to zero even if our result and the reference mesh are different. In order to evaluate and compare our VFDT adaptation, we define a triangle to surface error metric which is based on the previous vertex to surface error metric. The difference is instead of computing the distance from the resulting mesh vertices to the reference mesh surface, we compute the distance from each triangle centroid of the resulting mesh to the reference mesh surface. Our triangle to surface error metric is defined by:

$$\varepsilon_t = \sqrt{\frac{1}{A(M')} \sum_{i=1}^{N} A(t'_i)\, dist(ct'_i, M)^2} \qquad (1)$$

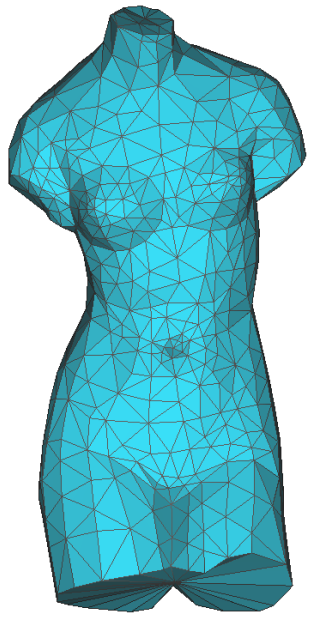
M is the initial reference mesh and M' is the triangulated result to evaluate. A(M') is the total area of M'. $A(t'_i)$ is the area of each triangle t'_i of the resulting mesh. ct'_i are the resulting mesh triangles centroid. $dist(ct'_i, M)$ is the minimal distance between the centroid ct'_i and the initial reference mesh M. This triangle to surface error metric is used to compare all results when our Marching Triangle VFDT adaptation is evaluated.

5. RESULTS

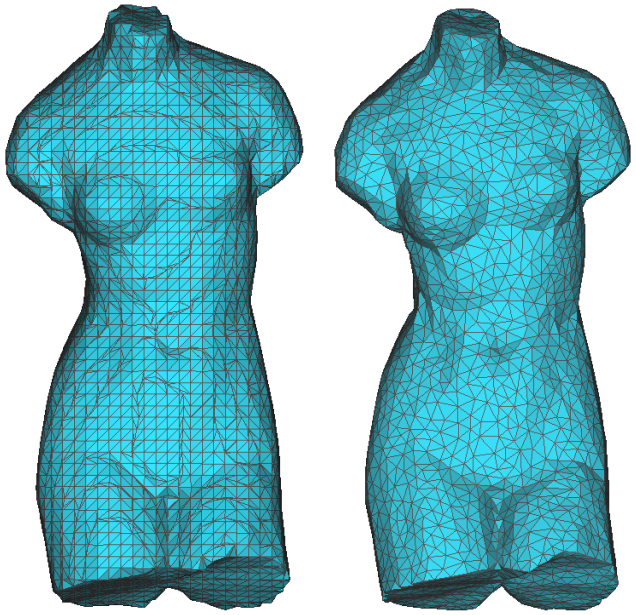
In this section we evaluate our improved algorithm on both SFDT and VFDT representations and we compare our results with Marching Cube triangulation using the vertex to surface and the triangle to surface error metrics with the procedure described in the previous section. Regarding the Marching Cube results, we implemented the Cubical Marching Squares [12] and the Extended Marching Cube [14] algorithms which are two recent and efficient versions and we kept the one showing the best result for each model tested.

First we used the Venus model reference mesh shown in Figure 8a) to compute its SFDT with an appropriate grid resolution according to the model level of details. Then we triangulated the SFDT with Marching Cube and our improved Marching Triangle algorithms and these results are shown in Figure 8. We compared these two results to the reference mesh using the vertex to surface error metric and Table 1 shows these error values along with the number of triangles and the triangulation computing times for both results. The timing measures were made using a Pentium 4 CPU computer with a 3.03GHz clock.

In Figure 8 both results are of good quality but we see that the triangles of Marching Cube result are more dependent on the voxels size. Figure 8b) result also shows small degenerated triangles forming elevation lines depending on the grid resolution which is the classic signature of Marching Cube algorithm. Marching Triangle result in Figure 8c) shows a more homogeneous triangulation and sharp edges such as the one at the bottom are better preserved compared to Marching Cube result. Table 1 also shows that Marching Triangle result is of better quality according to the error metric. Our result also contains fewer triangles, thus optimizing the model quality, storage and memory space with almost one third less triangles than Marching Cube result. The drawback of our algorithm is the computation time which is almost the double compared to Marching Cube. The triangulation time is still reasonable for this model but it could make a difference for example in real time applications on large models.



a) Reference mesh



b) Marching Cube c) Marching Triangle

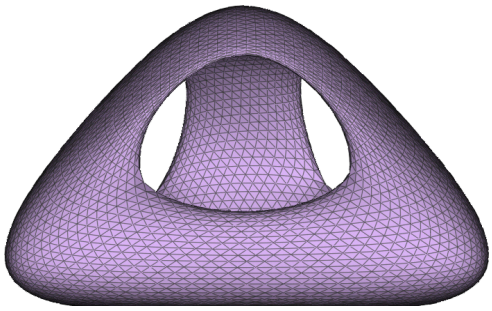
Figure 8: Triangulation results on the Venus model

Table 1: Triangulation parameters for the Venus model

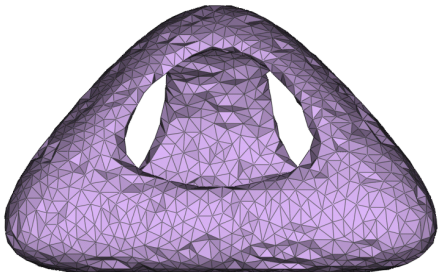
Parameters	Marching Cube	Marching Triangle	Difference
Nb Triangle	7 465	5 152	31.0%
Error	7.58×10^{-3}	6.88×10^{-3}	9.23%
Time (ms)	26.7	48.5	81.6%

We used a Genus3 virtual model to compare our algorithm adaptation to the VFDT and results are shown in Figure 9. For this evaluation a coarse grid resolution was used to highlight the advantage of the vector field adaptation. Starting with the initial model in Figure 9a) we computed both SFDT and VFDT of the model at same resolution. Then we triangulated both implicit representations with our Marching Triangle improved algorithm. Results in Figure 9b) and 9c) were compared to the initial model using our triangle to surface error metric definition and Table 2 shows the results for this model.

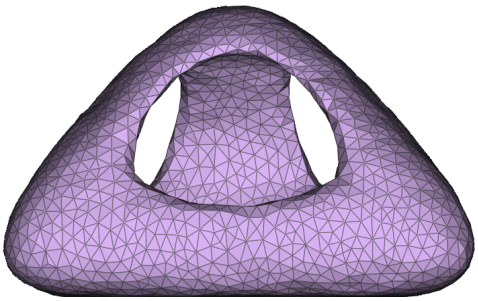
In Figure 9 we see that our VFDT adaptation is of better quality compared to the algorithm applied to the SFDT. Using the exact isosurface point at step 2 in the VFDT adaptation produces a more accurate result than the approximation of the isosurface with the SFDT. Table 2 error metric values show that our adaptation takes advantage of the VFDT improved representation to obtain a better result. The timings in Table 2 show that the simplification of step 2 in the algorithm adaptation to the VFDT is faster than with the SFDT but the difference is not significant compared to the overall algorithm timing. The resulting meshes numbers of triangles are very similar with both representations; our VFDT adaptation produced a few more triangles compared to the SFDT version.



a) Reference mesh



b) SFDT triangulation



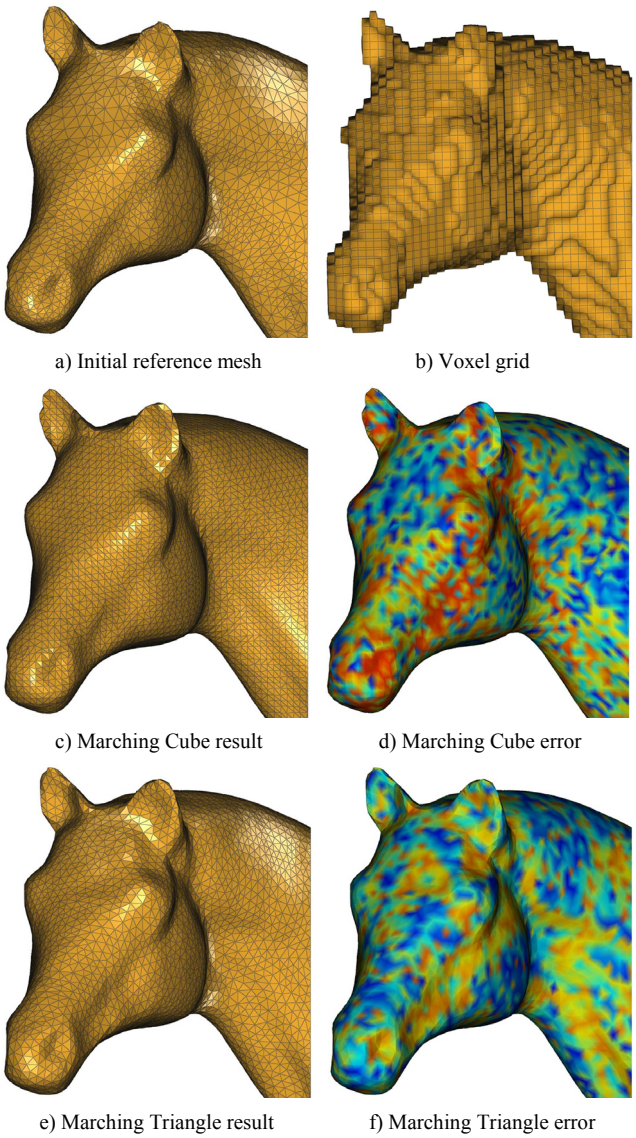
c) VFDT triangulation

Figure 9: Triangulation results on the Genus3 model

Table 2: Triangulation parameters for the Genus3 model

Parameters	SFDT Imp. Rep.	VFDT Imp. Rep.	Difference
Nb Triangle	3 481	3 615	3.71%
Error	$6.14x10^{-2}$	$5.63x10^{-2}$	8.31%
Time (ms)	33.6	32.7	2,68%

The Horse model shown in Figure 10 was used to compare previously introduced Marching Cube adaptation to the VFDT and our Marching Triangle adaptation to the VFDT. The voxel grid size shown in Figure 10b) was used to compute the VFDT of the initial mesh in Figure 10a). Figure 10c) and 10e) show both triangulation results over the VFDT.



a) Initial reference mesh b) Voxel grid

c) Marching Cube result d) Marching Cube error

e) Marching Triangle result f) Marching Triangle error

Figure 10: Triangulation results on the Horse model

The triangle to surface error metric was used to compare Figure 10 triangulations to the initial mesh and results are listed in Table 3 along with the other relevant parameters for this model. The local triangle to surface error metric computed at each triangle centroid has been converted into an error colormap in which red corresponds to the greatest and blue to the lowest error. These two colormaps are shown in Figure 10d) and 10f) for both triangulations.

Figure 10c) shows that Marching Cube adaptation to the VFDT is of better quality than the algorithm applied to the SFDT. The result contains no more small degenerated triangle. But the triangle sizes are still very dependent on the grid resolution and Marching Cube result still contains significantly more triangles compared to our Marching Triangle adaptation. Marching Cube adaptation to the VFDT provides a better approximation than the SFDT version but vertices positions are not exactly on the isosurface, they still are approximations. Therefore our Marching Triangle adaptation which takes fully advantage of the VFDT representation with exact vertices positions on the isosurface produces a more accurate result according to the error metric in Table 3. Visually in Figure 10 we also see that Marching Triangle colormap contains less red color errors compared to Marching Cube colormap. Even if our algorithm processing time is a bit faster over the VFDT compared to the SFDT, it is still much slower than Marching Cube adaptation to the VFDT.

Table 3: Triangulation parameters for the Horse model

Parameters	Marching Cube	Marching Triangle	Difference
Nb Triangle	57 251	46 716	18.4%
Error	$3.40x10^{-3}$	$3.19x10^{-3}$	6.18%
Time (ms)	245.4	429.7	75.1%

6. CONCLUSION AND FUTURE WORK

In this paper we designed a new Marching Triangle algorithm to improve the triangulation result of such method over the SFDT of digitized objects in their surface reconstruction process. Our contribution focused on improving several steps of the original algorithm to overcome the crack forming problem. We proposed a variable projection distance and a vertex position interpolation to provide a better isosurface approximation. We introduced a modified sphere for testing potential triangles geometry consistency before adding them to the resulting mesh. We also proposed testing new potential triangles which can lead to more complete results in particular cases. We structured an edge processing sequence which is more efficient during the triangulation process. Our new algorithm was simplified and adapted to the VFDT to provide more accurate results based on this improved representation. We compared our algorithm results to Marching Cube triangulation and demonstrated its relevancy based on error metric measurements. In that comparison procedure we also designed a new error metric which is more suitable for our algorithm adaptation to the VFDT.

Future work will include optimizing the processing time of every step of our new algorithm since it is a drawback compared to Marching Cube performances. We will mainly focus on the triangle geometry consistency testing step since it is the most time

consuming one of the overall algorithm. We will also work on adapting our new algorithm to other representations such as continuous implicit surfaces, point clouds and 3D volumetric datasets for medical and related applications. Adapting our algorithm to these representations will provide a useful tool to a wider range of applications in computer graphics. Surface-based triangulation algorithms such as Marching Triangle are more complex to design to obtain efficient results but in general their resulting meshes are of higher quality compared to volume-based methods which are usually simpler to implement.

7. REFERENCES

[1] Rocchini C., Cignoni P., Montani C., Pingi P., Scopigno R.: A Low Cost 3D Scanner Based on Structured Light. EUROGRAPHICS 2001, Manchester, UK, Computer Graphics Forum, Vol. 20, No. 3, 2001, 299-308.

[2] Besl P.J., McKay N.D.: A Method of Registration of 3D Shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 2, 1992, 239-256.

[3] Hilton A., Stoddart A.J., Illingworth J., Windeatt T.: Implicit Surface-Based Geometric Fusion. Computer Vision and Image Understanding, Vol. 69, No. 3, 1998, 273-291.

[4] Davis J., Marschner S., Garr M., Levoy M.: Filling Holes in Complex Surfaces Using Volumetric Diffusion. First International Symposium on 3D Data Processing, Visualization, and Transmission, Padua, Italy, 2002, 428-438.

[5] Fournier M., Dischler J-M., Bechmann D.: 3D Distance Transform Adaptive Filtering for Smoothing and Denoising Triangle Meshes. International Conference on Computer Graphics and Interactive Techniques, Kuala Lumpur, Malaysia, 2006, 407-416.

[6] Nooruddin F.S., Turk G.: Simplification and Repair of Polygonal Models Using Volumetric Techniques. IEEE Transactions on Visualization and Computer Graphics, Vol. 9, No. 2, 2003, 191-205.

[7] Lorensen W.E., Cline H.E.: Marching Cubes: A High Resolution 3D Surface Reconstruction Algorithm. ACM SIGGRAPH 1987, Anaheim, USA, 1987, Computer Graphics, 163-169.

[8] Hilton A., Stoddart A.J., Illingworth J., Windeatt T.: Marching Triangles: Range Image Fusion for Complex Object Modelling. International Conference on Image Processing, Lausanne, Switzerland, 1996, Vol. 1.

[9] Nielson G.M., Hamann B.: The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes. IEEE Conference on Visualization 1991, San Diego, USA, 1991, 83-91.

[10] Chernyaev E.V.: Marching Cubes 33: Construction of Topologically Correct Isosurfaces. Tech. Report, No. CN/95-17, CERN, Geneva, Switzerland, 1995.

[11] Lewiner T., Lopes H., Vieira A.W., Tavares G.: Efficient Implementation of Marching Cubes Cases with Topological Guarantees. Journal of Graphics Tools, Vol. 8, No. 2, 2003, 1-15.

[12] Chien-Chang H., Fu-Che W., Bing-Yu C., Yung-Yu C., Ming O.: Cubical Marching Squares: Adaptive Feature Preserving Surface Extraction from Volume Data. EUROGRAPHICS 2005, Dublin, Ireland, 2005, Computer Graphics Forum, Vol. 24, No. 3, 537-545.

[13] Chan S.L., Purisima E.O.: A New Tetrahedral Tesselation Scheme for Isosurface Generation. Computers and Graphics, Vol. 22, No. 1, 1998, 83-90.

[14] Kobbelt L.P., Botsch M., Schwanecke U., Seidel H.P.: Feature Sensitive Surface Extraction from Volume Data. ACM SIGGRAPH 2001, Los Angeles, USA, 2001, Computer Graphics, 57-66.

[15] Zhang N., Hong W., Kaufman A.: Dual Contouring with Topology-Preserving Simplification Using Enhanced Cell Representation. IEEE Visualization, Austin, USA, 2004, 505-512.

[16] Fournier M., Dischler J-M., Bechmann D.: A New Vector Field Distance Transform and its Application to Mesh Processing from 3D Scanned Data. The Visual Computer Journal, Vol. 23, No. 9-11, 2007, 915-924.

[17] Gopi M., Krishnan S., Silva C.T: Surface Reconstruction Based on Lower Dimensional Localized Delaunay Triangulation. Computer Graphics Forum, Vol. 19, No. 3, 2000, 467-478.

[18] Hartmann E.: A Marching Method for the Triangulation of Surfaces. The Visual Computer, Vol. 14, No. 3, 1998, 95-108.

[19] Bernardini F., Mittleman J., Rushmeier H., Silva C., Taubin G.: The Ball-Pivoting Algorithm for Surface Reconstruction. IEEE Transactions on Visualization and Computer Graphics, Vol. 5, No. 4, 1999, 349-359.

[20] Akkouche S., Galin E.: Adaptive Implicit Surface Polygonization Using Marching Triangles. Computer Graphics Forum, Vol. 20, No. 2, 2001, 67-80.

[21] Karkanis T., Stewart A.J.: Curvature-Dependent Triangulation of Implicit Surfaces. IEEE Computer Graphics and Applications, Vol. 21, No. 2, 2001, 60-69.

[22] Araujo B.R., Jorge J.A.P.: Curvature Dependent Polygonization of Implicit Surfaces. Brazilian Symposium on Computer Graphics and Image Processing, Curitiba, Brazil, 2004, 266-273.

[23] Xi Y., Duan Y.: A Region-Growing Based Iso-Surface Extraction Algorithm. IEEE International Conference on Computer-Aided Design and Computer Graphics, Beijing, China, 2007, 120-125.

[24] McCormick N.H., Fisher R.B.: Edge-Constrained Marching Triangles. International Symposium on 3D Data Processing Visualization and Transmission, Padova, Italy, 2002, 348-351.

[25] Moller T.: A Fast Triangle-Triangle Intersection Test. Journal of Graphics Tools, Vol. 2, No. 2, 1997, 25-30.

[26] Hilton A., Illingworth J.: Marching Triangles: Delaunay Implicit Surface Triangulation. Tech. Rep. EPSRC-GR/K04569, Centre for Vision Speech and Signal Processing, University of Surrey, Guildford, UK, 1997, 1-12.

Adaptation of Mouse Brain Gene Expression Data for further Statistical Parametrical Mapping Analysis

Anton Osokin[(1)], Alexey Lebedev[(2)], Dmitry Vetrov[(1)], Vladimir Galatenko[(2)], Dmitry Kropotov[(3)], Alexandr Nedzved[(4)], Anton Konushin[(1)], Konstantin Anokhin[(5)] *

(1) Department of Computational Mathematics and Cybernetics, Moscow State University, Moscow, Russia
osokin.anton@gmail.com, vetrovd@yandex.ru, ktosh@graphics.cs.msu.ru http://cs.msu.ru

(2) Department of Mechanics and Mathematics, Moscow State University, Moscow, Russia
alleb@list.ru, vgalat@castle.nmd.msu.ru, http://www.math.msu.ru

(3) Dorodnicyn Computing Center of the Russian Academy of Sciences
dkropotov@yandex.ru, http://www.ccas.ru

(4) United Institute of Informatics Problems, National Academy of Sciences of Belarus
abelotser@newman.bas-net.by, http://http://uiip.bas-net.by

(5) P.K. Anokhin Institute of Normal Physiology. E-mail:
k.anokhin@gmail.com, http://www.aha.ru/ niinf

Abstract

The paper describes a method for fully automatic 3D-reconstruction of mouse brain voxel model from a sequence of coronal 2D slices for statistical analysis of gene expression. Two images of each brain slice with different stains are used. The first stain highlights the histology of brain, which is used for slice matching. The second stain highlights the level of gene expression. The algorithm proceeds as follows. First, images are preprocessed to suppress image noise and equalize image brightness. Second we estimate the level of gene expression in each slice using the second stain. Then we construct 3D-model of the brain using the first stain. To do this all images are aligned via rigid-body transformations. After alignment neighboring slices are matched by estimation of non-linear deformations. As the distance between slices is significantly larger then image resolution we add intermediate virtual slices using morphing algorithm. Gene expression level is interpolated in identical way. The obtained 3D-model with the information about gene expression can be used for gene expression analysis via Statistical Parametric Mapping (SPM) package. The proposed method for 3D-reconstruction has been tested on images from Allen Brain Atlas, which is available in electronic form.

Keywords: *3D-Reconstruction, neuroimaging, image transformations, morphing, elastic deformations, image registration, B-splines.*

1. INTRODUCTION

The problem of gene expression analysis using only images of brain slices is very important in modern brain research [1]. Neurobiologists are now able to measure the activity of selected gene in every brain cell. This is usually done in vitro, i.e. on dead species. The extracted brain is frozen and then cut into slices. Each slice is double-stained by Nissl method to highlight histology and by special stain which reveals the neurons with expression of corresponding genes.

The further automatic analysis of gene expression is significantly more complicated problem, compared to analysis of brain activity, measured by other neuroimaging techniques, like fMRI or PET. For fMRI data a well-known Statistical Parametric Mapping (SPM) framework is widely used[4]. SPM uses 3D voxel model as input data and provides tools for model reconstruction from series of slice images. However, SPM package cannot be directly applied for gene analysis problem. First, due to technological aspects of brain cutting procedure brain slices differ in their actual size, shape and orientation, and SPM fails to correctly reconstruct model from such kind of data. Second, statistical analysis in SPM package relies on spatial coherency of brain activity, which is the reason why it uses uniform voxel models. fMRI data is acquired in fairly low resolution (128*128 or 256*256 images are a common case) and it is easy to acquire similar number of fMRI images to ensure that voxel model has same resolution in all dimensions. But for gene expression analysis image resolution should be significantly higher, and it is impossible to obtain corresponding number of slices for many technological reasons. Thus it is necessary to reconstruct intermediate layers from existing slices.

In this paper we propose a novel method for voxel model reconstruction from the set of brain slice images that allows further analysis by SPM package. Our method is fully automatic and consists of following steps. First, images are preprocessed to suppress image noise and equalize image brightness. Then all images are aligned via rigid-body transformations. Consecutive slices are matched by estimation of non-linear deformations between layers. Intermediate layers are then interpolated using estimated non-linear transformations. Finally, data is converted to the voxel models and saved in the NIFTI-1 file format [20] that is standard for the brain studies.

We have tested our method 3D-reconstruction on images from Allen Brain Atlas [3] and showed that it leads to comparable results with the ones in AGEA project where 5 times more slices are used for 3D-reconstruction.

The rest of paper is organized as follows. In section 2 we describe the preprocessing of brain slice images with histological and gene expression stains. Section 3 describes how gene expression level is computed. In section 4 image alignment and intra-slice interpolation method is described. Section 5 contains some experimental results. Some conclusions are given in the last section.

2. BRAIN SEGMENTATION AND ILLUMINATION CORRECTION

On experimental images of mouse brain slices noise level is very high due to uneven amount of stain in different part of the slice. This results in abrupt deformations of brain slice contour that de-

*This work was supported by Russian Foundation for Basic Research, projects Nos. 08-01-00405, 08-01-90016 and 09-04-12215.

grade alignment and matching. To suppress such defects we apply segmentation algorithm based on graph cuts with prior assumptions on slice shape. [6, 7, 8].

Uneven illumination of brain images is also a problem. For different brain images contrast is different and even within one separate image there are areas with different illumination levels. For illumination correction we apply Single Scale Retinex method [9]. First, Gauss filter with large radius is applied to the image:

$$g(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, -R \le x,y \le R.$$

We used $\sigma = 20, R = 100$. This operation gives an illumination map of the image I_{map}.

Afterwards the initial image is divided by obtained filtered image

$$I_{new}(i,j) = \frac{I(i,j)}{I_{map}(i,j)+1}.$$

This operation provides both equal local illumination within each image and equal illumination of different atlas images.

Figure 1 and figure 2 illustrate this procedure.

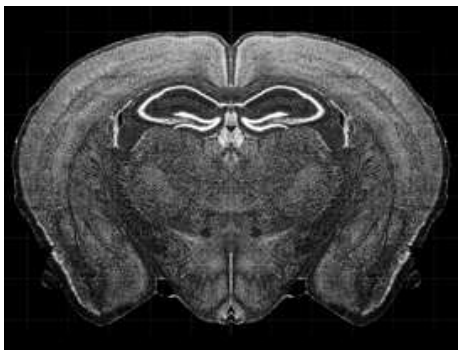


Figure 1: Histological brain image after Graph Cut algorithm without illumination correction.

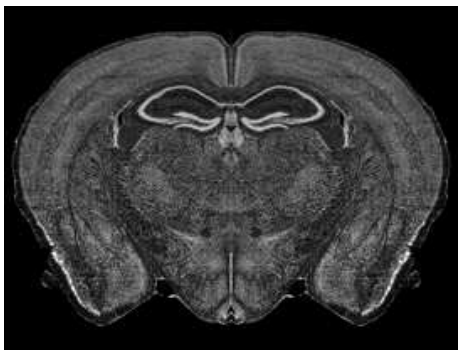


Figure 2: Histological brain image after illumination correction.

3. EXPRESSION DETECTION AND EVALUATION

To measure the gene expression in stained brain slice its image is acquired with very high resolution so that each stained nucleus is visible. But such high resolution make further processing and analysis computationally impractical and it should be reduced. Straightforward resolution decrease will smooth image and introduce errors in gene expression measurement. So we need a specific procedure for rescaling of such images.

For this purpose an expression level is evaluated for each image zone that corresponds to one pixel after the resolution reduction. First we evaluate the expression level for each pixel of the initial image, then apply a standard Gauss filter with radius close to the new pixel size (this also leads to smoothing required for the subsequent statistical analysis), and finally evaluate the expression level for each new pixel by simple summation and rescaling. Note that in following versions we plan to replace the application of Gauss filter by non-uniform smoothing method that takes account of anatomic structures.

The results of expression level detection and evaluation can be corrected manually if required by changing the default thresholds and other parameters (globally or locally, in a certain image fragment) and/or by setting the expression level in certain image points manually. However the designed procedure of expression detection and evaluation is fully automated and normally the manual correction is not required.

The most critical part of this step is the evaluation the expression level for each pixel of the initial image. In current implementation the fact of expression presence in a point is not simply binary, but is represented by a real number from 0 (no expression) up to 1 (maximum expression level). Intermediate values represent the relative expression level. Such approach is natural from the biological viewpoint, it also leads to the correctness of the computations (where correctness is understood in a standard mathematical sense) and hence eliminates the errors connected with close-to-threshold values.

The procedure of expression evaluation can be shortly described as follows. The image is first transformed from RGB to one-parametric color palette (in fact it is an analogue of the grey-scale palette, but different coefficients can be used for the transformation). Similarly to [10] two different one-parametric color schemes are used in order to reduce errors. For each color scheme the value of a sigmoid function (i.e., a smooth function that equals 0 for arguments less than the lower threshold and equals 1 for arguments greater than the upper threshold) is computed at each pixel: the color of the pixel in the one-parametric color palette is used as an argument, and the parameters of the sigmoid function depend on the distance from the brain border and the mean color value in the large neighborhood. The expression activity at each pixel is set to the product of the values of these sigmoid functions. Figures 3, 4 show the simplified result of expression detection.

4. 3D MODEL CONSTRUCTION

4.1 Image alignment

In fMRI imaging the alignment transformations between images are assumed to be either affine or rigid-body in 3D or 2D space [11]. In our case slices are already horizontally centered and aligned w.r.t symmetry line during imaging process. But cutting procedure introduces small deformations in vertical direction, which are different for each slice. So we limit the set of alignment transforms by vertical shifts and stretches.

Image alignment proceeds as follows. First, we search for the smallest surrounding rectangle of each slice. Then we consider rectangle border to be a function of slice number. Figure 5 shows example of top and bottom borders of brain rectangles without alignment. These functions are not smooth enough for 3D reconstruction. To smooth these functions we apply Savitzky-Golay filter.

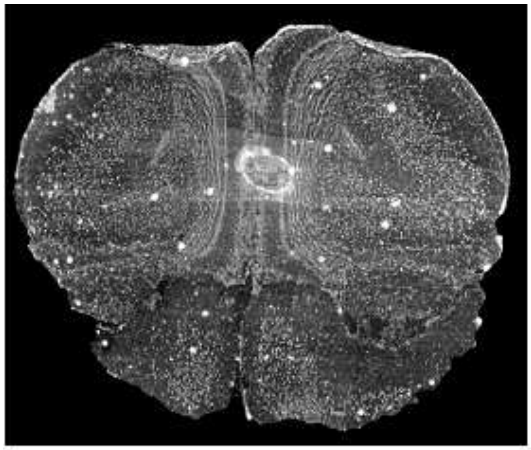


Figure 3: Initial image of a brain region.

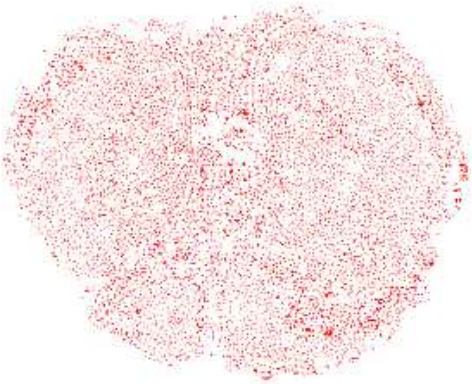


Figure 4: Image of the brain region where pixels with expression level greater than threshold (0.5) are marked.

The idea of Savitzky-Golay filtering is to find filter coefficients that preserve higher moments. Equivalently, the idea is to approximate the underlying function within the moving window not by a constant (whose estimate is the average), but by a polynomial of higher order (we used order 5 in our approach). For each point we least-squares fit a polynomial to the points in the moving window (we used window width 15), and then set the new value to be the value of that polynomial at the same position.

Savitzky-Golay filters can be thought of as a generalized moving average. Their coefficients are chosen this way to preserve higher moments in the data, thus reducing distortion of essential features of data like peak heights and line widths in a spectrum, while the suppression of random noise is improved. Figure 6 shows example of top and bottom borders of brain rectangles after alignment.

If we are interested in any specific section of mouse brain we can make additional alignment in appropriate plane. Such alignment makes specific section smoother but the whole model becomes less smooth. So in general we don't use specific plane alignment for 3D-model reconstruction.

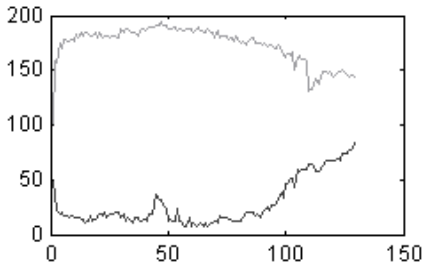


Figure 5: Not aligned top (U) and bottom (D) borders of mouse brain.

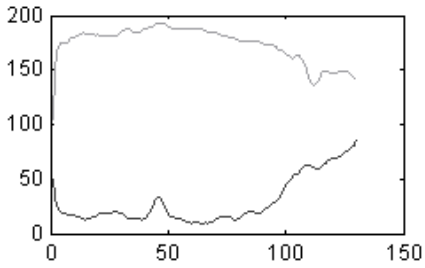


Figure 6: Aligned top (U) and bottom (D) borders of mouse brain.

4.2 Non-linear transformations

To obtain the voxel model with uniform resolution we need to fill the gaps between available slices by synthesizing of intermediate layers. For this task we estimate non-linear deformation between consecutive slices.

There exist many methods for non-linear deformation estimation: parametric model of the deformation [11, 12, 13, 14], hierarchical models [15, 16], nonparametric local methods [17], inclosed dynamic programming [18], optical flows [19]. In this paper we follow the papers [12, 13] and use the approach based on parametric model of deformation based on B-spline basis functions. Choice of B-splines as basis functions provides good quality of deformation and high speed of calculation because of limited number of basis functions.

A brain 3D model is a function:

$$F : \mathbb{R}^3 \to [0, 1].$$

From slices we know F values only at some discrete points. In slice plane expansion of discrete function to its continuous version is a weighted sum of surrounding discrete point colors. Expansion in other plane can be done in the same way (weighted sum of neighboring slices). However, this simple solution makes a 3D model not smooth enough. A better solution can be obtained using non-linear image deformations.

The input images are given as two 2-dimensional discrete functions:

$$f_1, f_2 : I \subset \mathbb{Z}^2 \to [0, 1].$$

Here I is a 2-dimensional discrete interval covering the set of all pixels in the image. Function values stand for intensities of corresponding pixels.

We denote continuous expansions of two images as f_1^c, f_2^c.

Our goal is to find a deformation of the first image to the second one in the following way:

$$f_1^c(g(x,y)) \approx f_2(x,y).$$

Here $g(x,y) : \mathbb{R}^2 \to \mathbb{R}^2$ is a deformation (correspondence) function between pixels.

We measure the difference between images by SSD (sum of squared deviations) criterion:

$$E = \sum_{(i,j)\in I} (f_1^c(g(i,j)) - f_2(i,j))^2.$$

So the problem is to minimize E with respect to deformation function g.

We consider deformation function as a linear combination of some basis functions:

$$g(x,y) = \sum_{k\in K} \vec{c_k} b_k(x,y).$$

Here K is a set of basis function indexes.

Family of deformation functions 4.2 transforms optimization problem in functional space into finite-dimensional optimization problem.

We use uniformly spaced cubic B-splines as basis functions.

A B-spline β_r of degree r is recursively defined as

$$\beta_r = \beta_{r-1} * \beta_0, r > 0.$$

β_0 is a characteristic function of $[-0.5, 0.5]$, $*$ is convolution operator.

Specifically, cubic B-spline is the following function:

$$\beta_3(x) = \begin{cases} 2/3 - (1-|x|/2)x^2, 0 < |x| \le 1, \\ (2-|x|)^3/6, 1 < |x| < 2, \\ 0, |x| \ge 2. \end{cases}$$

So we are looking for the deformation function in the family:

$$g(x,y) = \sum_{(k_x,k_y)\in K} \beta_3(x/h_x - k_x)\beta_3(y/h_y - k_y).$$

Centers of B-spline functions are placed on the regular grid $(k_x h_x, k_y h_y)$. Working with uniform splines is significantly faster with respect to nonuniform splines. In order to get complete control over g, we put some spline knots outside the image.

Finally the problem is to optimize SSD criteria E w.r.t. set of parameters c. Here we use gradient descent algorithm with feedback step size adjustment. In this algorithm parameter update rule is $\Delta c = -\mu \nabla_c E(c)$. After a successful step μ is multiplied by some value $\mu_f > 1$, otherwise it is divided by some other value $\mu_f^* > 1$.

An example of deformation field obtained from B-spline basis functions for a pair of neighboring slices from Allen Atlas is shown in figure 7.

Since we have deformation of the first image to the second one and vice versa, we can fill gaps between atlas slices with weighted sum of deformed neighboring slices:

$$F(x,y,z) = \alpha f_{1,k-1}^{\alpha}(x,y) + (1-\alpha) f_{2,k}^{1-\alpha}(x,y).$$

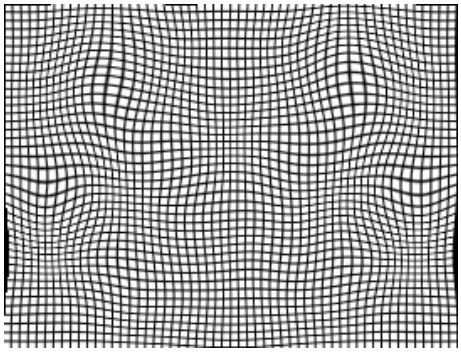


Figure 7: Deformation field for B-spline method. This deformation field is obtained by applying deformation of neighboring slices to regular grid.

Here $\alpha = \frac{z-z_{k-1}}{z_k - z_{k-1}}$, $z_{k-1} \le z < z_k$, z_k is a z-coordinate of slice number k.

$$f_{1,k-1}^{\alpha}(x,y) = f_{k-1}((x,y) + \alpha(g_{k-1}^{k}(x,y) - (x,y))).$$

$$f_{2,k}^{\alpha}(x,y) = f_{k}((x,y) + \alpha(g_{k}^{k-1}(x,y) - (x,y))).$$

Here $g_i^j(x,y)$ is a deformation function of slice number i to slice number j.

As a last step, the full set of input and intermediate slices is converted to voxel model and saved in the NIFTI-1 file format [20], which is a standard for the brain studies.

5. EXPERIMENTAL RESULTS

In order to test the proposed method for 3D-model construction we used it on coronal Allen Brain Atlas [21, 3].

The coronal Allen Brain Atlas is a set of full-color, high-resolution coronal digital images (132 images) of mouse brain accompanied by a systematic, hierarchically organized taxonomy of mouse brain structures. The Allen Brain Atlas is obtained from 8-week old C57Bl/6J male mouse brain prepared as unfixed, fresh-frozen tissue.

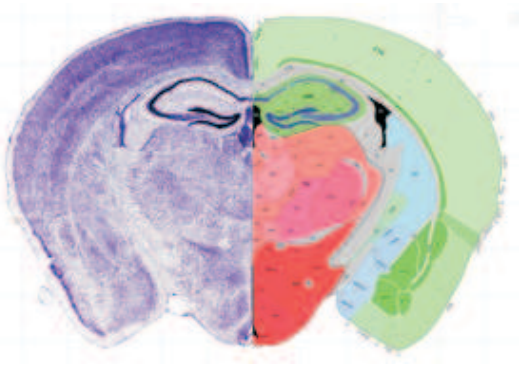


Figure 8: Allen Brain Atlas image.

On figure 8 the left half represents histological structure of one mouse brain slice. The right half shows color annotation of mouse brain structures made by human experts. We have used histological images to reconstruct a 3D voxel model. Afterwards we applied same morphing transformations to create virtual slices of annotated images to demonstrated that our method correctly preserves the inner brain structures.

Figures 9 and 10 show synthesized histological and color annotated axial slices of model built without illumination correction and slice alignment. Figures 11 and 12 show synthesized histological and color annotated axial slices of model built with both illumination correction and slice alignment, but without non-linear deformations between neighboring slices.

Figures 13 and 14 show synthesized histological and structural axial slices of model built with illumination correction, slice alignment and with non-linear deformations between neighboring slices.

These figures show significant improvement of 3D model quality with each step. 3D model from figures 13 and 14 is much smoother than the previous ones. It should be mentioned that not only brain borders become smoother but also borders of internal structures.

We also compared our method of virtual slices generation with the analogous method used in AGEA project [22]. Figure 15 shows appropriate axial brain view from AGEA project. The quality of view, obtained by our method, is very similar to one of AGEA project, but compared compared to full set of slices without gaps, used in AGEA, we have used only each 5th slice.

The results of the automated gene expression detection and evaluation were checked by human experts and were found to be satisfactory.

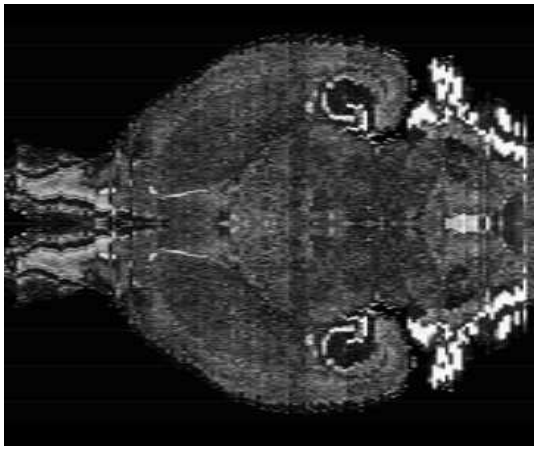


Figure 9: Axial histological view of 3D model without illumination correction, alignment and nonlinear deformations.

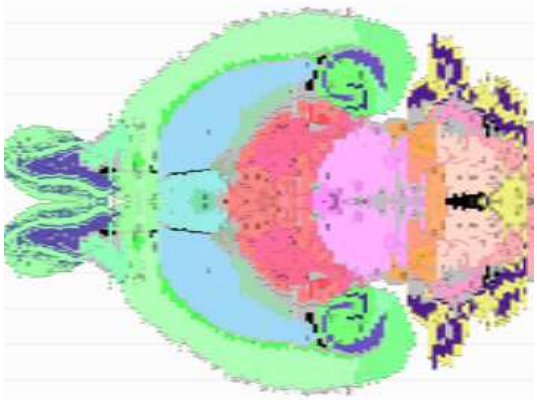


Figure 10: Axial structural view of 3D model without illumination correction, alignment and nonlinear deformations.

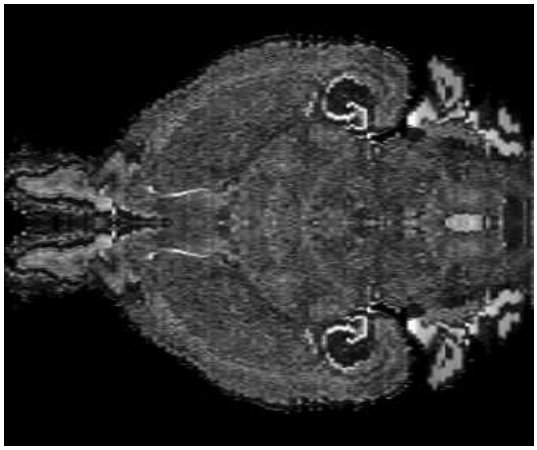


Figure 11: Axial histological view of 3D model with illumination correction and alignment.

6. CONCLUSION

We proposed an algorithm that constructs virtual slices of brain w.r.t. arbitrary section-plane. We have shown that such algorithm allows us to get synthetic images of relatively good quality both with histological and anatomical structure. Such algorithm opens great perspectives for further brain research as it provides the opportunity of discovering the anatomical structures in a single slice of real mouse brain. The procedure of slices' preparation is very time and labor consuming, that is why it is highly desirable to reduce the number of slices obtained from real mouse to minimum (in the limit to one which is of interest for biologists). The slice can be made in non-standard (coronal, sagittal, or axial) section-plane and it should be mapped into 3D-model of atlas brain. Our algorithm allows us to synthesize the image of an atlas brain w.r.t. any section-plane and hence is the key part of future method which will compute the best mapping. When it is done the anatomical structures in real brain slice can be found easily by projecting anatomical structure of atlas brain onto the virtual slice with further performing inverse mapping to adapt it to real brain slice. The algorithm for identifying anatomical structures in arbitrary brain slice is extremely useful for brain research as it allows to understand what structures are responsible for specific genes expression in specific situations.

Note that the quality of statistical analysis (including SPM analysis) can be significantly increased by using the knowledge about anatomic structures. A simple way of structural annotation of a model brain is based on the geometric alignment of this model with the model that integrates histologic data and anatomic (structural) data. Such model was constructed based on Allen Brain Atlas. The alignment can be built, e.g, using standard SPM procedures for geometric processing.

7. REFERENCES

[1] L. Ng, S.D. Pathak, C. Cuan, C. Lau, H. Dong, A. Sodt, C. Dang, B. Avants, P. Yushkevich, J.C. Gee, D. Haynor, E. Lein, A. Jones, and M. Hawlyrycz, "Neuroinformatics for genome-wide 3d gene expression mapping in the mouse brain," *IEEE Transactions on Computational Biology and Bioinfomatics*, vol. 4, no. 3, pp. 382–393, 2007.

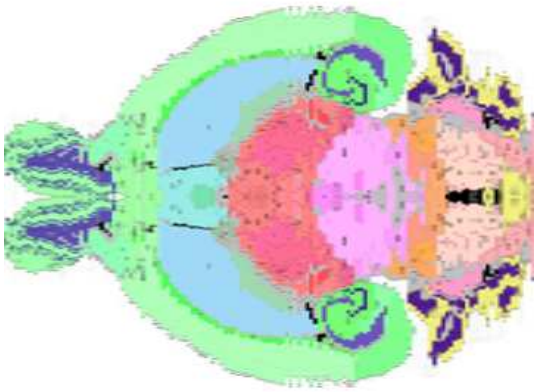


Figure 12: Axial structural view of 3D model with illumination correction and alignment.

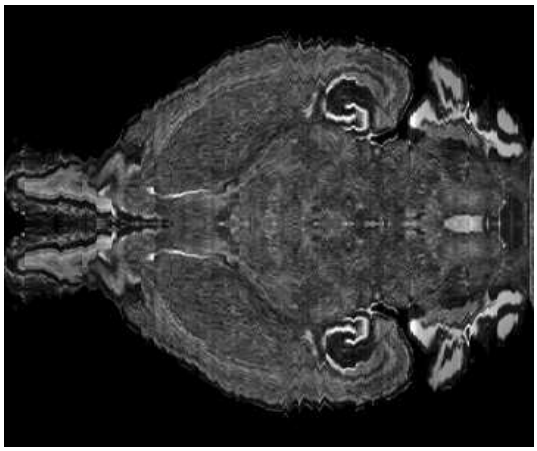


Figure 13: Axial histological view of 3D model with illumination correction, alignment and nonlinear deformations.

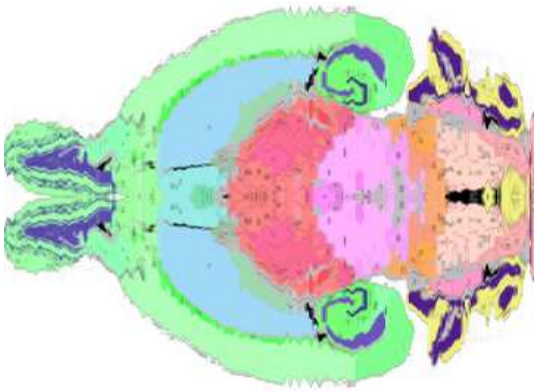


Figure 14: Axial structural view of 3D model with illumination correction, alignment and nonlinear deformations.

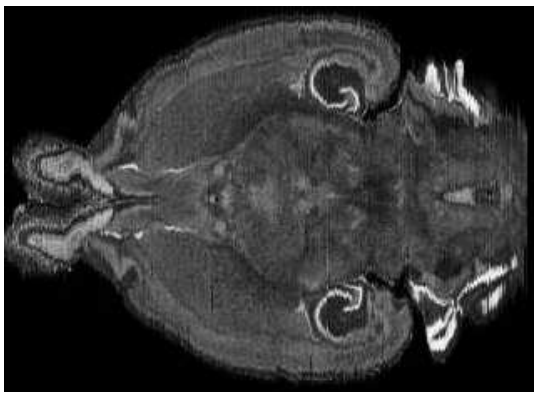


Figure 15: Axial histological view of 3D model of AGEA project.

[2] J. Bolyne, E.F. Lee, and A.W. Toga, "Digital atlases as a framework for data sharing," *Frontiers in Neuroscience*, vol. 2, pp. 100–106, 2008.

[3] Seattle (WA): Allen Institute for Brain Science, "Allen mouse brain atlas," Internet, Available from: http://mouse.brain-map.org, 2009.

[4] "Spm8," Internet, Available from: http://www.fil.ion.ucl.ac.uk/spm/, 2009.

[5] "Mricron," Internet, Available from: http://www.sph.sc.edu/comd/rorden/mricron/, 2009.

[6] Y. Boykov, O. Veksler, and R. Zabih, "Efficient approximate energy minimization via graph cuts," *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1222–1239, 2001.

[7] V. Kolmogorov and R. Zabih, "Energy functions can be minimized via graph cuts?," *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 147–159, 2004.

[8] Y. Boykov and V.Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.

[9] Y. Boykov and V.Kolmogorov, "Multi-scale retinex for color image enhancement," *Proceedings of ICIP*, vol. 3, pp. 1003–1006, 1996.

[10] J. P. Carson, T. Ju, C. Thaller, J. Warren, M. Bello, I. Kakadiaris, W. Chiu, and G. Eichele, "Automated characterization of gene expression patterns with an atlas of the mouse brain," *Proceedings of the 26th Annual International Conference of the IEEE EMBS*, vol. 2, pp. 2917–2920, 2004.

[11] R.S.J. Frackowiak, K.J. Friston, C. Frith, R. Dolan, C.J. Price, S. Zeki, J. Ashburner, and W.D. Penny, *Human Brain Function*, Academic Press, 2nd edition, 2003.

[12] J. Kybic, P. Thevenaz, A. Nirkko, and M. Unser, "Unwarping of unidirectionally distorted epi images," *IEEE Transactions on Medical Imaging*, vol. 19, no. 2, pp. 80–93, 2000.

[13] J. Kybic and M. Unser, "Fast parametric elastic image registration," *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1427–1442, 2003.

[14] Y. Wu, T. Kanade, C. Li, and J. Cohn, "Image registration using waveletbased motion model," *International Journal of Computer Vision*, vol. 38, pp. 129–152, 2000.

[15] P. Moulin, R. Krishnamurthy, and J. Woods, "Multiscale modeling and estimation of motion fields for video coding," *IEEE Trans. Image Processing*, vol. 6, pp. 1606–1620, 1997.

[16] O. Musse, F. Heitz, and J.-P. Armspach, "Topology preserving deformable image matching using constrained hierarchical parametric models," *IEEE Trans. Med. Imag.*, vol. 10, pp. 1081–1093, 2001.

[17] R. Bajcsy and S. Kovacic, "Multiresolution elastic matching.," *Computer Vision, Graphics, and Image Processing*, vol. 46, pp. 1–21, 1989.

[18] T. Ju, J. Warren, J. Carson, M. Bello, I. Kakadiaris, W. Chiu, C. Thaller, and G. Eichele, "3d volume reconstruction of a mouse brain from histological sections using warp filtering," *Journal of Neuroscience Methods*, vol. 156, pp. 84–100, 2006.

[19] J.L. Barron and D.J.Fleet ans S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[20] "Nifti-1 file format," Internet, Available from: http://nifti.nimh.nih.gov/, 2009.

[21] E.S. Lein et al.., "Genome-wide atlas of gene expression in the adult mouse brain," *Nature 445*, pp. 168–176, 2007.

[22] Seattle (WA): Allen Institute for Brain Science, "Agea project," Internet, Available from: http://mouse.brain-map.org/agea/, 2009.

ABOUT THE AUTHORS

Anton Osokin — 4th-year student of Computational Mathematics and Cybernetics department of Moscow State University. E-mail: Osokin.Anton@gmail.com

Lebedev Alexey — 3rd-year student of Mechanics and Mathematics department of Moscow State University. E-mail: alleb@list.ru

Dmitry Vetrov — PhD, researcher in the chair of Mathematical Methods of Forecasting, Computational Mathematics and Cybernetics department of Moscow State University. E-mail: VetrovD@yandex.ru

Galatenko Vladimir — PhD, associate professor of Mechanics and Mathematics department of Moscow State University. E-mail: vgalat@castle.nmd.msu.ru

Dmitry Kropotov — junior researcher in Dorodnicyn Computing Centre of the Russian Academy of Sciences. E-mail: DKropotov@yandex.ru

Alexandr Nedzved — PhD, researcher in United Institute of Informatics Problems, National Academy of Sciences of Belarus. E-mail: abelotser@newman.bas-net.by

Anton Konushin — PhD, researcher in Graphics and Mutlimedia Lab, Computational Mathematics and Cybernetics department of Moscow State University. E-mail: ktosh@graphics.cs.msu.ru

Konstantin Anokhin — doctor of medicine, member-correspondent of the Russian Academy of Medical Sciences, head of systemogenesis department in P.K. Anokhin Institute of Normal Physiology. E-mail: k.anokhin@gmail.com

Fast Image Matching with Visual Attention and SURF Descriptors

Vitaly Pimenov
Faculty of Applied Mathematics and Control Processes
Saint–Petersburg State University, Saint–Petersburg, Russia
vitaly.pimenov@gmail.com

Abstract

The paper describes an image matching method based on visual attention and SURF keypoints. Biologically inspired visual attention system is used to guide local interest point detection. Interest points are represented with SURF descriptors. One–to–one symmetric search is performed on descriptors to select a set of matched interest point pairs. Pairs are then weighted according to attention distribution and weights are summed up yielding a similarity score. Images are considered to be near–duplicates if similarity score exceeds a certain threshold. Experimental results illustrate high accuracy and superior computational efficiency of proposed approach in comparison with other matching techniques.

Keywords: *image matching, visual attention, local interest points, near–duplicate detection.*

1. INTRODUCTION

Image matching problem had in recent years gained remarkable attention in computer vision [1, 2], robotics [3, 4, 5] and information retrieval [6, 7, 8, 9]. A broad spectrum of practical applications such as object recognition [2, 10], content–based image retrieval [9], news retrieval [8], medical imaging [11], copyright infringement detection [7], robotic navigation [4], scene reconstruction [12] and other depends on efficient techniques for finding correspondences between images.

Image matching problem is also called *near–duplicate image identification* [6, 13, 7] or *transformed image identification* [14] in literature and consists of three stages: feature detection, feature representation and feature matching [10]. Matching problem is quite close to image registration problem [15], but, as opposed to registration, transform model estimation and image resampling are not needed for matching. The goal of matching is to detect near–duplicate images. Near duplicates are images that are equal despite the slight degree of variations caused by geometric and photometric transforms, such as lighting and viewpoint changes, differences in frame acquisition time, motion, editing etc. The complexity of matching arises from a big multitude of possible transforms. Several examples of near duplicate images are shown on Fig. 1.

In response to an increasing practical demand a variety of matching methods was developed. The performance of these methods was evaluated in diverse scenarios resulting in conclusion that local interest point matching methods are most promising [16, 10, 17, 18, 6]. Local interest points (keypoints) are salient regions detected over image scales [2]. Image matching in this case becomes a keypoint descriptor matching, where descriptor is a vector of features extracted from image patch around interest point at given scale. Different keypoint detectors and descriptors were proposed for image matching [2, 19, 10, 20, 21]. Later several studies were conducted to assess their quality (see e.g. [16, 20, 10, 18]).

Although local interest point matching remains to be the most accurate matching method its major drawback is high computational complexity: comparing two 640×480 images can take up a million of descriptor comparisons. Real–time implementation (e.g. for mobile robot vision) becomes prohibitively difficult. A variety of hardware focused approaches was already proposed for most critical tasks: VLSI–based solutions [22], FPGA architectures [23, 24], parallel [25] and DSP [26] systems.

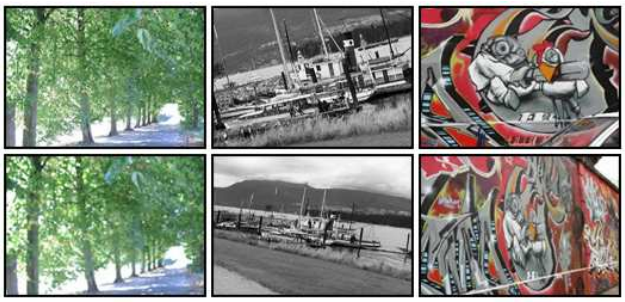


Figure 1: Examples of near duplicate image pairs.

Software alternative lies in keypoint filtering. It is known that geometric verification techniques such as RANSAC [27] require a small number of keypoint for reliable matching [28, 7, 2]. Different techniques were proposed to reduce the number of descriptor comparisons. Most of them rely on two–stage matching, where first stage is rough comparison (e.g. via descriptor discretization) to filter out non–neighbors, and second one is fine matching involving remaining descriptors.

Such approaches successfully reduce computational load, however they are still far away from real–time requirements. Best to our knowledge *matching time* estimations reported in [6] and [13] are 0.028 sec and 0.015 sec respectively. These times do not include *detection time* that is necessary to detect keypoints and compute descriptors. According to results reported in [10] detection time for widely used methods exceeds 0.45 sec per image. At the same time, mobile robot operating at moderate 3 Hz frequency has 0.33 sec for whole control system loop execution. Considering content–based retrieval where detection time is not as important as matching time, we see that 0.015 sec per pair matching results in poor 66 comparisons per second.

To overcome this difficulty on software level much simpler detection and matching methods are employed, e.g. Shi–Tomasi operator [21] or color histograms [29] that allow 10^{-5} sec matching time. However, in general, these methods cannot achieve accuracy level of their keypoint–based competitors.

Current paper proposes novel image matching method guided by a biologically motivated visual attention system. The key advantage of such system is suppression of detected keypoints number. It is achieved during detection phase performed with SURF detector [10] by filtering out points that gain little attention. As a result matching speed increases by more that an order of magnitude, detection speed increases by 2 – 4 times. Experimental results show high accuracy of matching, comparable to accuracy of descriptor matching implemented without filtration.

The remaining sections are organized as follows. Section 2 surveys related work in visual attention guided image matching and descriptor filtering. Section 3 describes proposed visual attention system. Interest point matching method is discussed in Sect. 4. Experiments and evaluation results are presented in Section 5. Finally, Sect. 6 concludes the paper and outlines directions of further research.

2. RELATED WORK

Visual attention has emerged recently as a powerful tool to make computational vision more effective, since it allows focusing analysis on restrained image areas [30, 4, 31]. This section presents a survey of modern attention–based image matching methods and descriptor filtering algorithms.

At the heart of modern computational attention theories is the concept of *saliency map*. As originally proposed in [32], saliency map refers to a *"topographically arranged map that represents visual saliency of a corresponding visual scene"* [33].

Typically, attention–based matching methods employ saliency map to detect interest points — point with high saliency. To describe interest point various approaches are used in literature.

Saliency based image matching method is presented in [34]. The foundation of this method is *Scale – Saliency* algorithm developed by Kadir and Brady [35] aimed to detect image regions salient by means of Shannon entropy measure [36]. There are two drawbacks of such approach. First is the use of normalized intensity histograms to describe interest points, since its invariance to geometric and photometric transforms is debatable. Second, *Scale – Saliency* algorithm suffers from the curse of dimensionality when it is applied to multidimensional data, e.g. RGB images. Although this issue was resolved recently [37], there is no experimental evidence proving the efficiency of multidimensional *Scale – Saliency* in image matching tasks.

The question of salient regions detection repeatability is considered in [38]. Comparison with popular Difference–of–Gaussians [2] and Harris–Laplace [19] detectors reveals the superior repeatability of detection performed by biologically inspired visual attention system. Therefore, it is concluded that filtering out descriptors corresponding to regions that cannot be detected with high repeatability can significantly reduce computational load. This result supports the motivation of current research.

Works by Stentiford and colleagues [39, 40, 17, 41] focus on constructing attention based similarity measures with applications to content–based image retrieval, motion detection and tracking. Reported results show superior performance of their approach in terms of recall–precision graph when compared to color histograms and Gabor signatures. However, these techniques had not yet been tested against local interest point detectors and no exact data concerning matching speed is available.

More sophisticated bio–inspired attention system is proposed in [9] for content–based image retrieval. In order to improve quality of detection this system combines Stentiford model of attention [41] with Itti–Koch model [30]. RGB and HMMD color intensity histograms were used as descriptors. Reported experimental result suggest high accuracy of this approach, however it was not compared to interest point based detectors.

Broad comparative study described in [18] aimed to evaluate modern near–duplicate detection methods on a large scale collection (more than one million web images) had revealed high performance of bio–inspired retina–like image matching method. The study reported poor performance of SURF [10], interest point detection method, however descriptor matching algorithm used in experiments in highly debatable, as it is mentioned by authors themselves. Furthermore, image collection was built by applying image transforms to an initial image collection, thus validity of evaluation result for natural near–duplicates detection is arguable in this case.

At the same time, question of interest point matching scalability has already been tackled from the perspective of pruning the number of descriptors used to match [28]. SIFT [2] and PCA–SIFT [7] descriptors were chosen in this research. Pruning was made on the basis of point contrast. Experimental results show reduction of execution time to $1/50$ of the original approach without any significant loss of accuracy. Nevertheless, this study focuses on image retrieval task, therefore only retrieval query speed is evaluated, while matching speed is unknown.

Summing up, attention–based approaches to image matching became quite popular along with the development of human vision models. In the light of results presented in [28], employing visual attention to filter out insignificant interest points seems to be a promising approach to improve efficiency of image matching.

3. VISUAL ATTENTION SYSTEM

The purpose of visual attention for image matching tasks if defined as following: *determine the importance of image regions in terms of attention distribution*.

Computation is performed at two stages: first, saliency map is computed from an input image; second, an absolute measure of saliency is built from the saliency map.

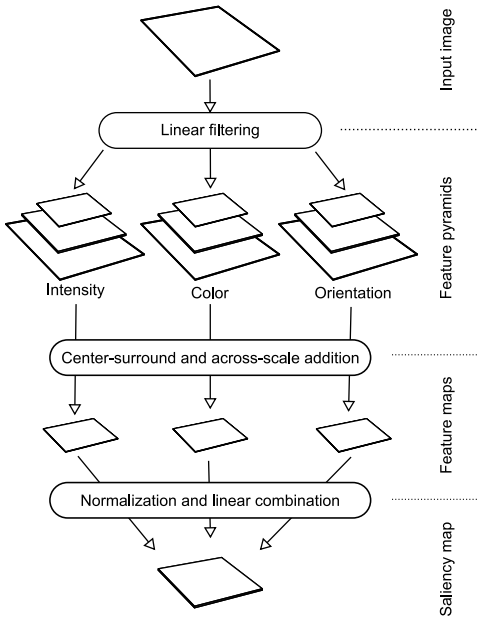


Figure 2: Overview–diagram of saliency map calculation.

In current research saliency map is implemented on pixel basis as it was proposed in [30, 4]. For each pixel saliency is computed from three feature channels mimicking human vision [42]: intensity, color and orientation. After than, per–channel maps are normalized and fused into a saliency map. Figure 2 illustrates processing flow, its details are outlined in Sect. 3.1.

Absolute measure of saliency is necessary because saliency maps contain relative values. Two distinct methods are proposed and described in Sect. 3.3 to solve this problem.

3.1 Feature Computation

Feature computation is roughly based on procedures proposed for visual attention system *VOCUS* [4]. In following subsections a brief description of procedures for each feature channel is offered.

Input image I is used to build monochrome and LAB Gaussian pyramids of five scales: $(s_0, \ldots, s_4)$ and $(\bar{s}_0, \ldots, \bar{s}_4)$ respectively; and Difference–of–Gaussians pyramid [2] of four scales $(DoG_1, \ldots, DoG_4)$. First two scales (s_0, s_1) are not considered in calculations to ensure robustness to noise [30].

3.1.1 Intensity

Intensity feature maps are computed by applying *center–surround mechanism* to images of Gaussian pyramid. On–center and off–center intensity [4, 42] of a pixel is computed by comparing its value with mean value of surrounding pixels. Such calculation is consistent with recent findings about surround suppression mechanisms in visual cortex [31]. Mean value is calculated by summing pixel values in 7×7 quadratic area Ω around pixel of interest:

$$I^{[s_i]}_{on-center}(x,y) = s_i(x,y) - \frac{\sum\limits_{(\bar{x},\bar{y})\in\Omega} s_i(\bar{x},\bar{y}) - s_i(x,y)}{|\Omega| - 1} \quad . \quad (1)$$

For efficiency reasons integral images [43] are used to compute a sum in equation (1).

On–center and off–center intensity maps are built for scales (s_2, s_3, s_4). Resulting maps are summed up by *across–scale addition* ($\bigoplus$) [4] — smaller maps are scaled to size of biggest one and then maps are summed up by pixels — leading final intensity maps $I_{on-center}$ and $I_{off-center}$:

$$I_{on-center} = \bigoplus_{i\in\{2,3,4\}} I^{[s_i]}_{on-center} \quad .$$

3.1.2 Color

Color feature maps are computed in terms of intensities for colors *red*, *green*, *blue* and *yellow*. Such choice of basis colors corresponds to human vision [44]. As original RGB image is converted into CIE LAB universal color space [44], Euclidean distance ρ_{LAB} between colors corresponds to human perception and can be used to calculate color intensity:

$$C^{color}_i(x,y) = \rho_{\mathrm{LAB}}(s_i(x,y), color) \quad .$$

Feature maps are computed for each basis color by applying center–surround mechanism and across–scale addition to corresponding pyramids:

$$C^{color}_{on-center} = \bigoplus_{i\in\{2,3,4\}} I^{[C^{color}_i]}_{on-center} \quad .$$

3.1.3 Orientation

Orientation feature maps highlight edges having basis orientations θ: 0°, 45°, 90° and 135°. Feature map for each orientation is computed by applying corresponding Gabor filter [44] of specified orientation to images of DoG pyramid:

$$O^\theta_i(x,y) = (G_\theta \star \mathrm{DoG}_i)(x,y) \quad .$$

Gabor filters simulate response of orientation–selective neurons in visual cortex[42]. Filtered pyramids are summed up via across–scale addition yielding four orientation maps:

$$O^\theta = \bigoplus_{i\in\{2,3,4\}} O^\theta_i \quad .$$

3.2 Saliency Map

Feature maps are then normalized and fused into a combined map. Normalization operator $\mathcal{N}(\cdot)$ is adopted from [4]:

$$\mathcal{N}(I)(x,y) = \frac{1}{\sqrt{m}} I(x,y) \quad ,$$

where m is a number of local maxima above threshold that was chosen to be 0.65% of map's global maximum. This is necessary to smooth maps having a lot of local maxima. Normalized maps are summed up with equal weights yielding a combined map:

$$M = \frac{1}{3}\mathcal{N}\left(I_{on-center} + I_{off-center}\right) + \frac{1}{3}\mathcal{N}\left(\sum_{color\in\{R,G,B,Y\}} \left(C^{color}_{on-center} + C^{color}_{off-center}\right)\right) + \frac{1}{3}\mathcal{N}\left(\sum_{\theta\in\{0^\circ,45^\circ,90^\circ,135^\circ\}} O^\theta\right) \quad .$$

Equal weights are used for simplicity reasons although it is known that variable weights are preferable [45, 46].

Example of an image and its saliency map is shown on Fig. 3.

Figure 3: Saliency map example. On the left: source image. On the right: final saliency map

3.3 Absolute Measure of Saliency

Saliency map contains relative values, but for purposes of descriptor filtering it is necessary to have absolute measure of saliency, reusable between images.

Two methods are proposed to solve this problem. First method is based on introducing a measure of attention equivalency. Second one is normalization of a saliency maps to a single scale.

3.3.1 Attention equivalency measure

To build measure of attendance it is convenient to divide an image into a set of attended points and a set of unattended points. Straightforward thresholding is ineffective because of saliency variations, even within a single object [47]. To overcome this difficulty a fuzzy set theoretic methods can be developed. Method employed for current research is a simplified variant of *fuzzy growing* [47].

Let $\Omega = \{g_k, k = \overline{0, L-1}, L = 256\}$ be a set of saliency values. Two fuzzy sets are defined: set of attended points B_A and set of unattended points B_U with membership functions (2) and (3) respectively.

$$\mu_A(g_k) = \begin{cases} 1 & g_k \geq a \\ \dfrac{g_k - u}{a - u} & u < g_k < a \\ 0 & g_k \leq u \end{cases} \quad , \quad (2)$$

$$\mu_U(g_k) = \begin{cases} 0 & g_k \geq a \\ \frac{g_k - a}{u - a} & u < g_k < a \\ 1 & g_k \leq u \end{cases} . \quad (3)$$

Parameters a and u in (2) and (3) are constants that determine optimal fuzzy 2–partition. Details on these calculations are to be found in [47]. Function $\mu_A(g_k)$ is an absolute measure of saliency.

3.3.2 Saliency Map Normalization

Less complicated way to introduce absolute saliency values it to normalize all saliency maps to a single scale. In this case saliency values can be compared. Further motivation for this kind of processing is provided by an attention conservation hypothesis, proposed in [48]. It claims that total amount of saliency is invariant: perception causes only redistribution of that amount among input stimuli (i.e. image pixels).

Neurobiological evidence is used to determine the total amount of saliency. It is known that fovea comprises about 1% of retinal size but is responsible for over 50% of information [42]. As saliency map is a $W \times H$ table with values in $[0, \ldots, 255]$, total attention amount is calculated as $I = \frac{(255 \times 1\%(W \cdot H))}{50\%}$. Value of I is then used to scale the map.

4. INTEREST POINT MATCHING

Five step procedure is performed to match a pair of images:

1. Saliency maps are computed for both images according to Sect. 3.. Either of methods described in Sect. 3.3 is used to build absolute measure of saliency.
2. Local interest points are detected, non–salient points are pruned; SURF descriptors are calculated for remaining.
3. One–to–one symmetric search is performed on descriptors to select a set of matched interest point pairs.
4. Outlying false matches are identified and filtered out.
5. Remaining pairs are weighted by their saliency. Weights are summed up yielding a similarity score. Images are considered as near–duplicates if similarity score exceeds a threshold.

4.1 Interest Point Detection and Description

SURF detector and descriptor [10] were used in current research. The motivation for such choice is two–fold. First, performance of SURF is proved to be equal or superior to performance of other methods, such as SIFT [2], PCA–SIFT [1] and GLOH [16], in independent evaluative studies [49, 10]. Second, its computational efficiency is significantly better in comparison with aforementioned methods. SURF was successfully applied in vision–based mobile robot navigation [50, 49] and handle recognition [51].

The purpose of detector is to find scale–invariant points. SURF detector is based on calculating approximate Hessian response for image points. Although calculating Gaussian response is optimal for scale–space analysis [52], it is shown in [10] that due to aliasing errors its actual performance is not as perfect as in theory. Equal performance can be achieved by approximating Gaussian with Hessian calculated with box filters (Fast–Hessian [10]). This processing can be very efficiently implemented on the basis of integral images [43].

After detection non–salient keypoints are pruned. Non–salient points are points with saliency $\bar{g}$ such that either $\mu_A(\bar{g}) = 0$ or $\bar{g} < SaliencyThreshold$ in case of normalization.

Example of pruning results is illustrated on Fig. 4. Number of points after filtration reduces to up to 1/10 of whole.

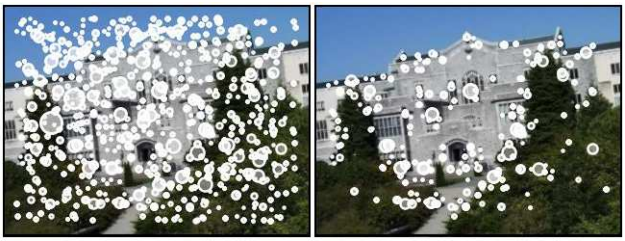


Figure 4: Attention–guided interest point filtering. On the left: image is shown with whole set of detected interest points. On the right: same image is shown with points remained after filtering.

Afterwards rotation invariant SURF descriptors (64–dimensional vectors) are computed for remaining interest points. Descriptor calculations are based on computing Haar wavelet responses in the vicinity of interest point that is implemented with integral images too. Exact details on above operations are given in [10].

4.2 Descriptor Matching

One–to–one symmetric search suggested in [6] is used in current research: given two sets of descriptors $\{P\}$ and $\{Q\}$ extracted from a pair of images (I_1, I_2), it returns pairs of closest descriptors. Algorithm 1 describes proposed search strategy.

Algorithm 1 One–to–one symmetric search

```
Given two sets of descriptors {P} and {Q}
for ∀P̄ ∈ {P} do
    Q̄ ⇐ NearestNeighbor(P̄, {Q})
    P* ⇐ NearestNeighbor(Q̄, {P})
    if P̄ == P* then
        if ρ(P̄, Q̄) < DistanceThreshold then
            pair (P̄, Q̄) is added to the result
        end if
    end if
end for
```

Function $NearestNeighbor(A, \{B\})$ in Algorithm 1 returns descriptor $\bar{B} \in \{B\}$ nearest to given A. Distance between descriptors is measured with Euclidean metric $\rho(P, Q)$. Value of $DistanceThreshold$ is experimentally chosen to be 0.2.

Straightforward implementation of $NearestNeighbor(A, \{B\})$ via exhaustive search is computationally prohibitive. To overcome this difficulty several solutions were proposed in literature: *K–d* trees [20], locality sensitive hashing [28], LIP–IS (local interest point index structure) [6], LIP–IS with inverted index [13]. For this work an extension of the latter approach is proposed.

Threshold–based matching is used instead of *nearest–neighbor–distance–ratio*, originally employed for SIFT descriptor matching [2], because explicit threshold value is necessary for chosen indexing approach. Furthermore it was found in experiments that keypoint filtering makes difference between results obtained with both algorithms almost negligible.

Basic idea behind LIP–IS is following. As soon as $\rho(P, Q)$ is most resource consuming part of matching procedure, performance can be improved if distance value will not be calculated for descriptor pairs such that $\rho(P, Q) \gg DistanceThreshold$. For this purpose

rough estimation of distance is computed. It is done with help of descriptor quantization. Original 64–bit vector of double values $P = (p_1, \ldots, p_{64})$ is transformed to $\hat{P} = (\hat{p}_1, \ldots, \hat{p}_{64})$, where $\hat{p}_i$ take on values from a discrete set $H = (h_1, \ldots, h_N)$. It is convenient to use $N = 8$, in this case distance can be estimated with fast bit operations.

Inverted index is used to further reduce sets of descriptors to compare — by indexing groups of descriptors P that have equal values in first k dimensions of corresponding $\hat{P}$ vectors. Then each descriptor is only compared to descriptors belonging to the same group in inverted index. As reported in [13] inverted index reduces matching time to 20–60% of original LIP–IS time. At the same time, it is built only once for each image and has very moderate memory footprint.

In this work an improvement to an inverted index is proposed. It was noticed during experiments that indexing first k dimensions often has a little effect because for most descriptors values corresponding to these dimension become equal after quantization. It makes inverted index useless in such cases. To get over this difficulty is was proposed to index k dimensions having maximum variance across set $\{B\}$. With this modification preliminary filtration time is steadily reduced to $20 \pm 5\%$ of original LIP–IS time. New structure was called *Maximum Variance Inverted Index (MVII)*.

4.3 False Match Identification

Due to quantization and overall descriptor design some degree of false matches can appear in nearest–neighbor search results. Outlying false matches can be identified relatively easy.

For each descriptor pair angle and distance between corresponding points are calculated. Then mean and standard deviation are computed for angles and distances across all matched pairs.

Finally, pairs such that difference between angle or distance from corresponding mean greatly exceeds respective standard deviation are considered to be false matches and are thereupon pruned. Figure 5 illustrates described procedure.

Geometrical verification techniques, such as RANSAC [27], used in several studies for same purposes (e.g. in [19]) are not employed in this research because simple procedure described above was found to be sufficient for images of test collection.

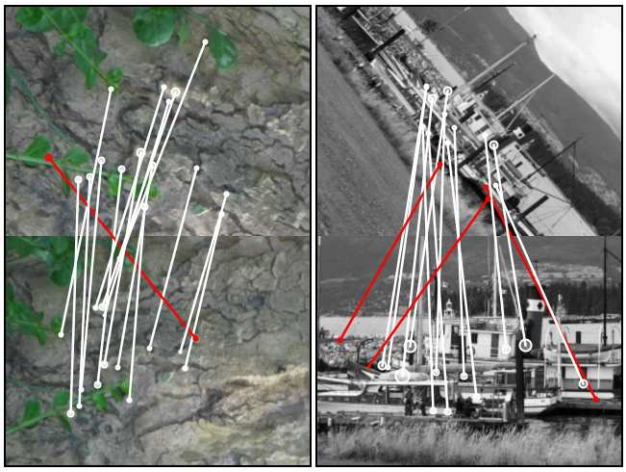


Figure 5: False match identification. White strokes depict correct matches, red strokes depict false matches.

4.4 Similarity Score

As soon as matched descriptor pairs are identified and false matches are pruned we need to make a decision of matching. For this purpose a similarity score is computed

$$S = \begin{cases} \frac{1}{255} \sum\limits_{(P,Q)\in N} (g_P + g_Q), \text{ in case of normalization,} \\ \sum\limits_{(P,Q)\in N} \left(\mu_A^{I_1}(g_P) + \mu_A^{I_2}(g_Q) \right), \text{ otherwise.} \end{cases} \quad (4)$$

In (4) N is a set of matched interest point pairs; g_P and g_Q are saliency values for points P and Q respectively; $\mu_A^{I_k}$ is membership function of attended points set computed for image I_k.

If similarity score (4) exceeds threshold $\bar{S}$ than images are said to be near duplicates. Threshold value $\bar{S}$ is a variable that determines tradeoff between recall and precision.

5. EXPERIMENTS AND RESULTS

Experiments were conducted to assess the performance of proposed image matching method against different approaches. Although various setups can be used for assessment (for instance, content–based image retrieval tasks and image collection clustering), in this research we have chosen to focus solely on image pair matching. Such decision is an attempt to reduce task specific bias, e.g. influence of clustering method on clustering results.

5.1 Data Set

Evaluation was carried out on real world images with different geometric and photometric transforms. We have adopted a data set from [16], that is widely used in comparative studies (e.g. in [10, 19, 20, 53]). Collection consists of 8 groups of near duplicates, 49 images in total, and is publicly available on the Internet[1]. Images are produced with lighting and viewpoint changes, blurring, zoom, rotation and compression. Sample images are shown on Fig. 1.

5.2 System Implementation

Software used for experiments was implemented in Java to simplify performance evaluation and analysis via code profiling. All tests were run on Intel Core 2 Duo 1.83 GHz machine with 2 Gb memory under Microsoft Windows XP operating system.

5.3 Experiment Setup

Four image matching methods based on SURF descriptors were compared in quality and efficiency tests:

1. Naive matching without any filtering.
2. Matching with contrast filtering proposed in [28]: top M keypoints with highest contrast value are selected for matching.
3. Attention–guided matching with threshold filtering: top M keypoints with highest saliency are selected for matching.
4. Attention–guided matching with similarity score (4).

Last method was tested with fuzzy measure and normalization. In case of normalization a variety of $SaliencyThreshold$ values were used to assess its influence on matching quality and speed.

Following procedure was performed to evaluate the quality of matching. Each image of test collection was compared with each of

[1] http://lear.inrialpes.fr/people/mikolajczyk/

Table 1: Performance evaluation results. N# denotes normalization with a given value of $SaliencyThreshold$.

Method	Average precision	Average Recall	Saliency map time (ms)	Detection time (ms)	Matching time (ms)
Naive matching	1.0	0.94	0	340	294
Top–contrast	0.98	0.89	0	340	24
Top–saliency	0.99	0.90	106	340	24
Similarity (Fuzzy)	0.91	0.81	320	300	210
Similarity (N30)	0.93	0.85	118	300	95
Similarity (N50)	0.91	0.85	118	160	40
Similarity (N70)	0.91	0.80	118	115	12
Similarity (N90)	0.92	0.67	118	95	3
Similarity (N110)	0.91	0.59	118	85	0.8

remaining images. Results of comparisons: true and false matches, were recorded. As soon as we know original groups of near duplicates, number of correct matches can be calculated for each group.

To assess accuracy of matching two metrics were computed for each group and in average: recall and precision:

$$recall = \frac{\text{\# correct true matches}}{\text{group size}} \ ,$$

$$precision = \frac{\text{\# correct true matches}}{\text{\# true matches}} \ .$$

Three metrics were also calculated to assess speed efficiency of each method: saliency map computation time, detection time and matching time. Saliency map computation time is spent to build saliency map and equivalency classes. Detection time is spent to detect interest points and compute their descriptors for a single image. Matching time is spent to match a pair of images: it includes time to build all relevant index structures. All metrics were calculated in average across all images and all pairs respectively.

For index efficiency test following methods were chosen: no indexing; LIP–IS; LIP–IS and inverted index; LIP–IS with MVII.

5.4 Results

Table 1 summarized method performances. Resulting recall–precision graphs are shown on Fig. 6. Average precision is almost unaffected by saliency threshold, however average recall decays as threshold value increases. Exact dependencies of accuracy and average matching time on saliency threshold are also demonstrated on Fig. 6. Analyzing these plots together we can see that variable saliency threshold allows to find a suitable tradeoff between accuracy and speed.

In comparison between fuzzy measure and normalization the latter approach is a clear winner. Experiments have shown that building optimal fuzzy partition of saliency map is computationally ineffective. At the same time performance of this approach does not exceed performance of normalization achieved with considerably lower costs.

Performances of contrast–based filtering and implemented alike attention thresholding are near equal. The reason for this is following: to reach acceptable accuracy levels we have to use top 300 interest points for both approaches. Experiments have shown that difference between set of top 300 points with highest contrast and set of top 300 points with highest saliency is negligible for images of test collection. Therefore we see nearly identical recall and precision. However attention thresholding is accompanied with additional costs because saliency map must be built. Thus top–contrast method is preferable.

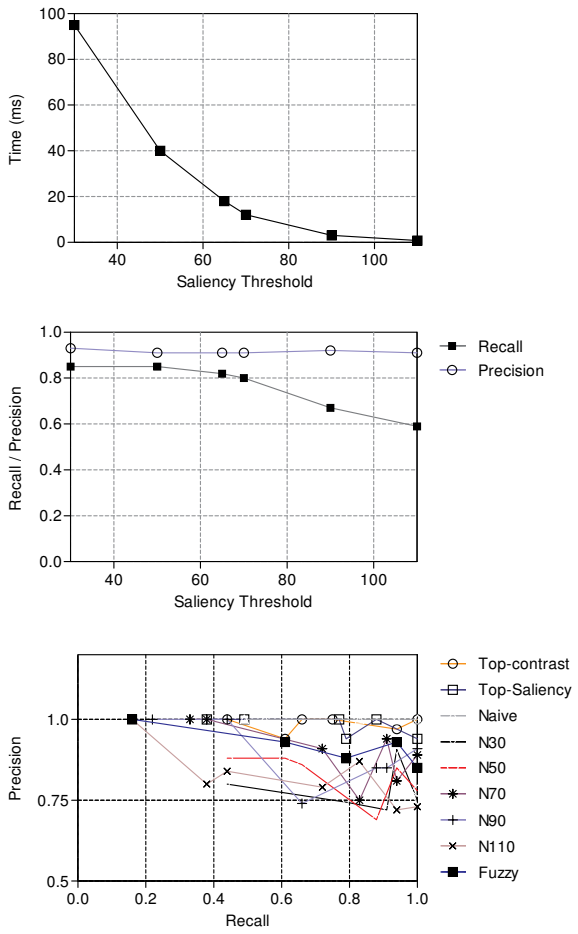


Figure 6: Evaluation results. Upper plot: dependency of average matching time on saliency threshold. Middle plot: dependency of accuracy metrics on saliency threshold. Lower plot: recall–precision graphs.

Table 2 contains obtained results that prove superior performance of LIP–IS with MVII in comparison with other techniques. We can see that original inverted index has a little effect on LIP–IS performance. We also rely on results reported in [6] indicating that LIP–IS outperforms locality sensitive hashing.

Table 2: Index structure efficiency.

Method	Matching time (ms)
No indexing	340
LIP–IS	28
LIP–IS + Inverted Index	24
LIP–IS + MVII	7

6. CONCLUSION

The paper proposed an image matching method based on visual attention and SURF keypoints. Biologically inspired visual attention system was used to guide local interest point detection and significantly reduce the number of interest points used in matching. Experimental results have shown attractive performance of new method in comparison with several different methods.

For time–critical tasks attention–guided matching based on normalization and *similarity score* is an attractive choice since it allows to increase matching speed by more that an order of magnitude (24.5 times for N70) with performance loss near 10% for average precision and average recall. At the same time, computations required to construct saliency map are a weak point of this approach. Although detection time decreases by up to 4 times, additional costs related to saliency map computation almost nullify this advantage.

In tasks where detection time is not constrained while matching time is critical, for instance, in content–based image retrieval tasks, *top–contrast* thresholding is most accurate method. But in cases where performance can be sacrificed for the sake of matching speed, *similarity score* methods can be applied as they reduce matching time by up to 30 times (for N110) from *top–contrast*.

Although saliency map computations are a weak point from speed efficiency standpoint, the use of visual attention has been proven as an effective method to achieve near real–time matching efficiency without significant loss of quality. Further research will be directed towards development of faster saliency map computation methods.

7. REFERENCES

[1] Y. Ke and R. Suthanakar, "Pca-sift: A more distinctive representation for local image descriptors," *Comput. Vis. and Pattern Recogn.*, vol. 2, pp. 506–513, 2004.

[2] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Intern. J. of Comput. Vis.*, vol. 60, pp. 91–110, 2004.

[3] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, 2007.

[4] S. Frintrop, *VOCUS: A Visual Attention System for Object Detection and Goal-Directed Search*, Ph.D. thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, 2006, `http://tinyurl.com/frintrop`.

[5] T. D. Barfoot, "Online visual motion estimation using fastslam with sift features," in *Proc. of IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems, 2005*, 2005, pp. 579–585.

[6] W. L. Zhao, C. W. Ngo, H. K. Tan, and X. Wu, "Near-duplicate keyframe identification with interest point matching and pattern learning," *IEEE Trans. on Multimedia*, vol. 5, no. 9, pp. 1037–1048, 2007.

[7] Y. Ke, R. Suthanakar, and L. Huston, "Efficient near-duplicate detection and sub-image retrieval," in *Proc. of ACM Multimedia Conf.*, 2004, pp. 869–876.

[8] X. Wu, C.-W. Ngo, and Q. Li, "Threading and autodocumenting news videos," *Signal Processing Magazine*, vol. 23, no. 2, pp. 59–68, 2006.

[9] O. Marques, L. M. Mayron, G. B. Borba, and H. R. Gamba, "An attention-driven model for grouping similar images with image retrieval applications," *EURASIP J. of Applied Signal Processing*, vol. 2007, no. 1, pp. 116–116, 2007.

[10] H. Bay, A. Ess, T. Tuytelaars, and Van Gool L., "Surf: Speeded up robust features," *Comput. Vis. Image Underst.*, vol. 3, no. 110, pp. 346–359, 2008.

[11] X. Zheng, M. Zhou, and X. Wang, "Interest point based medical image retrieval," pp. 118–124, 2008.

[12] P. Labatut, J.-P. Pons, and R. Keriven, "Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts," in *IEEE 11th Intern. Conf. on Computer Vision. ICCV 2007*, 2007, pp. 1–8.

[13] V. Pimenov, "Near-duplicate image detection with local interest point extraction," in *Proc. of the Sixth Russian Information Retrieval Evaluation Seminar, ROMIP'2008*, 2008, pp. 145–159, Available in russian at `http://tinyurl.com/pimenov08`.

[14] M. Awrangjeb and G. Lu, "An improved curvature scale–space corner detector and a robust corner matching technique for transformed image identification," *IEEE Trans. Image Process.*, vol. 17, no. 12, pp. 2425–2441, 2008.

[15] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, pp. 977–1000, 2003.

[16] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, 2005.

[17] L. Chen and F. W. M. Stentiford, "An attention based similarity measure for colour images," in *Proc. of 16th Intern. Conf. on Artificial Neural Networks ICANN 2006*, 2006, pp. 481–487.

[18] B. Thomee, M. J. Huiskes, E. Bakker, and M. S. Lew, "Large scale image copy detection evaluation," in *MIR '08: Proc. of the 1st ACM intern. Conf. on Multimedia Information Retrieval*, 2008, pp. 59–66.

[19] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Proc. of the 7th Eur. Conf. on Computer Vision*, 2002, pp. 128–142.

[20] K. Mikolajczyk and J. Matas, "Improving descriptors for fast tree matching by optimal linear projection," in *Proc.of IEEE Intern. Conf. on Computer Vision*, 2007, pp. 1–8.

[21] J. Shi and C. Tomasi, "Good features to track," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.

[22] D. Kim, K. Kim, J.-Y. Kim, S. Lee, and H.-J. Yoo, "An 81.6 gops object recognition processor based on noc and visual image processing memory," in *Proc. of IEEE Custom Integrated Circuits Conf.*, 2007, pp. 443–446.

[23] H.D. Chati, F. Muhlbauer, T. Braun, C. Bobda, and K. Berns, "Hardware/software co-design of a key point detector on fpga," in *Proc. of Intern. Symposium on Field–Programmable Custom Computing Machines*, 2007, pp. 355–356.

[24] S. Se, H. Ng, P. Jasiobedzki, and T. Moyung, "Vision based modeling and localization for planetary exploration rovers," in *Proc. of 55th Intern. Astronautical Cong.*, 2004, pp. 1–11.

[25] V. Bonato, E. Marques, and G. A. Constantinides, *Reconfigurable Computing: Architectures, Tools and Applications*, chapter A Parallel Hardware Architecture for Image Feature Detection, Springer, Heidelberg, 2008.

[26] J. Ferruz and A. Ollero, "Real-time feature matching in image sequences for non-structured environments. applications to vehicle guidance," *J. of Intelligent and Robotics Systems*, vol. 28, no. 1-2, pp. 85–123, 2000.

[27] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[28] J. J. Foo and R. Sinha, "Pruning sift for scalable near-duplicate image matching," in *ADC '07: Proc. of the 18th Conf. on Australasian Database*, 2007, pp. 63–71.

[29] W. Jia, H. Zhang, X. He, and Q. Wu, "A comparison on histogram based image matching methods," in *2006 IEEE Intern. Conf. on Advanced Video and Signal Based Surveillance*, 2006, p. 97.

[30] C. Koch, L. Itti, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, , no. 20, pp. 1254–1259, 1998.

[31] N.D.B. Bruce and J.K. Tsotsos, "Spatiotemporal saliency: Towards a hierarchical representation of visual saliency," in *5th Int. Workshop on Attention in Cognitive Systems*, 2008, pp. 98–111.

[32] C. Koch and S. Ullman, "Shifts in selective visual attention: Towards the underlying neural circuitry," *Human Neurobiology*, vol. 4, pp. 219–227, 1985.

[33] E. Niebur, "Saliency map," *Scholarpedia*, vol. 2, no. 8, pp. 2675, 2007.

[34] J. S. Hare and P. H. Lewis, "Scale saliency: Applications in visual matching, tracking and view-based object recognition," in *Proc. of Distributed Multimedia Systems 2003 / Visual Information Systems 2003*, 2003, pp. 436–440.

[35] T. Kadir and M. Brady, "Saliency, scale and image description," *Intern. J. of Comp. Vision*, vol. 2, no. 45, pp. 83–105, 2001.

[36] S. Gilles, *Robust Description and Matching of Images*, Ph.D. thesis, University of Oxford, 1998.

[37] P. Suau and F. Escolano, "Multi-dimensional scale saliency feature extraction based on entropic graphs," in *Proc. of the 4th Intern. Symposium on Advances in Visual Computing*, 2008, vol. II, pp. 170–180.

[38] S. Frintrop, "The high repeatability of salient regions," in *Proc. of ECCV Workshop "Vision in Action: Efficient Strategies for Cognitive Agents in Complex Environment"*, 2008.

[39] F. W. M. Stentiford, "An attention based similarity measure with application to content-based information retrieval," in *Proc. of the Storage and Retrieval for Media Databases Conf., SPIE Electronic Imaging*, 2003, pp. 221–232.

[40] L. Chen and F. W.M. Stentiford, "Comparison of near-duplicate image matching," in *Proc. of 3rd Eur. Conf. on Visual Media Production*, 2006, pp. 38–42.

[41] F. W. M. Stentiford, "Attention-based similarity," *Pattern Recognition*, vol. 40, no. 3, pp. 771–783, 2007.

[42] S.E. Palmer, *Vision Science, Photons to Phenomenology*, MIT Press, Cambridge, 1999.

[43] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. of the 2001 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, 2001, vol. 1, pp. 511–518.

[44] D.A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice Hall, Berkeley, 2003.

[45] H. Hügli and Bur. A., "Adaptive visual attention model," in *Proc. of Image and Vision Computing New Zealand*, 2007, pp. 233–237.

[46] B. Rasozadeh, A. Tavakoli Targhi, and Eklundh J.-O., "An attentional system combining top-down and bottom-up influences," in *Proc. of Intern. Workshop on Attention in Cognitive Systems*, 2007, pp. 123–140.

[47] Y.-F. Ma and H.-J. Zhang, "Contrast-based image attention analysis by using fuzzy growing," in *MULTIMEDIA '03: Proc. of the 11th ACM Intern. Conf. on Multimedia*, 2003, pp. 374–381, `http://tinyurl.com/yfma03`.

[48] A. V. Galsko, "The model of attention dynamics in perception process," *Zh. Vyssh. Nerv. Deiat.*, vol. 58, no. 6, pp. 738–754, 2008.

[49] E. Frontoni, A. Ascani, A. Mancini, and P. Zingaretti, "Performance metric for vision based robot localization," in *Robotics Science and Systems 2008, Workshop on Good Experimental Methodologies*, 2008.

[50] F. Dayoub and T. Duckett, "An adaptive appearance-based map for long-term topological localization of mobile robots," in *Proc. of the IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems*, 2008, pp. 3364–3369.

[51] E. Jauregi, E. Lazkano, J. M. Martínez-Otzeta, and B. Sierra, *European Robotics Symposium 2008*, chapter Visual Approaches for Handle Recognition, Springer, Heidelberg, 2008.

[52] J. Koenderink, "The structure of images," *Biological Cybernetics*, vol. 50, pp. 363–370, 1984.

[53] G. J. Burghouts and J.-M. Geusebroek, "Performance evaluation of local colour invariants," *Comput. Vis. Image Underst.*, vol. 113, no. 1, pp. 48–62, 2009.

ABOUT THE AUTHOR

Vitaly Pimenov is a Ph.D. student at Saint–Petersburg State University, Faculty of Applied Mathematics and Control Processes.

Grouping from motion using the medoid shift and topological relations

Alexey Chernyavskiy
State Research Institute of Aviation Systems (FGUP GosNIIAS), Moscow, Russia
achern@gosniias.ru

Abstract

We consider the problem of extracting various moving objects on an image by analyzing the topological relations induced by the positions of a sparse set of putative matching points on two images of one scene. Topological relations among triples of matched point pairs are found, and we formulate the task as a hypergraph clustering problem which is then performed by the medoid shift, a non-parametric mode-seeking algorithm similar to mean shift. The method finds the number of clusters automatically and filters out outliers. We report the performance of our method on a synthetic dataset.

Keywords: *topology, medoid shift, clustering, hypergraph, motion segmentation, image matching.*

1. INTRODUCTION

This work addresses the problem of analysis of visual motion. This problem arises in video surveillance as well in other cases when one needs to detect the image changes in order to establish correspondences between pixels across several frames. The image changes can then be interpreted and the 3D structure of the scene can be analyzed. In video tracking the optical flow field is usually available from the analysis of several consecutive frames. The situation is different when the observer moves as well, and when the period of time that elapses between two images is longer.

A common way of finding several independently moving regions of an image is to detect motions by fitting a motion model, such as a homography or fundamental matrix, to the sets of matching points [7]. The sets of points which are explained by the motion model are then removed from consideration and the process is repeated until all the motions are found [2]. In case of a large number of mismatches this approach is not feasible. Whenever a subset of points is removed from consideration, the ratio of noisy matches relative to all the remaining matches increases, and robust filtering methods such as RANSAC [6] become less and less efficient.

The formulation of the problem that is being solved in this work is the following. Given two images and a sparse set of possibly corresponding points cluster the points into groups that belong to separately moving regions of an image. We do so by finding topological relations among pairs of points. By analyzing the topological relations among points one can detect parallax effects which are important depth cues. The topological relations are written down as an affinity matrix of the point pairs, and this matrix is clustered using the medoid shift algorithm. Several independent motions can be recovered simultaneously and the method is robust to outliers.

In Section 2 we formulate the topological relations among pairs of possible point matches, many of which may be outliers. We show that triples of point pairs may be thought of as vertices of a hypergraph. The penalties assigned to a triple whenever at least one point pair within it violates the *sidedness* constraint are the weights of the hypergraph edges. In Section 3 we describe how the *medoid shift* [8], a feature space analysis method similar to the *mean* shift [4], may be used to cluster the affinity matrix derived from the hypergraph using distance values only. The application of medoid shift to hypergraph clustering is given in Section 4. Results on synthetic data are shown in Section 5.

2. TOPOLOGICAL RELATIONS

Topological relations are one of the most stable relations among points of an image. These relations do not change under a wide range of affine transformations that may be applied to an image, such as scaling, translation or rotation. That is why it has been suggested in [5], in the context of image matching, to check whether topological constraints are satisfied among pairs of points and detect outliers. Suppose that we are given two images to be compared and matched, and a set of N corresponding points $\Phi = \{(\mathbf{x}_1^i, \mathbf{x}_2^i)\}_{i=1,\dots,N}$, which are projections of some points in three-dimensional space onto two image planes. For each triple of points $(\mathbf{x}_m^1, \mathbf{x}_m^2, \mathbf{x}_m^3)$ belonging to the same m-th image a *sidedness* function may be computed:

$$side(\mathbf{x}_m^i, \mathbf{x}_m^j, \mathbf{x}_m^k) = \operatorname{sgn}\left(\det \begin{bmatrix} x_m^k - x_m^j & x_m^i - x_m^j \\ y_m^k - y_m^j & y_m^i - y_m^j \end{bmatrix} \right), \quad (1)$$

where $\mathbf{x}_m^i = (x_m^i, y_m^i)$ and the subscript denotes the number of the image (1 or 2), while the superscript denotes the number of the point. This function assumes the value of +1 if the point $\mathbf{x}_m^i$ is located to the right side with respect to the vector pointing from $\mathbf{x}_m^j$ to $\mathbf{x}_m^k$; otherwise the function takes the value of -1. If, for a triple of points $(\mathbf{x}_1^i, \mathbf{x}_1^j, \mathbf{x}_1^k)$ belonging to the first image and for the triple of corresponding points $(\mathbf{x}_2^i, \mathbf{x}_2^j, \mathbf{x}_2^k)$ located in another image, $side(\mathbf{x}_1^i, \mathbf{x}_1^j, \mathbf{x}_1^k) \neq side(\mathbf{x}_2^i, \mathbf{x}_2^j, \mathbf{x}_2^k)$, we say that the sidedness constraint is violated. In this way, it is possible to derive a penalty function for each point pair:

$$p(i) = \sum_{j,k \in \Phi \setminus i;\, j<k} \left| side(\mathbf{x}_1^i, \mathbf{x}_1^j, \mathbf{x}_1^k) - side(\mathbf{x}_2^i, \mathbf{x}_2^j, \mathbf{x}_2^k) \right|. \quad (2)$$

By denoting

$$h(\mathbf{x}^i, \mathbf{x}^j, \mathbf{x}^k) = \left| side(\mathbf{x}_1^i, \mathbf{x}_1^j, \mathbf{x}_1^k) - side(\mathbf{x}_2^i, \mathbf{x}_2^j, \mathbf{x}_2^k) \right|, \quad (3)$$

the formula (2) may be rewritten as

$$p(i) = \sum_{j,k \in \Phi \setminus i;\, j<k} h(\mathbf{x}^i, \mathbf{x}^j, \mathbf{x}^k). \quad (4)$$

The penalty function can be normalized by the maximal number of possible sidedness constraints that each point pair can violate: $p_N(i) = p(i)/[(N-1)(N-2)]$. As a result, the penalty of each pair will belong to [0,1]. An iterative topological filtering procedure has been proposed in [5], in which point pairs (putative matches) having the largest penalty are marked as outliers and removed from consideration. The penalty of the remaining point pairs is then recomputed and the process is repeated until all the pairs' penalties are below a user-specified threshold. Topological filtering has been successfully applied for solving various image matching problems ([5], [3]).

In this work we propose to use topological constraints for segmenting objects that have moved between the moments when two images of one scene have been taken. We assume that the motions which are present on the image may be considered as piecewise-rigid, and so, when one considers pairs of points belonging the surfaces that are moving in a similar way (automobiles, buildings), there are few cases of sidedness violations among such pairs of points. On the contrary, if for a triple of points some points belong to a moving object while other points are part of a structure undergoing a separate movement, then the sidedness constraint has a high likelihood of being violated. Therefore, it seems reasonable to introduce a measure of distance based on topological relations. Point pairs belonging to two views of the same moving will exhibit good topological intra-class affinity, while points belonging to differently moving segments of an image will show a high inter-class dissimilarity.

As can be seen from (3), the topological relations involve triples of points. Since one is able to compute the sidedness violations for all the sets of point triples, the information about the relations that exist among triples can be represented as a hypergraph. A *hypergraph* is a generalization of a graph, where edges can connect any number of vertices instead of just two as in an ordinary graph. Formally, a weighted undirected hypergraph is a pair $H = (X, h)$ where X is a set of vertices, and subsets of X containing k vertices are called hyperedges. The function $h: X^k \to \mathrm{R}^+$ associates non-negative weights to each hyperedge consisting of k vertices. Since the hypergraph is undirected the function h does not depend on the order of its arguments. While there are numerous ways of finding sub-structures within a hypergraph, a common practice is to find a way to reduce the hypergraph to an ordinary weighted undirected graph $G = (X, g)$ over the same set of vertices, with edge weights given by a function $g: X^2 \to \mathrm{R}^+$ of two variables. One of the methods used to approximate a hypergraph with a graph is called clique expansion [1]. For each hyperedge z consisting of k vertices, a k-clique (a completely connected graph on k vertices; $k = 3$ in the case of triadic topological relations) is considered. The task of approximating $h(z)$ is reduced to the task of assigning weights to each edge within the k-clique associated with hyperedge z. According to [1], the weights of the ordinary graph G are given by the formula

$$g(\mathbf{x}^i, \mathbf{x}^j) = \sum_{\mathbf{x}^i, \mathbf{x}^j \in z; \forall \mathbf{x}^k \notin \{\mathbf{x}^i, \mathbf{x}^j\}} h(z) \Big/ \mu(N,k), \tag{5}$$

where $\mu(N,k) = \dfrac{(N-2)!}{(k-2)!(N-k)!}$ is the number of hyperedges that contain a particular pair of vertices. In our case $k = 3$, and therefore $\mu(N,k) = N-2$. The information about the edge weights can be written as a $N \times N$ symmetric matrix $\mathbf{G}$. Having the distance matrix $\mathbf{G}$ on hand, the affinity matrix is defined by

$$\mathbf{W}(i,j) = \exp(-\mathbf{G}^2(i,j)/2\sigma^2) \tag{6}$$

with σ a free parameter.

In the next section a novel algorithm is demonstrated, which allows to cluster data based on its topology-based affinity and remove outliers using the medoid shift.

3. GRAPH CLUSTERING USING MEDOID SHIFT

In this work the task of motion segmentation from two images of a scene is solved by clustering putative matching points into separate sets using an affinity measure computed by using topological relations among triples of matched point pairs. The points in the images can be thought of as vertices of a weighted undirected graph, and the affinity matrix consists of weights associated with the edges of this graph. The graph clustering problem involves data clustering based on the affinity matrix.

There are many efficient methods for data segmentation that make use of eigendecompositions of the affinity matrix (see [9] for a review and analysis). Many of them require specifying the number of clusters explicitly, while in typical computer vision tasks the number of clusters is not known a priori. Although it is possible to infer the number of clusters by analyzing the structure of eigenvectors of the affinity matrix, it involves a complicated numerical method **Error! Reference source not found.**. That is why we turned to a non-parametric mode-seeking technique, the medoid shift, which does not require the number of clusters to be known. Instead, it automatically finds the number of clusters during execution.

The medoid shift [8] is similar to the mean shift, an algorithm which is widely used for data clustering in computer vision problems [4]. We will briefly review both methods. Given N samples denoted by the set $\{\mathbf{x}^i\} \in \mathrm{R}^d$, $i = 1,...,N$, Parzen kernel density estimation is used to evaluate the underlying distribution function at a point by

$$f(\mathbf{x}) = c_o \sum_{i=1}^{N} \Phi\left(\left\| (\mathbf{x} - \mathbf{x}^i)/\sigma \right\|^2 \right), \tag{7}$$

where $\Phi(\cdot)$ is a radially symmetric kernel function [4], c_0 is a positive scalar, and $\sigma > 0$ is the bandwidth. In addition, $\Phi(x)$ is the *shadow* of the kernel $\varphi(x)$, i.e. $\varphi(x) = -\Phi'(x)$.

Mode-seeking is the process of finding local maxima of the density of the data. It is assumed that modes are good candidates for being centers of clusters. During mode-seeking, each point is initially denoted by $\mathbf{y}^0$, and the set of intermediate points traversed on the way to the mode is denoted by $\{\mathbf{y}^k\} \in \mathrm{R}^d$, $k = 1,...,K$. Each step of mean shift moves along the direction of highest gradient from the current point. Given the current point $\mathbf{y}^k$, its position on the next iteration of the method is denoted by $\mathbf{y}^{k+1}$ and found according to

$$\mathbf{y}_{mean}^{k+1} = \arg\min_{\mathbf{y}} \sum_{i=1}^{N} \left\| \mathbf{x}^i - \mathbf{y} \right\|^2 \varphi\left(\left\| (\mathbf{x}^i - \mathbf{y}^k)/\sigma \right\|^2 \right). \tag{8}$$

By differentiating the above equation and setting the first derivative to zero, one can obtain the formula for the position update of a point:

$$\mathbf{y}_{mean}^{k+1} = \sum_i \mathbf{x}_i \varphi\left(\left\|(\mathbf{x}^i - \mathbf{y}^k)/\sigma\right\|^2\right) \Big/ \sum_i \varphi\left(\left\|(\mathbf{x}^i - \mathbf{y}^k)/\sigma\right\|^2\right). \quad (9)$$

In this way, the updated position of a point is the weighted mean of the sample points. The mean shift method converges when the position $\mathbf{y}_{mean}^{k+1}$ does not change over the course of several iterations. By applying the mean shift procedure to each of the sample points, one can obtain modes of the data. Points that converged to the same modes are said to be part of the same clusters. The data on which the mean shift (and medoid shift) operate do not have to be linearly separable or form clusters of spherical shape with sharp boundaries. This allows the mean shift to work well in various tasks of computer vision, such as image segmentation, discontinuity preserving smoothing [4], and object tracking.

The medoid shift [8] is the extension of mean shift. An advantage of the medoid shift over the mean shift is that it can be applied to cases when the notion of mean is not defined and/or the mean of data points cannot be readily computed. Medoid shift finds modes of the data even when only a distance measure between samples is defined. This is precisely the case in the current work. The medoid, which is an extension of the median in the one-dimensional case (it is defined as the most centrally located point in a set of samples), is always part of the initial dataset. The update rule (8) is transformed and becomes

$$\mathbf{y}_{medoid}^{k+1} = \arg\min_{\mathbf{y}\in\{\mathbf{x}_i\}} \sum_{i=1}^{N} \left\|\mathbf{x}^i - \mathbf{y}\right\|^2 \varphi\left(\left\|(\mathbf{x}^i - \mathbf{y}^k)/\sigma\right\|^2\right). \quad (10)$$

The difference between (8) and (10) is that in the former case the updated position is a data point taken from the *d*-dimensional search space, while in the latter case $\mathbf{y}_{medoid}^{k+1}$ is the data *sample* that minimizes the function. In other respects, the mode-seeking procedure is the same as in the mean shift algorithm. The medoid shift provides a straightforward way of handling the outliers. Typically, outliers are located 'far' (in terms of topology-based affinity) from other data, and they do not merge with any clusters other than the ones they are centers of. A simple threshold T on the cardinality of a cluster allows differentiating between good clusters, consisting of many elements, and small clusters (containing only one data point in the extreme case) which are deemed as outliers. In this work we used the value $T = 10$. Numerically, the medoid shift is much faster than the mean shift because the search space is greatly reduced. The numerical implementation of the medoid shift is straightforward (see [8] for details). We use (6) with $\sigma = 0.2$ for computing the matrix $\mathbf{W}$.

4. PROPOSED METHOD

Before proceeding to the outline of the proposed algorithm, it should be made clear that formula (4) is not optimal for computing the topology-based distances between pairs of putative matches because it is highly affected by contributions from point pairs that are located far away from the point pair under consideration. Suppose that one needs to compute the distance $g(i, j)$ between pairs $\mathbf{x}^i$ and $\mathbf{x}^j$. Suppose that these pairs belong to the same moving object or to the image background and are close to each other. A third point $\mathbf{x}_1^k$ and its probable match $\mathbf{x}_2^k$ may be outliers (mismatches) or pseudo-outliers (true matches that belong to another moving object) and this may lead to sidedness violation and affect the affinity between $\mathbf{x}^i$ and $\mathbf{x}^j$.

Assuming that moving surfaces that are to be segmented from the images are local, we propose to weigh points' contribution to the affinity measure according to their geometrical distance from the pair under consideration. Precisely, the modified formula for $g(i, j)$ is the following:

$$g(\mathbf{x}^i, \mathbf{x}^j) = \sum_{\mathbf{x}^i, \mathbf{x}^j \in z; \forall \mathbf{x}^k \notin \{\mathbf{x}^i, \mathbf{x}^j\}} f\left[h(z)\right] \Big/ (N-2), \quad (11)$$

where

$$f\left[h(z)\right] = \begin{cases} 0, & \text{if } d_m^{ijk}/d_m^{ij} > t, \\ 1, & \text{if } d_m^{ijk}/d_m^{ij} < t, \quad \forall m \in \{1,2\} \\ h(z), & \text{otherwise,} \end{cases} \quad (12)$$

and $d_m^{ij} = \left\|\mathbf{x}_m^i - \mathbf{x}_m^j\right\|$, $d_m^{ijk} = \min(d_m^{ik}, d_m^{jk})$, $m \in \{1,2\}$.

In other words, the sidedness violations arising from the points which are far away from the pair for which the topological distance is computed are ignored. The modification given by (11) and (12) enhances the structure of the affinity matrix, leading to better intra-class similarity and larger inter-class dissimilarity. In this work we have used the value $t = 2$. The distance matrix for various values of t is shown in Figure 2 for a test problem described in the next section.

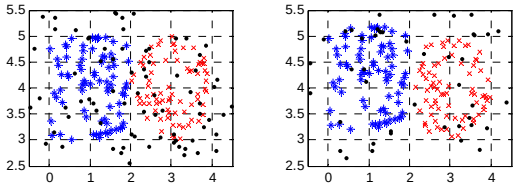


Figure 1: A synthetic example involving a square and a circle undergoing translation and rotation.

The outline of our algorithm is the following. Given σ, T, t, and N putative point matches of two images of the same scene :

1. Compute sidedness violations (3) for each triple of corresponding point pairs.
2. Compute the distance matrix using the sidedness violations and formulas (11) and (12). Compute the affinity matrix using (6).
3. Find modes of the data by running the medoid shift algorithm on the affinity matrix $\mathbf{W}$. Data points that converged to the same mode are part of a common cluster. Prune the clusters that contain fewer than T elements. They are considered as outliers. Clusters that contain more than T elements will correspond to surfaces of objects that undergo various movements.

5. RESULTS

We created a synthetic example consisting of two moving objects shown in Figure 1. The first object consists of 75 points randomly placed within a square of size 2. The second object is consists of 75 points randomly placed inside a circle of radius 1 centered at coordinates (3,4). In a second image, the square is moved by 0.2 in the y-direction, while the circle is rotated by 45° clockwise about its center. Our goal was to segment these two motions in the presence of 75 noisy points, a noise level of 33%.

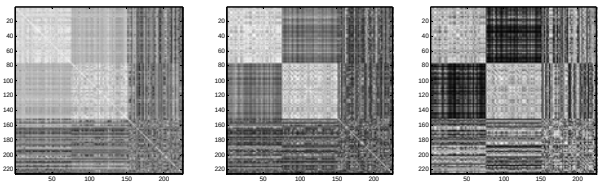


Figure 2: Distance matrix for the model from Figure 1 with various values of t : $t = 2$ (center) and $t = 1$ (right). The threshold t was not used in the left image. Lighter color indicates higher similarity among points.

In Figure 2 we illustrate the impact of the parameter t on the structure of the distance matrix. For illustration purposes, the points are numbered according to the number of object they are part of. When t is not used (Figure 2, left), the points exhibit a high similarity in-between clusters which makes it hard to differentiate between objects. When t is very small (Figure 2, right), the inter-cluster similarity is low, but the intra-cluster similarity becomes lower as well, since sidedness violations for even nearby points are ignored. We demonstrate in Figure 3 the two segmented objects (only the right frame is shown). Notice that while some points were incorrectly segmented, their number is low, and the number of correct points that were not thrown out is quite high. For comparison, we also show the results produced by the mean shift. For the mean shift, each data sample consisted of four parameters: the coordinates of each point in the first frame, and the two velocity components (difference of coordinates between the putative matches). The search domain for each parameter was normalized to lie between 0 and 1, and $\sigma = 0.1$ was used. A total of seven clusters were found, but much more outliers remain. This happens because both translation and rotation are present in the image, but the mean shift fails to take it into account.

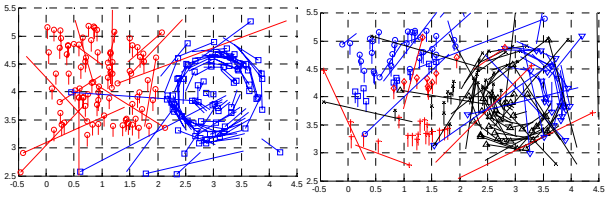


Figure 3: Two structures segmented by our algorithm (left) and seven clusters found by the mean shift (right).

6. CONCLUSION

In this work we have presented a method to extract various moving regions from topological relations among pairs of putative matches found using two images of a scene. The main contribution of our work is that we have formulated the topological relations in the contest of a hypergraph. The affinity matrix of the corresponding ordinary graph is then clustered using medoid shift. Clusters correspond to areas of the image that undergo independent motions, which is indicative of independently moving objects. Outliers (wrong matches) are detected as well. Compared to [5], the topological clustering introduced in this work leads to a smaller number of pairs that are wrongly marked as outliers, since it does not assume global scene rigidity.

The method was compared to a mean shift implementation that took into account only the shift vectors (velocities) of the matched points, without using the topological relations. Our method outperformed the mean shift implementation on synthetic data.

In the future we plan to validate our method on real datasets, as well as investigate the role of parameters σ and t which should be chosen dynamically based on local dataset density.

7. REFERENCES

[1] S. Agarwal, Lim Jongwoo, L. Zelnik-Manor, P. Perona, D. Kriegman and S. Belongie: Beyond pairwise clustering, In *IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, USA, pp. 838-845, 2005.

[2] P. Bhat, K.C. Zheng, N. Snavely, A. Agarwala, M. Agrawala, M.F. Cohen, M.F. Curless and B. Curless: Piecewise Image Registration in the Presence of Multiple Large Motions, In *IEEE Conference on Computer Vision and Pattern Recognition*, New York, USA, pp. 2491-2497, 2006.

[3] Yu.B. Blokhinov, D.A. Gribov and A.S. Chernyavskiy: Image matching problem for certain cases of perspective photography, *Journal of Computer and Systems Sciences International*, vol. 47, pp. 959-973, 2008.

[4] D. Comaniciu and P. Meer: Mean shift: a robust approach toward feature space analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 603-619, 2002.

[5] V. Ferrari, T. Tuytelaars and L. Van Gool: Wide-baseline multiple view correspondences, In *IEEE Conference on Computer Vision and Pattern Recognition*, Madison, USA, pp. 718-725, 2003.

[6] M.A. Fischler and R.C. Bolles: Random Sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Communications of the ACM*, vol. 24, pp. 381-395, 1981.

[7] R. Hartley and A. Zisserman: Multiple View Geometry in Computer Vision. – Cambridge University Press, 2004. - 672 p.

[8] Y.A. Sheikh, E.A. Khan and T. Kanade: Mode-seeking via Medoidshifts, In *IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, pp. 1-8, 2007.

[9] Y. Weiss: Segmentation using eigenvectors: a unifying view, In *IEEE International Conference on Computer Vision*, Kerkyra, Greece, pp. 975-982, 1999.

About the author

Alexey Chernyavskiy received his specialist degree in Applied Mathematics from Moscow State University in 2000. In 2003 he completed his M.S. degree in Geophysics at the University of Utah, USA. He now works at the State Research Institute of Aviation Systems (GosNIIAS) in Moscow, Russia. His scientific interests include image matching in the presence of large motions and pattern recognition. His contact email is achern@gosniias.ru .

Disparity Estimation in Real-Time 3D Acquisition and Reproduction System

Artem Ignatov, Victor Bucha, Michael Rychagov
Samsung Research Center,
1-st Brestskaya St. 29, Moscow, 125047, Russia
{a.ignatov, v.bucha, michael.rychagov}@samsung.com

Abstract

The paper concerns the method of disparity estimation and refinement based on iterative filtration of raw disparity estimate. Raw disparity estimate is obtained by conventional stereo-matching methods. In suggested algorithm, 6-8 iterations are sufficient to compute high-quality disparity map, suitable for virtual views rendering, which is essential step of any 3D reproduction system. The method is suitable for highly-parallel processing on modern GPU. In contrast to previous methods of stereo matching implemented on GPU, the proposed approach provides correct disparity computation for "problem areas", i.e. large uniform areas, occlusion areas and areas with repetitive patterns. Presented method was implemented in a real-time 3D acquisition and reproduction system. For GPU programming, an extension of C programming language provided by NVIDIA Compute Unified Device Architecture (CUDA) was used. Experimental results confirm the usefulness and robustness of the method.

Keywords: *disparity estimation, real-time processing, GPU computing, 3D reproduction, video processing.*

1. INTRODUCTION

Nowadays, humanity enters new era of digital television, i.e. the 3D television. The attention of academic and business environment is absorbed by the great opportunities for new application possibilities, which advanced techniques may provide. The 3D TV may bring an effect of real presence of participants in real-time video conferences. Also, the reality of computer games can be significantly improved by playing in 3D. Any synthetic video rendered by computer graphics applications can be immediately viewed on modern 3D auto-stereoscopic displays. However, for the real scenes, captured by stereo-camera or multi-camera setups there are still a lot of tasks to be solved. For example, for video-conferencing, all calculations should be computed in real-time. The necessary computations include cameras calibration, disparity estimation, and several views generation according to 3D reproduction device requirements. Cameras calibration could be performed off-line, if the cameras geometry is fixed. However, the disparity estimation of participants, and simultaneous view rendering should be performed on acceptable frame-rate, convenient to the users.

To cope with high volumes of calculations, some researches have tried to realize stereo-matching algorithms on modern GPU of video-boards. Recent works aiming stereo analysis proposed to exploit very efficient parallel Single Instruction Multiple Data (SIMD) architecture of modern GPU. Real-time implementation of computationally intensive methods of disparity estimation becomes possible thanks to SIMD architecture. J. Mairal et al. [1] reported that he achieved nearly video frame rate stereo reconstruction. The approach is from family of variational methods based on deformable models. Proposed method computes dense stereo from 3 cameras and entirely implemented on a GPU.

R. Yang et al. in [2], tried to force OpenGL toward conventional stereo computation, based on SAD window calculation with WTA optimization. The authors applied several OpenGL features for speeding-up of the algorithm, namely mip-map mechanism of texture generation for aggregation raw costs of different scale; p-buffers, which are the user-allocated off-screen buffers for fragment output. Unlike the frame buffers, they can be used directly as a texture, thereby eliminating excessive CPU memory read-ins.

One more example of using GPU for stereo analysis is the work of A. Brunton et all [3], which presents a novel implementation of Bayesian belief propagation for graphics processing units found in most modern desktop and notebook computers.

Most recent results of disparity estimation are obtained using NVIDIA CUDA technology. S. Grauer-Gray et all [4] described an efficient CUDA-based GPU implementation of the belief propagation algorithm that can be used to speed up stereo image processing and motion tracking calculations without loss of accuracy. Achieved acceleration in comparing with CPU realization is reported as a factor of five.

J. Gibson et al. [5] described how to accelerate the calculation of depth from stereo images by using a GPU. The CUDA was employed in novel ways to compute disparity using BT (Birchfield–Tomasi) cost matching. The challenges of mapping a sequential algorithm to a massively parallel thread environment and performance optimization techniques were considered.

2. REAL-TIME 3D ACQUISITION AND REPRODUCTION SYSTEM

We developed the system for acquisition of the dynamic 3D scene and simultaneous reproduction on auto-stereoscopic display (Figure 1). To manage the high demands on computation, the system uses parallel computation architecture of modern GPU and proposed method on disparity estimation was highly optimized for such architecture.

A system for three-dimensional video acquisition and reproduction includes a stereo content acquisition stage, disparity

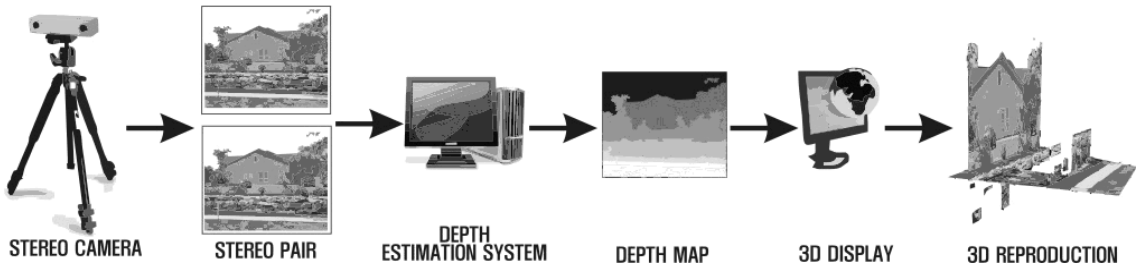


Figure 1: 3D acquisition and reproduction system

estimation stage, virtual view synthesis stage and 3D reproduction stage. The acquisition stage includes stereo cameras to acquire multiple video streams of dynamic scene. The disparity estimation stage includes computation units for real-time disparity estimation using obtained video streams. The view synthesis stage includes computation units for real-time virtual views synthesis on the basis of the computed disparity. The 3D reproduction stage includes computation units and 3D display for volumetric reproduction based on several virtual views generated on the previous stage.

The present system provides high-quality real-time 3D video acquisition and reproduction on 3D display.

Developed 3D acquisition and reproduction system includes stereo-camera Bumblebee 2 by Point Grey Research, modern PC with video-board NVIDIA GeForce 8800GTX, which has 128 stream processors, and auto-stereoscopic 3D monitor by Phillips.

For effective control, playback and record of 3D video, the corresponding software was developed. The acquisition part was written on the basis of Point Grey Research SDK. The stereo-matching, i.e. disparity/depth estimation was implemented using CUDA technology. This is the software/hardware architecture, which grants access to GPU stream processors for development of any tasks which could be effectively parallelized. Stereo-matching as image and video processing technique is the exemplar of application, which is well suited for the parallel-computation architecture. Since every pixel of image or video frame could be processed independently of other pixels. One of the main benefits of CUDA is a support of programmers-friendly environment for easy application development.

To decrease excessive memory reads/writes the CUDA application result was mapped to OpenGL Pixel Buffer Object (PBO) with simultaneous rendering on 3D monitor.

3. PROPOSED METHOD FOR DISPARITY ESTIMATION AND REFINEMENT

The dense disparity estimation problem consists of finding a unique mapping between the points belonging to the two images of the same scene (stereo pair). This is an ill-posed problem especially for textureless and occluded image areas. The mapping between corresponding points is called a *disparity vector*. In case of rectified geometry, the vector is scalar, and it is called a *disparity*. Thus, a *depth* is function of disparity with invert proportional dependence.

According to recent taxonomy [6], disparity estimation (stereo matching) algorithms generally perform the following four steps:

1. matching cost computation;
2. cost (support) aggregation;
3. disparity computation / optimization;
4. disparity refinement.

According to taxonomy, proposed method of disparity estimation concerns to the methods of disparity refinement. And could be applied for raw disparity estimate, obtained by disparity computation with matching cost calculation.

Proposed method relies on an idea of convergence from rough estimate toward the consistent disparity map through subsequent iterations of the disparity filter. On each iteration, the current disparity estimate is refined by filtration with accordance to images from stereo-pair. The reference image is defined as a color image from a stereo-pair, for which the disparity is estimated. And, the matched image is defined as other color image from the stereo-pair.

The disparity filter applies weighted average of neighboring pixels to current pixel of disparity map. Pixels, which are participated in filtration, are defined as *disparity reference pixels*. Corresponding pixels of color image are defined as *color reference pixels*. And pixels which are mapped by disparity values of color reference pixels are defined as *target pixels*.

In previous approaches the filter weights were reflected by proximity and similarity measures [7-8]. Proximity measure assigns weight of reference pixel, based on distance between current and reference pixels in spatial domain. While similarity measure assigns weight of reference pixel based on similarity between pixels of color image corresponding to reference and current disparity pixels.

During our development we concluded that proximity measure is less important, than similarity one. Since the filter usually operates in small local area of the image, the spatial measure constraint could be defined implicitly. And more attention should be paid on color similarity evaluation.

We propose the following methods for similarity computation.

1. Similarity computation with two pixel area comparison.
2. Similarity computation with one pixel area comparison.
3. Similarity computation with two single pixels comparison.

First two methods of similarity computation are concern to pixel area comparison rather then comparison of single pixels. This is done to strengthen the pixel similarity criterion. We studied that similarity computation based on pixel areas comparison gives more visually pleasant results, rather than individual pixels comparison.

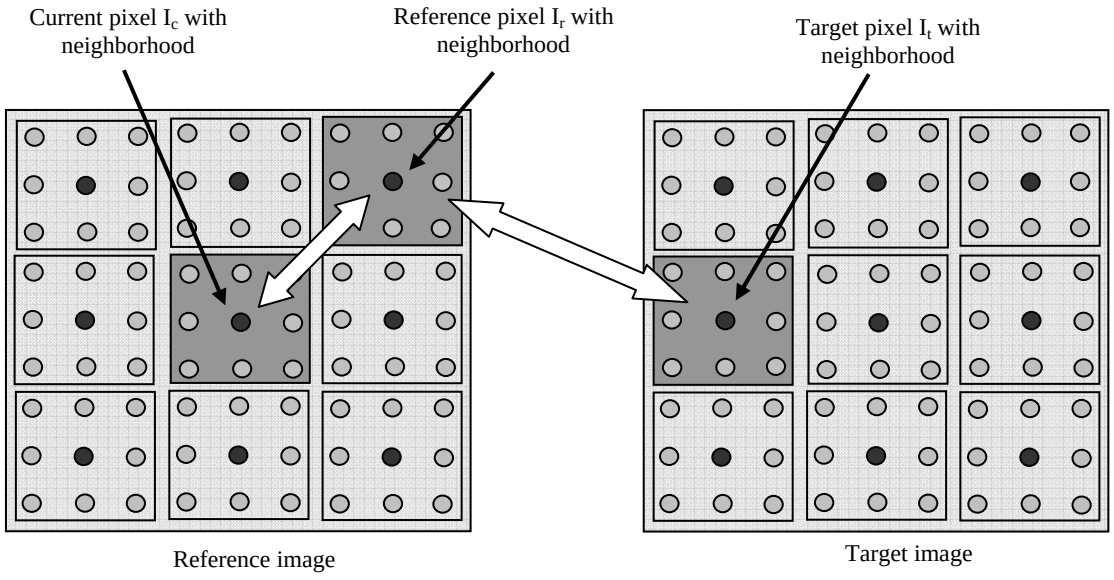


Figure 2: Pixels similarity measure computation**:** a) two comparisons of pixel neighborhoods, b) one comparison of pixel neighborhoods

In the first method of similarity computation, the weight is twofold and reflects the degree of similarity of current pixel with reference ones and target ones. First comparison is done between current pixel neighborhood with reference pixel neighborhood. The second one is carried out between reference pixel neighborhood with target pixel neighborhood (Exemplars of comparisons are shown in Figure 2 a) by wide white arrows). In this case the weights of disparity filter are computed as follows

$$w_r = e^{\frac{-C_1(x_r,y_r)}{\sigma_r} - \frac{C_1(x_t,y_t)}{\sigma_t}}, \qquad (1)$$

where C_1 () stands for a function used for pixel neighborhood comparison, σ_r and σ_t are the parameters of filter strength adjustment.

The second method of similarity computation is based on single comparison of pixel neighborhood areas of current and target pixels (Exemplar of comparison is shown in Figure 2 b) by wide white arrow). This type of similarity computation assumes that the current pixel is more or like similar with reference one due to their proximity to each other. And penalty is given only when the reference pixel is mapped to target pixel which is not similar with current pixel. In this case the weights of disparity filter are computed as follows

$$w_r = e^{\frac{-C_1(x_t,y_t)}{\sigma_t^2}},$$

where $C_1()$ stands for a function used for pixel neighborhood comparison. It is defined as

$$C_1(x_r, y_r) = \frac{1}{N \cdot M} \times \times \sqrt{\sum_{T\in\{R,G,B\}} \sum_{j=-\lfloor N/2 \rfloor}^{\lfloor N/2 \rfloor} \sum_{i=-\lfloor M/2 \rfloor}^{\lfloor M/2 \rfloor} \left(I_T(x_{c+i}, y_{c+j}) - I_T(x_{r+i}, y_{r+j}) \right)^2}$$

where $I_T(x_c, y_c)$ stands for current pixel intensity with coordinates x_c and y_c for color channel T, $I_T(x_r, y_r)$ denotes the reference pixel intensity with coordinates x_r and y_r for color channel T, N and M – are pixel area dimensions.

The third method of similarity computation is derived from the first one, setting pixel area dimensions to one. The difference from the first method is reflected in following equation for disparity filter weight computation

$$w_r = e^{\frac{-C_2(x_r,y_r)}{\sigma_r} - \frac{C_2(x_t,y_t)}{\sigma_t}}. \quad (2)$$

In comparison with Eq. 1, Eq. 2 uses different function for pixel comparison, which is defined as

$$C_2(x_r, y_r) = \sqrt{\sum_{T\in\{R,G,B\}} \left(I_T(x_{c+i}, y_{c+j}) - I_T(x_{r+i}, y_{r+j}) \right)^2},$$

where $I_T(x_c, y_c)$ stands for current pixel intensity with coordinates x_c and y_c for color channel T, $I_T(x_r, y_r)$ denotes the reference pixel intensity with coordinates x_r and y_r for color channel T.

According to selected method for similarity computation between pixels in filter window the disparity map at k-th iteration is given as

$$d_k(x_c, y_c) = \frac{1}{Norm} \cdot \sum_{s=-\lfloor K/2 \rfloor}^{\lfloor K/2 \rfloor} \sum_{p=-\lfloor L/2 \rfloor}^{\lfloor L/2 \rfloor} w_r \cdot d_{k-1}(x_r, y_r),$$

where $d_k(x_c, y_c)$ stands for the disparity map at k-th iteration for current pixel with coordinates (x_c, y_c), $d_{k-1}(x_r, y_r)$ denotes the disparity map at (k-1)-th iteration for reference pixel with coordinates $(x_r = x_{c+p}, y_r = y_{c+s})$, w_r denotes the weight of reference pixel, normalization factor is computed as

$$Norm = \sum_{s=-\lfloor K/2 \rfloor}^{\lfloor K/2 \rfloor} \sum_{p=-\lfloor L/2 \rfloor}^{\lfloor L/2 \rfloor} w_r \cdot$$

4. GPU IMPLEMENTATION OF PROPOSED METHOD

Proposed method was implemented for running on GPU. The following modifications of the algorithm were considered:

1. Adaptation of filter strength according to iteration;
2. Filter kernel size estimation according to iteration;
3. Separable processing of image rows and columns;
4. Histogram-based implementation of post-processing median filter.

Figure 3 represents algorithm flow-chart. According to flow-chart, the first step of the algorithm is to compute a raw disparity estimate. This could be done by utilization of conventional stereo-matching methods based on window correlation computation. We used SAD metric in 5x5 window. The exemplar of raw disparity estimate is presented in Figure 7 a). It is worth to mention high level of noise presented in raw disparity maps. Such noise will cause eye-fatigue when viewing this content on? auto-stereoscopic displays.

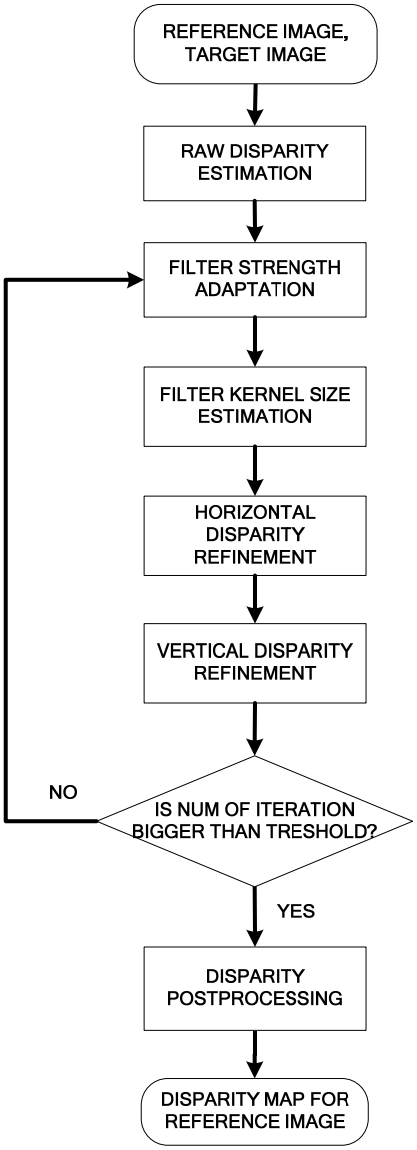


Figure 3: Disparity estimation algorithm flow-chart

After computation of raw disparity, the algorithm proceeds to series of iteration of disparity filter. On each iteration, filter strength (σ_r and σ_t are the parameters) is adjusted according to following formula

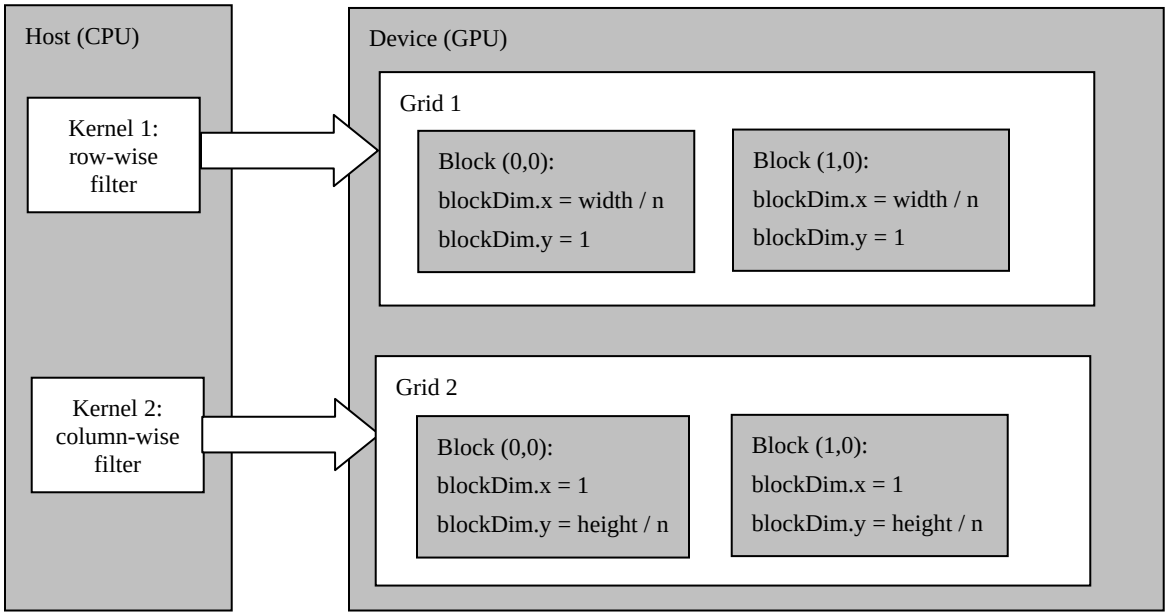


Figure 4: Computational grid arrangement for CUDA framework

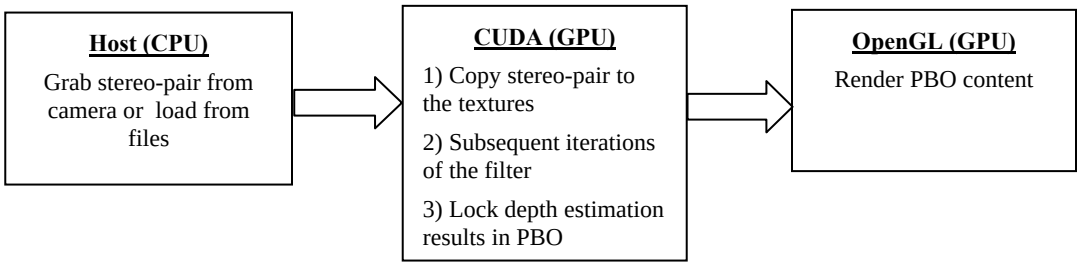


Figure 5: Data flow in proposed implementation

$$\sigma(k) = a_1 \cdot k + b_1, \quad (3)$$

where k is a number of iteration, a_1, b_1 are the linear coefficients.

After filter strength adaptation the algorithm for disparity refinement estimate the filter kernel radius by following formula

$$R(k) = a_2 \cdot k + b_2, \quad (4)$$

where k is a number of iteration, a_2, b_2 are the linear coefficients.

Since we did not interested in abrupt change of parameters from iteration to iteration, a linear dependency has provided us with simple solution to make parameters adaptable.

The idea behind the adaptation of parameters is the following. Since the reliability of source disparity is low, the algorithm starts with smallest filter strength and smallest kernel size. Then, the filter strength and kernel size are increased from iteration to iteration, allowing more resembling pixels to participate in filtration process. Also the adaptation of filter radius leads to decreasing of computational cost. This will be shown by concrete example in Section 5.

After adaptation of parameters the disparity refinement filter is applied. For speeding up of calculation the disparity filter could be defined in separable manner formulated in two passes. The first pass is row-wise processing. The second pass is column-wise one. The parameters of computational grid for CUDA framework is represented in Figure 4. The figure shows separable configuration of disparity processing. The host side resided on CPU has two kernels, which invoke corresponding two-dimensional grids of threaded blocks on the device (GPU). In the first grid, the blocks are organized to process rows of image. Parameters of blocks are set as (blockDim.x = width / n, blockDim.y = 1), where *n* is usually set to *halfwarp* value. This is HW-dependent parameter and it designates a number of stream processors on the multiprocessor. We set this value to 16, since we have used the NVIDIA GeForce 8800GTX. For the future generation of graphics boards this value will increase.

Since the method was implemented in GPU, it gave us the possibility of immediate rendering of disparity estimation results through CUDA – OpenGL interoperability option. More precisely, the disparity estimation results with reference image were locked in pixel-buffer object by CUDA. Then the OpenGL

part of program used that buffers for rendering. This operation has excluded additional data flow to CPU and again to GPU for rendering. Proposed data flow is described in Figure 5. Note that we used textures for storing the stereo-pair. The memory latency during fetching from texture is less, than from global memory. This happens due to texture caching.

5. EXPERIMENTAL RESULTS

Figure 6 represents result of disparity estimation from live stereo-video, while Figure 7 presents the result of disparity estimation for "Tsukuba" image from Middlebury image dataset.

Table 1 and Table 2 summarize objective evaluations of proposed methods. Two metrics were considered: root-mean square (RMS) distance to ground truth and bad pixel metric. Bad pixel is considered as the one, which differs from ground-truth pixel for 1 value. In the tables, the "bad_pixels_all" parameter represents the number of bad pixels in all image, while the "bad_pixels_nonocc" parameter corresponds to the number of bad pixels in non-occluded areas.

For "Tsukuba" image 4 types of filter were considered. The first three of them are classified according to method of similarity computation. For these experiments the radius and filter weight were kept constant for all iterations. The fourth method is the third method with adaptive parameters setting, according to Eq. 3 and Eq. 4.

According to Table 2, the best result for the "Tsukuba" was achieved when using third method with normalization. The resulted disparity map contains only 4 % of bad pixels. Here normalization means the forcing disparity values to neighboring disparity level. This operation decreases number of bad pixel about 1.5-2 times. However, during rendering into auto-stereoscopic display, the visual difference was not captured between normalized and non-normalized results. In our opinion the bad pixel metric does not reflect the quality of disparity properly, since resulted disparity map could have strong distortions of objects shape, and have small value of bad pixels simultaneously.

Proposed methods increase the quality of disparity map in occlusion areas also. This confirms by results in Table 1 and Table 2 (The "rms_error_occ" and "bad_pixels_occ" parameters correspond to value of RMS and number of bad pixels in occlusion areas). According to tables the number of bad pixels in occlusion areas is decreased by half in comparison with raw disparity map. And the RMS is improved by almost 3 times. Correct handling of disparity discontinues, especially in occluded regions greatly facilitates virtual view rendering. The comparative results of state-of-the-art stereo-matching methods and explanation of comparison parameters could be found in [6].

The visual quality of rendering results on the auto-stereoscopic monitor Phillips was nearly the same for 1-st, 3-rd and 4-th methods, while 2-nd method shows the artifacts in the area of lamp and bust. The 2-nd requires less computation than 1-st method. The method works well for smooth areas, while for strong depth discontinues it fails to compute disparity correctly.

To decrease computational burden without degrading in quality the adaption of filter radius was introduced. This means that the computational complexity of 4-th method is lower. This could be illustrated by following calculation. Since we used 6 iterations with radius 20 for the 1-st and 3-rd methods, this requires the 6 x 20 = 120 computations of weights. For the 4-th method with adaptive parameters $a_2 = 5$ and $b_2 = 1$, we get the following computation for the 6 iterations: 1+6+11+16+21+26 = 81. The computation of weight includes calculation of two color distances with exponent. So decreasing the number of weight computation introduces theoretical performance boost about 30% (1 – 81/120). This confirms by software execution time also (Table 3).

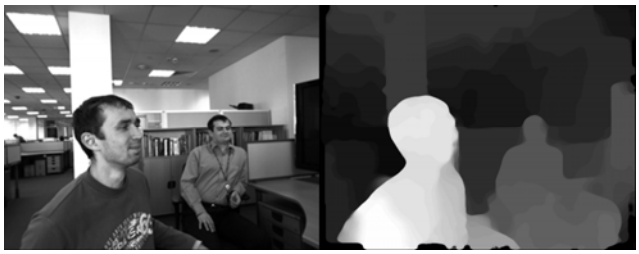


Figure 6: Live 3D capture from stereo camera:
Reference image with correspond disparity

Table 3 shows that maximum performance gain was achieved for 3-rd method. GPU implementation of the method executes 656 times faster than CPU one with float point arithmetic. Also the adaptation of radius in the 4-th method has lead to the least execution time with only 18 msec for frame. This corresponds to 50 fps. Correspondingly, the MDE (Million disparity estimation, MDE = width*height*disp_levels*fps) for tsukuba image with width=384, height =288 and disp_levels = 16 is equal to 97. At the same time, proposed approach shows higher throughput for real-time stereo-matching, and MDE is equal to 294 in this case.

6. CONCLUSION

Recent multi-view 3D displays require multiple views to be synthesized and rendered for 3D scene reproduction. Such virtual views can be generated from stereo content and disparity map. The key factor of high quality virtual view generation is a usability of disparity map which determined by the correct disparities values.

Proposed method of disparity estimation outputs disparity map of high quality computed on TV frame rates. Thus, developed technique could be used for various real-time applications, such as immersive videoconferencing or augmented reality.

7. REFERENCES

[1] J. Mairal, R. Keriven and A. Chariot, "Fast and efficient dense variational stereo on GPU", Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06), pp. 97-104.

[2] R. Yang, M. Pollefeys, "Multi-resolution Real-Time Stereo on Commodity Graphics Hardware" CVPR 2003, pp 211-218.

[3] A. Brunton, C. Shu, G. Roth "Belief Propagation on the GPU for Stereo Vision", Proceedings of the 3rd Canadian Conference on Computer and Robot Vision (CRV'06) 0-7695-2542-3/06, 2006.

[4] S. Grauer-Gray, C. Kambhamettu, and K. Palaniappan "GPU Implementation of Belief Propagation Using CUDA for

Cloud Tracking and Reconstruction", 5th IAPR Workshop on Pattern Recognition in Remote Sensing (PRRS 2008).

[5] J. Gibson and O. Marques "Stereo Depth with a Unified Architecture GPU", Computer Vision and Pattern Recognition Workshops (CVPRW 2008), pp. 1 – 6.

[6] D. Scharstein and R. Szeliski. "A Taxonomy and Evaluation of Dense two-frame stereo correspondence algorithms", IJCV, volume 47(1), pp. 7-42, 2002.

[7] K. J. Yoon and I. S. Kweon "Adaptive Supprot-Weight Approach for Correspondence Search" IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, No 4, April 2006

[8] F. Boughorbel "Adaptive Filters for Depth from Stereo and Occlusion Detection", Stereoscopic Displays and Applications XIX, Proceedings of the SPIE, Volume 6803, 2008.

Table 1. Comparative results for Tsukuba image

Evaluation criteria*	**Raw depth** (SAD 5x5 window)	**1-st method** (2 pixel areas comparison, radius 20)	**2-nd method** (1 pixel area comparison, radius 10)	**3-rd method** (2 pixels comparison radius 20)	**4-th method** (with adaptive parameters a_2=5, b_2=1)
rms_error_all	1.89	0.955	1.163	0.865	0.998
rms_error_nonocc	1.732	0.931	1.143	0.832	0.976
rms_error_occ	5.029	1.62	1.743	1.694	1.619
bad_pixels_all	0.154	0.099	0.159	0.096	0.135
bad_pixels_nonocc	0.137	0.09	0.149	0.086	0.126
bad_pixels_occ	0.826	0.444	0.573	0.458	0.478

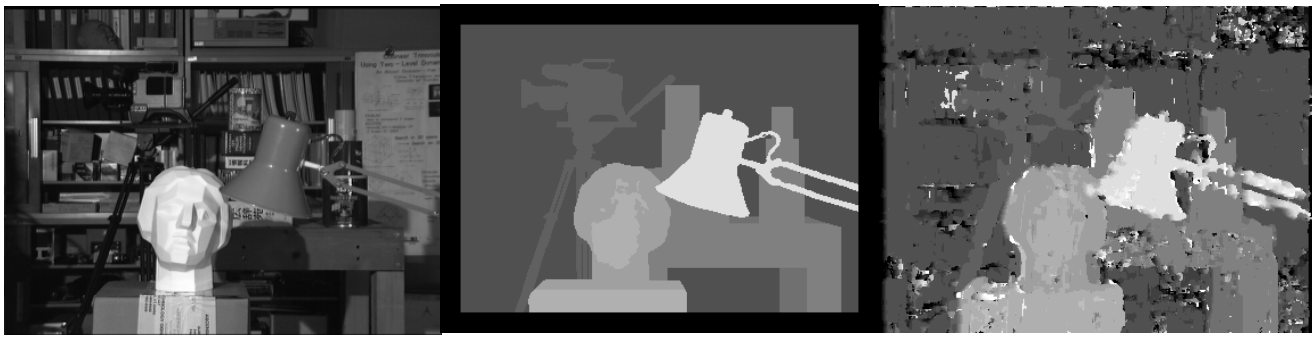
*) For explanation of the evaluation criteria, i.e. "rms_error_all" etc. see [6]

Table 2. Comparative results for Tsukuba image with normalization

Evaluation criteria	**Raw depth** (SAD 5x5 window)	**1-st method**	**2-nd method**	**3-rd method**	**4-th method**
rms_error_all	1.89	0.972	1.176	0.932	1.039
rms_error_nonocc	1.732	0.946	1.156	0.899	1.018
rms_error_occ	5.029	1.683	1.777	1.775	1.646
bad_pixels_all	0.154	0.073	0.092	0.042	0.073
bad_pixels_nonocc	0.137	0.065	0.083	0.034	0.065
bad_pixels_occ	0.826	0.382	0.434	0.323	0.361

Table 3. Performance analysis of Tsukuba image stereo matching

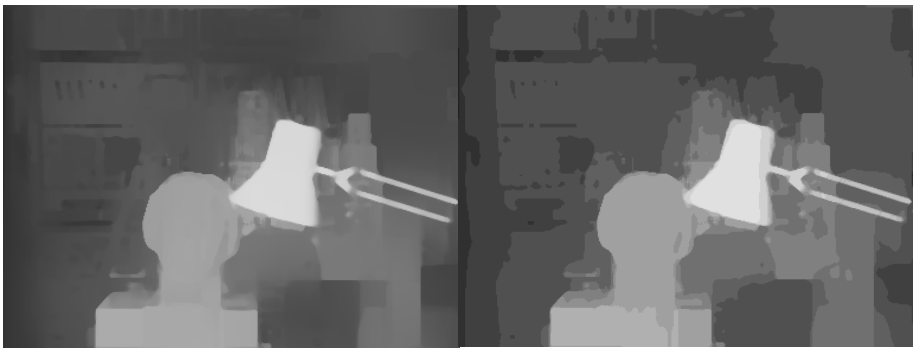
	Time (msec)			
	1-st method	**2-nd method**	**3-rd method**	**4-th method**
GPU implementation (nVidia 8800GTX)	3031.5	504.47	24.5	18.33
CPU implementation– 1 core (INTEL Core2Quad Q6600, 2.4 GHz)	720305.5	126290.6	16069.2	11024.32
Performance gain (times)	237	250	656	601



a)

b)

c)



d)

Figure 7: Experimental results for Tsukuba image **a)** color image, ground truth, raw depth (SAD with 5x5 aggregation window) (from left to right) **b)** proposed 1-st, 2-nd 3-rd methods according to Table 1 (from left to right) **c)** proposed 1-st, 2-nd 3-rd methods with normalization (from left to right) **d)** proposed 4-st method without and with normalization (from left to right)

Occlusion Handling in Trinocular Stereo using Composite Disparity Space Image

Mikhail Mozerov, Ariel Amato and Xavier Roca
Computer Vision Center, Departament Informatics
Universitat Autònoma de Barcelona, Barcelona, Spain
mozerov@cvc.uab.es

Abstract

In this paper we propose a method that smartly improves occlusion handling in stereo matching using trinocular stereo. The main idea is based on the assumption that any occluded region in a matched stereo pair (middle-left images) in general is not occluded in the opposite matched pair (middle-right images). Then two disparity space images (DSI) can be merged in one composite DSI. The proposed integration differs from the known approach that uses a cumulative cost. A dense disparity map is obtained with a global optimization algorithm using the proposed composite DSI. The experimental results are evaluated on the Middlebury data set, showing high performance of the proposed algorithm especially in the occluded regions. One of the top positions in the rank of the Middlebury website confirms the performance of our method to be competitive with the best stereo matching.

Keywords: Stereo vision, graph cut techniques, trinocular stereo.

1. INTRODUCTION

Stereo is a fundamental problem for a wide variety of tasks in computer vision [1]. A numerous of approaches have been proposed to solve the problem [2-5], but still a satisfying solution has not been received, since the stereo matching problem is an ill-posed one. Recently an excellent progress has been made in stereo matching due to the effectiveness of the global optimization techniques [6-9]. Nevertheless, occlusion remains a key problem in stereo matching.

Occlusion handling is one of the most important parts of many stereo matching techniques, and ignoring the occlusion can spoil the disparity map estimation by a sensitive inaccuracy. Recently many approaches were proposed to overcome the negative effect of occlusion in stereo matching [12-16]. The last taxonomy of such approaches can be found in the work [2].The constraints that are typically used in the process of occlusion handling are: ordering constraint, uniqueness constraint visibility constraint. The visibility constraint can also be considered a parameter to be recovered for each point of the stereo image frame like the desired disparity value itself. Formally, there exist two different cases of occlusion handling: two frame stereo and multi-frame stereo. However, from the reported researches, it is difficult to recognize the principal difference between these two approaches (see e.g. [10] and [13]). Nevertheless, it is necessary to realize that the two frames stereo differs fundamentally from the multi-frame stereo. For the one baseline stereo system invisibility of some region of points in one stereo image relative another point of view is almost inevitable norm. Hence, the task of occlusion handling in this case is to localize the stereo image regions where the matching between two frames has no sense, see a zoomed region in Fig. 1. Consequently, such information can be used for the further interpolation of the dense disparity map in these particular areas, or for modification of the data term in the energy minimization problem, like it has been done in the work [10].

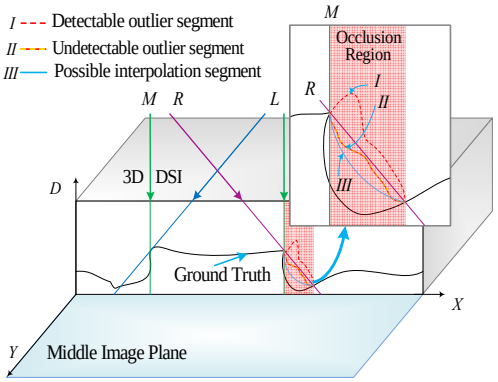


Figure 1: Occlusion region in 3D DSI.

However, even if we suppose that the right disparity map (R in Fig. 1.) is recovered perfectly in the left occluded region it is not guarantee that the occluded region can be detected. Indeed, detectable outlier in the left disparity map (I segment in Fig. 1.) makes a part of truly recovered right disparity map invisible. In contrast, for the multi baseline stereo it is feasible (except some special cases) to match every points in the middle frame with one of the neighbor frames, left or right. As a result, in general, reliable matching yields more accurate disparity map estimation than the interpolation. The problem is that without knowing the desired disparity map it is impossible to distinguish to which neighbor frame left or right the ambiguous region should correspond. Such a speculation force to use iterative approach: recover coarse disparity map and hence visibility then update data term for the further step of the algorithm iteration.

To avoid an iteration process it was proposed to use different cumulative costs for the multiple view matching [15]. This idea was adopted for occlusion handling in the work [16]: authors proposed to use acentric windows that improve algorithms with cumulative costs in the occluded regions. However such approach supposes to work with augmented costs that integrates pixelwise matching dissimilarity in a neighbourhood (of the image space or of a view sequences). Consequently, the pre-processing step smoothes edges of the resultant disparity map. This is why global optimization techniques are preferable versus local windows methods.

In this work we propose essentially new approach, which allows ridding of iterative process and in the same time avoiding negative consequences of the cumulative integration. To achieve this goal we introduce a composite 3D DSI (some researchers call it correlation volume), which is logical superposition (in contrast with cumulative summation) of two single DSI: the middle frame to the left frame DSI and the middle frame to the right frame DSI. Then a global optimization algorithm uses the prepared 3D array of the matching cost values to recover optimal disparity map. The rest of this paper is organized as follows: Section 2 describes the proposed composite DSI. Section 3 describes the full algorithm of the dense disparity map recovering. Section 4 is devoted to the experimental results. Concluding remarks are made in Section 5.

2. COMPOSITE DSI

The DSI representation is very popular in stereo matching [1, 2] due to the clear geometric interpretation of this model. Indeed, the desired disparity map should coincide with one of all the possible surfaces in the DSI. Furthermore, the global optimization approach assumes that an integral of the initial cost values plus inter-pixel smoothness term over such a surface should satisfy a chosen optimality criterion. For the trinocular stereo the 3D approach assumes that DSI has dimensions row $0 \le x \le X_{max}$, column $0 \le y \le X_{max}$, and disparity $0 \le d \le D_{max}$. All the three stereo images suppose to be rectified, each element (x, y, d) of the DSI projects to the pixel (x, y) in the middle image and to the pixel $(x-d, y)$ and $(x+d, y)$ in the left image and in the right image respectively. Let $E_{ML}(x, y, d)$ denote the DSI_{ML} cost value assigned to element (x, y, d) of the middle to left images and $E_{MR}(x, y, d)$ denote the DSI_{MR} cost value assigned to element (x, y, d) of the middle to right images matching space. All cost values (or pixelwise distance) are calculated using one of the convenient pixel-to-pixel matching metrics. In our work Euclidian distance between two compared color vectors is considered:

$$E_{ML}(x,y,d) = \left|\mathbf{I}_M(x,y) - \mathbf{I}_L(x+d,y)\right|, \qquad (1)$$
$$E_{MR}(x,y,d) = \left|\mathbf{I}_M(x,y) - \mathbf{I}_R(x-d,y)\right|,$$

where $\mathbf{I}_M(x,y)$, $\mathbf{I}_L(x,y)$ and $\mathbf{I}_R(x,y)$ are color vectors values of the middle left and the right stereo images respectively. Let us consider a very simple synthetic example: a square foreground patch on the plane background Figs. 2-4.The DSI related to the middle to left stereo matching (images a,b of Fig. 2) has an uncertainty region due to occlusion, a white triangle segment in the left part of Fig. 2 (d). A disparity map in Fig. 2. (e) obtained by a simple dynamic programming algorithm is indeed optimal, but not coincide with the ground truth of this pair. The same problem arises in the case of the middle to right stereo matching. One of possible way to solve occlusion problem in this case is to try merging two resultant disparity maps, but such a fitting might involve edge analysis or other iterative methods.

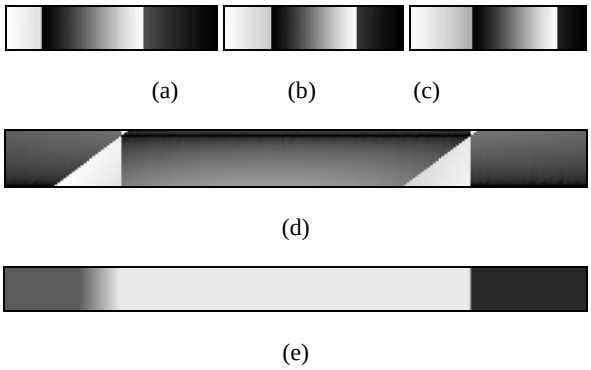


Figure 2: (a,b,c) – left middle and right synthetic images; (d) - a 2D slice of DSI_{ML}; (e) – an optimal disparity map related to DSI_{ML}.

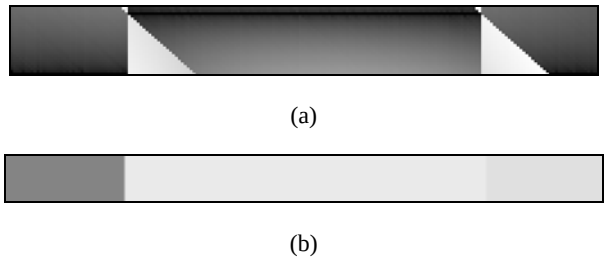


Figure 3: (a) - a 2D slice of DSI_{MR}; (b) – an optimal disparity map related to DSI_{MR}.

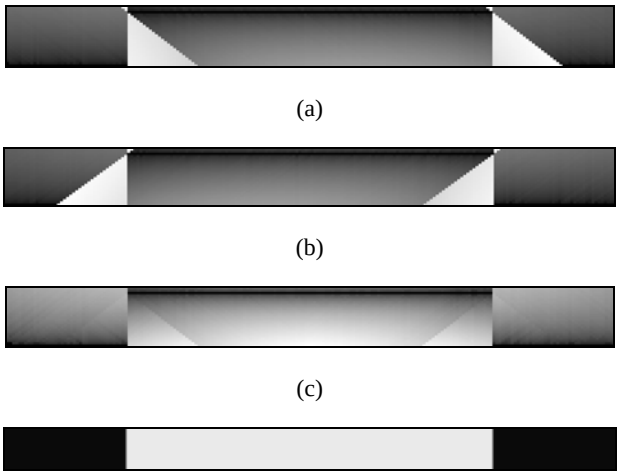
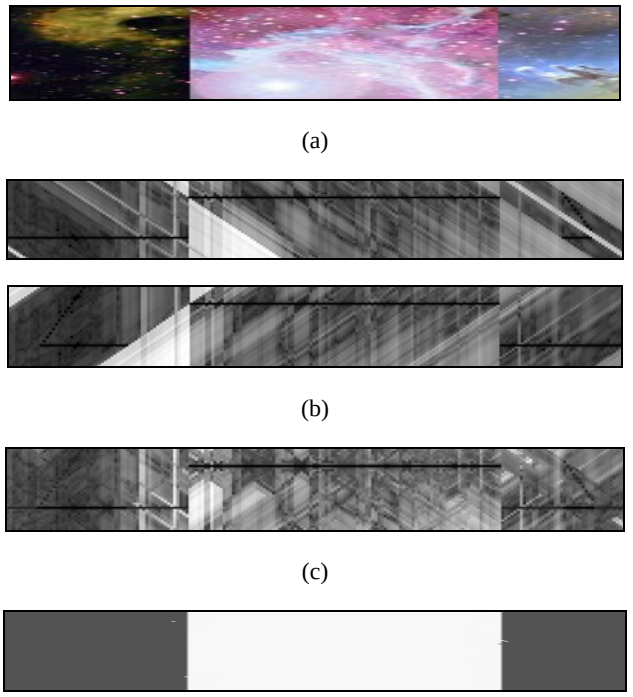


Figure 4: (a) - A 2D slice of DSI_{MR} ; (b) - a 2D slice of DSI_{ML} ; (c) - a 2D slice of composite DSI_C; (d) – an optimal disparity map related to DSI_C.

We propose a smart solution: merge two DSI before apply a global optimization algorithm (see Fig. 5.) in such a way that the resultant cost value of the composite DSI is going to be the minimum of two initial cost values of related E_{MR} and E_{ML}

$$E_C(x,y,d) = \min\left\{E_{ML}(x,y,d),\ E_{MR}(x,y,d)\right\}, \qquad (2)$$

were $E_C(x, y, d)$ is a cost function of the composite DSI.



(a) (b) (c) (d)

Figure 5: (a) - Middle image of synthetic stereo; (b) – are two 2D slices of DSI_{MR} ; and 2D slice of DSI_{ML} ; (c) – is a 2D slice of composite DSI_C; (e) – is an optimal disparity map related to DSI_C.

The advantage of such an approach is obvious, we solve a global optimization problem just once, and furthermore, the resultant solution in general coincides with ground truth and does not have to be corrected. To confirm our intuitive guess by visual representation we propose to consider another synthetic example with more rich texture on the reconstructed surface that is illustrated in Fig. 6. The black line of minimal costs coincides with ground truth, but for the DSI_{MR} and the DSI_{MR} this line is interrupted inside occlusion regions, in contrast in the line is has no uncertainty regions, see Fig. 5. (c).

One delicate moment in our method is the viewing geometry: the three camera centers lie at a single straight line perpendicular to the single viewing direction, the distance between middle left and middle right cameras are strictly equal. Such geometry is used for example in two base line stereo camera (e.g. Point Grey Research Inc.). To extend such a configuration over more general optical setup it is necessary just to preserve one line camera centers constraint. Of course, if optical axes of the cameras are not parallel the reciprocal homography of the three stereo images has to be done. The baselines inequality also can be solved in this case by a calibration process, but we have to note, that a considerable difference in the baseline values in trinocular stereo complicate a lot all calculations, due to inverse proportionality of the scene depth versus its image disparity. This is the reason, why we do not involve matching information of the left-right images pair in the reconstruction process: at the first glance the reconstruction accuracy for such pair is higher than for middle-left and middle-right, nevertheless DSI of this pair is not coincide with composite DSI.

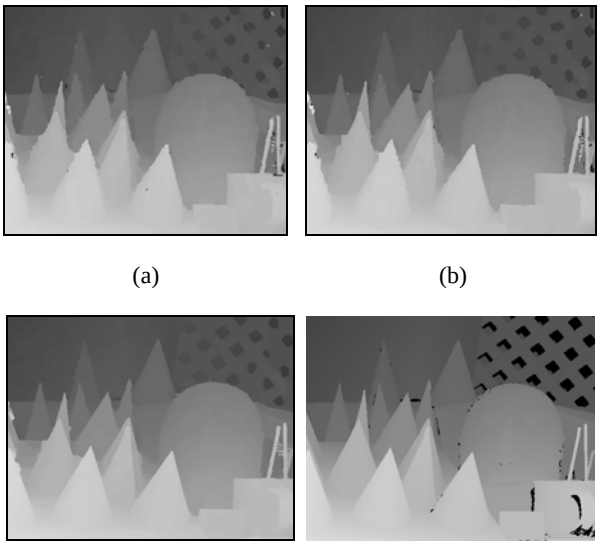
Of course, to obtain a good result with real stereo images it is necessary to use all standard steps of stereo matching, like preprocessing, global optimization and post processing. In the next section we describe our algorithm more precisely. However we have note that the solution of any 2D global optimization problem is always a tradeoff between accuracy and computational complexity. There exist a lot of papers which consider this problem, but the main goal of this work is to show the advantage of using composite DSI and we do not pretend in this paper to contribute in the field of MRF energy minimization.

3. MATCHIMG ALGORITHM

First step of our algorithm consists of calculation a composite DSI. This step was described in the previous section and based on Eqs. (1) - (2). Second step include standard preprocessing or filtering of the composite DSI. The most important part of our algorithm is the solution of the global optimization problem. It is natural to assume that the best matching of stereo images is achieved with the disparity map $P(x,y)$, which minimizes the objective function related with the composite DSI

$$O(P(x,y)) = \sum_{x,y,d \in P(x,y)} E_C(x,y,d) + \sum_{x,y,d \in P(x,y)} \sum_{i,j \in \Omega 1} S(x+i,y+j,d), \qquad (3)$$

where S is a smoothness term. Thus, we can formulate our problem as follow: find the disparity map function $P(x,y)$, which minimize the objective function in Eq. (3)



(a) (b) (c) (d)

Figure 6: (a) – Reconstruction only by using the global optimization step; (b) – reconstruction with preprocessing ; (c) – disparity map rectified with weighted median filter. (d) – ground truth of Cones.

$$P(x,y) = \arg\min_{P} O\left(P(x,y)\right). \qquad (4)$$

To solve the global optimization problem we use Graph Cut method [8], which is a modification of the very well known α-

expansion method proposed by Boykov at al. [6].

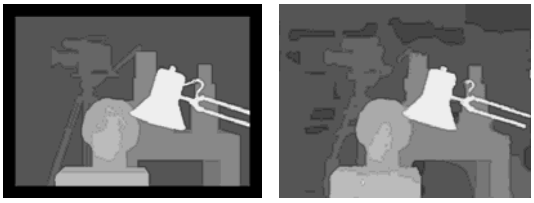
We also included an additional step for the resulting disparity map rectification. This step consist of the weighted median filtering

$$P_F(x,y) = \underset{i,j\in\Omega}{med}\left\{w_{i,j}P(x+i,y+j)\right\}; \qquad (5)$$

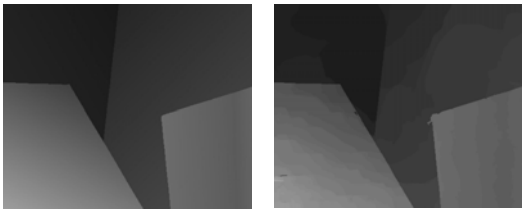
where Ω is a neighborhood region and w_{ij} is a weight factor that depends on the Euclidian distance between points (x,y) and $(x+i,y+j)$ and also depends on the Euclidian distance in the RGB color space between color vectors $\mathbf{I}_M(x,y)$ and $\mathbf{I}_M(x+i,y+j)$. The impact of each step of the algorithm is illustrated in Fig. 6.

4. EXPERIMENTAL RESULTS

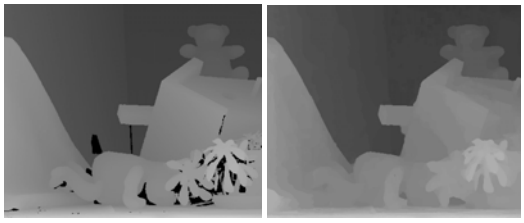
In this section experimental results are presented. In our experiments we used the stereo images benchmark from the Middlebury data set. Fig. 8. shows the corresponding disparity map obtained by our approach. The comparison with other stereo matching algorithms is sown in Table I. The first number in each column shows the error rate of the method, and the second number denotes the rank. Our method is in yellow. If the error threshold is chosen equal to 1.5 instead of traditional 1 like it is in second part of Table I, the performance of our method put the obtained result to the top of the rank. The ability of the proposed algorithm can be verified by the software that might be provided as additional material.



(a)



(b)



(c)

Figure 7: Middlebury benchmark test images: right column are the disparity maps obtained by the proposed algorithm, left column ground truth (a) – Tsukuba; (b) – Venus; (c) –Teddy.

The smoothness term in (3) is a cut linear function

$$S(x+i,y+j,d) = \lambda\begin{pmatrix}\left(\left|d(x,y)-d(x+i,y+j)\right|-1\right)\times\\ \times H\left(g-\left|d(x,y)-d(x+i,y+j)\right|\right)+1\end{pmatrix}; \qquad (6)$$

where $H()$ denotes Heaviside step function, parameters λ and g can set experimentally to optimize the output result, but as the rule of thumb we use λ is equal to mean value of the error term in (3) $\langle E_C(x,y,d)\rangle$ and the cut threshold g is equal to 3. Anyway, in [7] one can find the reason to choose one or another MRF energy representation for the objective function in (3).

Table I: The rank in Middlebury website evaluation.

Error Threshold = 1 (Error Threshold...) — Sort by nonocc / Sort by all / Sort by disc

Algorithm	Avg. Rank	Tsukuba ground truth nonocc	Tsukuba ground truth all	Tsukuba ground truth disc	Venus ground truth nonocc	Venus ground truth all	Venus ground truth disc	Teddy ground truth nonocc	Teddy ground truth all	Teddy ground truth disc	Cones ground truth nonocc	Cones ground truth all	Cones ground truth disc	Average Percent Bad Pixels
CoopRegion [41]	3.6	**0.87** 1	**1.16** 1	**4.61** 1	0.11 2	0.21 2	1.54 3	5.16 8	8.31 5	13.0 7	2.79 5	7.18 2	8.01 6	4.41
AdaptingBP [17]	3.7	1.11 8	1.37 4	5.79 9	**0.10** 1	0.21 3	**1.44** 1	4.22 3	7.06 3	11.8 4	**2.48** 1	7.92 4	7.32 3	4.23
DoubleBP [35]	4.5	0.88 3	1.29 2	4.76 3	0.13 4	0.45 8	1.87 6	3.53 2	8.30 4	**9.63** 1	2.90 6	8.78 11	7.79 4	4.19
YOUR METHOD	4.8	0.99 7	1.30 3	5.29 7	0.20 9	0.38 7	2.27 10	5.03 6	**6.37** 1	11.7 3	2.67 2	**4.03** 1	7.16 2	3.95
OutlierConf [42]	5.3	0.88 2	1.43 6	4.74 2	0.18 8	0.26 4	2.40 11	5.01 5	9.12 7	12.8 6	2.78 4	8.57 7	**6.99** 1	4.60
SubPixDoubleBP [30]	7.2	1.24 13	1.76 16	5.98 10	0.12 3	0.46 9	1.74 5	**3.45** 1	8.38 6	10.0 2	2.93 7	8.73 10	7.91 5	4.39
AdaptOwrSegBP [33]	11.8	1.69 25	2.04 23	5.64 8	0.14 5	**0.20** 1	1.47 2	7.04 17	11.1 9	16.4 13	3.60 13	8.96 10	8.84 12	5.59
SymBP+occ [7]	12.6	0.97 5	1.75 14	5.09 5	0.16 6	0.33 5	2.19 8	6.47 11	10.7 8	17.0 17	4.79 27	10.7 23	10.9 22	5.92

Error Threshold = 1.5 (Error Threshold...) — Sort by nonocc / Sort by all / Sort by disc

Algorithm	Avg. Rank	Tsukuba ground truth nonocc	Tsukuba ground truth all	Tsukuba ground truth disc	Venus ground truth nonocc	Venus ground truth all	Venus ground truth disc	Teddy ground truth nonocc	Teddy ground truth all	Teddy ground truth disc	Cones ground truth nonocc	Cones ground truth all	Cones ground truth disc	Average Percent Bad Pixels
YOUR METHOD	3.2	0.99 8	1.30 3	5.29 9	0.10 2	0.19 2	1.33 3	2.19 4	**3.38** 1	6.40 3	**1.65** 1	**2.85** 1	**4.91** 1	2.55
CoopRegion [41]	4.7	**0.87** 1	**1.16** 1	**4.61** 1	0.11 7	0.20 4	1.54 8	3.00 6	5.02 4	8.80 7	2.33 6	6.37 3	6.89 8	3.41
AdaptingBP [17]	4.8	1.08 9	1.33 5	5.79 10	0.10 4	0.20 3	1.42 5	**1.91** 1	3.63 2	**6.05** 1	2.18 5	7.06 6	6.52 7	3.11
DoubleBP [35]	4.8	0.88 3	1.29 2	4.76 4	0.11 5	0.42 13	1.47 6	2.03 2	5.46 6	6.38 2	1.98 2	7.50 10	5.92 3	3.18
SubPixDoubleBP [30]	5.6	0.92 4	1.33 4	5.00 6	0.11 5	0.42 12	1.47 6	2.03 3	5.59 6	6.57 4	2.02 3	7.56 11	6.04 4	3.26
OutlierConf [42]	5.9	0.88 2	1.43 7	4.74 3	0.16 11	0.22 5	2.17 12	2.95 5	5.73 7	8.42 6	2.05 4	7.27 7	5.58 2	3.47
AdaptOwrSegBP [33]	9.3	1.26 16	1.51 8	4.69 2	0.12 8	**0.16** 1	1.38 4	5.05 16	8.01 9	12.3 13	2.51 11	7.59 12	7.20 11	4.31

5. CONCLUSION

We propose a method to handle occlusion in stereo matching using trinocular stereo. The main advantage of the approach is that we solve a global optimization problem just once, and the resultant solution does not have to be corrected in the occluded regions. Three stereo images are used instead of two, thus competition with two image matching is not correct. Anyway, accumulation of information in a short baseline stereo does not automatically leads to more accurate results. Probably it is the reason why there are not a lot of announced comparisons with the Middlebury data set for more than two images. At least we are not aware about such evaluations.

6. ACKNOWLEDGEMENTS

This work has been supported by EC grant IST-027110 for the HERMES project and by the Spanish MEC under projects TIC2003-08865 and DPI-2004-5414. M. Mozerov acknowledges the support of the Ramon y Cajal research program, MEC, Spain.

7. REFERENCES

[1] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *International Journal of Computer Vision,* 47(1/2/3):pp. 7-42, 2002.

[2] Y. Ohta and T. Kanade, “Stereo by intra – and intra-scanline search using dynamic programming,” *IEEE Trans.on PAMI*, 7(2): 139-154, 1985.

[3] S. Roy and I. J. Cox. “A maximum-flow formulation of the N-camera stereo correspondence problem,” *Proc. Int’l Conf. Computer Vision*, :492-499, 1998.

[4] H. Zhao. “Global optimal surface from stereo,” *Proc. Int’l Conf. Pattern Recognition*, 1:101-104, 2000.

[5] C. L. Zitnik and T. Kanade. A cooperative algorithm for stereo matching and occlusion detection. IEEE Trans. Pattern Analysis and Machine Intelligence, 22(7): 675-684, 2000.

[6] Y. Boykov, O. Veksler, and R. Zabih. “Fast approximate energy minimization via graph cuts,” *IEEE Trans.on PAMI*, 23(11): 1222-1239, 2001.

[7] V. Kolmogorov and R. Zabih. “What energy functions can be minimized via graph cuts,” *IEEE Trans.on PAMI*, 26(2): 147-159, 2004.

[8] J. Sun, N.N. Zheng, and H.Y. Shum, “Stereo matching using belief propagation,” *IEEE Trans.on PAMI*, 25(7):787-800, 2003.

[9] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. “Efficient Belief Propagation for Early Vision,” *International Journal of Computer Vision*, 70(1):41-45, 2006.

[10] J. Sun, Y. Li , S.B. Kang, H-Y. Shum, “Symmetric stereo matching for occlusion handling,” *CVPR05*, (2):399-406, 2005.

[11] Y. Wei, L. Quan. “Asymmetrical occlusion handling using graph cut for multi-view stereo,” *CVPR05*, (2):902-909, 2005.

[12] Y. Wei, L. Quan. “Asymmetrical occlusion handling using graph cut for multi-view stereo,” *CVPR05*, (2):902-909, 2005.

[13] M.-A. Drouin, M. Trudeau, S. Roy. “Geo-consistency for wide multi-camera stereo,” CVPR05, (2):351-358, 2005.

[14] M.-A. Drouin, M. Trudeau, S. Roy. “Geo-consistency for wide multi-camera stereo,” CVPR05, (2):351-358, 2005.

[15] *****Proc. in IEEE Conference on* (CVPR*****),, *****.

[16] S.B. Kang, R. Szeliski, J. Chai, “Handling occlusions in dense multi-view stereo,” *Proc. in IEEE Conference on Computer Vision and Pattern Recognition* (CVPR 2001), pp. I-103-110, 2001.

Robust Shape from Focus via Markov Random Fields

Victor Gaganov*, Alexey Ignatenko**
*Keldysh Institute of Applied Mathematics RAS, Moscow, Russia
**Lomonosov Moscow State University, Moscow, Russia
{vgaganov, ignatenko}@graphics.cs.msu.ru

Abstract

In this paper we study a problem of 3D scene reconstruction from a set of differently focused images, also known as the ***shape from focus*** (SFF) problem. Existing shape from focus methods are known to produce unstable depth estimates in areas with poor texture and in presence of strong highlights. So in this work we focus on the robustness of 3D scene structure recovery. We formulate a shape from focus problem in a Bayesian framework using Markov Random Fields and present an SFF method that yields a globally optimal surface with enforced smoothness priors. Although shape from focus has been studied for quite a long time there is no widely accepted test set for evaluation of SFF algorithms. Therefore we present a test set composed of 27 image sets with hand-labeled ground truth. We quantitatively evaluate our method on this test set and present the comparison results. These results demonstrate that our method is robust to highlights and untextured regions and that it outperforms the state-of-the-art.

Keywords: *computer vision, 3D reconstruction, shape from focus, Markov Random Fields, MRF, energy minimization.*

1. INTRODUCTION

Recovering the 3D structure of the scene from images is one of key challenges in computer vision. There are many approaches to this problem, each exploiting different image cues. Shape from focus and shape from defocus (SFD) are two such approaches that exploit image focus to estimate 3D geometry of the scene. SFD approach [3,4] attempts to extract scene depth information measuring the relative blurriness of scene images, taken with different focus settings. SFF approach [1,2] scans the scene, taking a sequence of images with different lens focus settings. Instead of varying the lens settings one can gradually move the camera along the viewing direction, taking images. After that for each scene point a best-focused image is found. For a sharp feature on an image the position of this feature in 3D space can be determined, and the uncertainty of an estimate is, by definition, the depth of field.

Shape from defocus can obtain scene depth estimates given as few as two images of the scene. The reconstruction can be done fast and an SFD approach can even be applied to dynamic scenes. In the same time SFF approach needs a fairly large number of images to obtain reasonable reconstruction of the scene geometry and hence SFF is limited to static scenes. However there are still many applications that may benefit from a reliable SFF method. Robust and accurate SFF algorithm is especially desired in such fields as mineralogy [9] and industrial inspection of small-scale objects [16]. In these applications a perfectly static object is observed using an optical microscope and the goal is to obtain a 3D model of this object. Also, for a general static scene shape from focus has greater applicability then shape from defocus since it makes mild assumptions about the scene and the image formation process. While SFD methods have to explicitly model the camera blurring process, the only assumption made by the SFF algorithms is that some quantitative measure of blur is minimized at the position of best focus. For these reasons we have chosen to investigate and improve shape from focus algorithms.

Although there are many different SFF methods [1,2,5-9] almost all of them are essentially local. For each pixel of each image these methods calculate a sharpness measure that is used to determine the quality of image focus. To reduce negative effects introduced by camera noise the values of sharpness measure for each pixel are averaged over a local window centered in this pixel. This local window is called the evaluation window and the averaging process itself is called aggregation. After the aggregation for each pixel SFF methods search for an image that has the largest value of sharpness measure. Clearly, for objects that have textureless regions on their surface such a procedure will produce unstable results. Local SFF algorithms also fail to produce reliable depth estimates when imaging conditions are imperfect. In the event that some parts of the image are underexposed or, to the contrary, contain a strong highlight local SFF methods would fail to produce reasonable depth estimates. One way to handle this is to use aggregation with larger evaluation windows. However using large evaluation windows results in coarse reconstruction and fine details of the scene surface may be lost. Also, in general, size of textureless regions can be beyond the reasonable size of an evaluation window.

In this paper we present a new SFF method that is robust and can produce stable depth estimates even in textureless regions. Our method is based on a Markov Random Fields (MRFs) theory. MRFs have a long history of use in computer vision and have proven to be a useful tool for many problems, such as image segmentation [10] and stereo correspondence [11]. We cast a shape from focus problem in a Bayesian framework using MRFs. In contrast to existing approaches, this yields a global SFF method. In our method the 3D model of the scene is found as a minimum of MRF energy function. This energy function combines sharpness cues taken from images and smoothness priors on reconstructed surface shape. Enforcing smoothness priors allows our algorithm to produce stable results in poorly exposed, textureless and highlighted regions. To verify the stability and robustness of the proposed method we have conducted experiments on the real world data. We have collected a diverse test set for SFF methods evaluation. Our test set is composed of images of different mineralogical samples. For each set of images in our test set we have hand-labeled the ground truth. We have quantitatively evaluated our method on this test set and the results show, that our method outperforms existing SFF methods in terms of robustness.

The rest of the paper is organized as follows. In Section 2 we introduce some notation and overview the related work. We

present our new robust shape from focus method in Section 3. In Section 4 we describe out test set for SFF methods evaluation and present experimental results. Finally, we provide concluding remarks in Section 5.

2. RELATED WORK

2.1 Image formation geometry

We begin with the essential concepts of optics that the SFF algorithms are based upon. Figure 1(a) shows a simple model of the image formation process – the thin lens model. All the light rays that are radiated by the scene point P and are refracted by the lens converge in the point Q. The relationship between object distance *p*, image distance *q* and focal length *f* of the lens is given by the thin lens law:

$$\frac{1}{p}+\frac{1}{q}=\frac{1}{f} \quad (1)$$

If the point Q coincides with the camera sensor, then and image of scene point P would be perfectly focused on a photograph. However if for some scene point P' corresponding convergence point Q' is not on a sensor plane the energy received by the sensor from P would be distributed over a patch on a sensor plane, resulting in image blur (see Figure 1(b)). The blur radius increases with the distance between Q' and a sensor plane.

The depth of field of an optical system is a part of 3D space that can be seen in-focus on a photograph. Size of the depth of field is a limiting factor for all shape from focus algorithms. However in practice the size of the depth of field can be deceased by using long focal lengths and large aperture sizes. For an optical microscope the depth of field size can be as small as several micrometers.

2.2 Shape From Focus

Shape from focus algorithms extract depth information from a set of differently focused images. We will denote sets of images of the same size by capital letters with hat $\hat{I}=\{I_1,..,I_N\}$. An images set $\hat{I}$ can be obtained ether by gradually varying the focal settings of the camera, or by moving the camera along the viewing direction. SFF algorithms assume that for each image the internal parameters of an optical system and the camera position are known. Given this information for a sharp feature of an image its position in 3D space can be estimated. Only the depth of field of an optical system limits precision of such an estimate. Since all the images in the set $\hat{I}$ are taken with the same camera orientation the 3D model of the scene can be represented as a depth map. Size of the depth map $D=\{d(x,y)\}, d(x,y)\in[0,N-1]$ coincides with size of images in set $\hat{I}$.

Existing SFF algorithms do not operate directly on images from set $\hat{I}$. First, a sharpness measure function is computed for each image in the set, resulting in N sharpness images $\hat{S}=\{S_1,..,S_N\}$. Sharpness measure function should produce high response to image regions that contain high frequencies since focused image usually have more high frequencies then defocused images. The most popular sharpness measures for SFF algorithms are the Tenengrad operator [6] and the Modified Laplacian (ML) that was proposed by Nayar and Nakagawa [2]. In this work we have used ML operator that is a discrete version of the following function

$$ML(x,y)=\left|\frac{\partial^2 I(x,y)}{\partial x^2}\right|+\left|\frac{\partial^2 I(x,y)}{\partial y^2}\right| \quad (2)$$

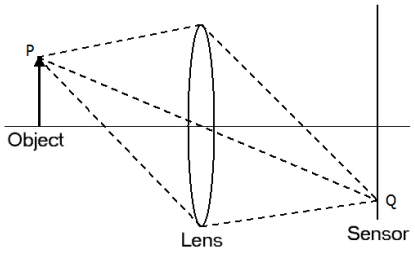
Raw values of sharpness measure are extremely sensitive to camera noise and for this reason existing SFF algorithms perform sharpness measure aggregation. In an SFF method proposed by Nayar [2] sharpness measure values are averaged over a square evaluation window centered in a target pixel

$$SML(x_0,y_0)=\sum_{(x,y)\in\Omega(x_0,y_0)} ML(x,y) \quad (3)$$

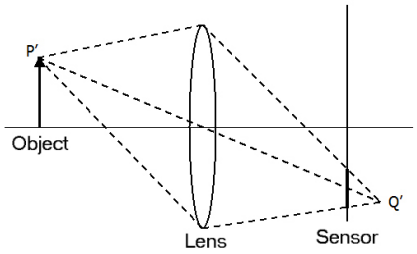
After averaging the resulting depth map is obtained by a winner-takes-all (WTA) strategy. For each pixel an image with a maximum value of averaged sharpness measure is selected

$$D(x,y)=\arg\max_i(SML_i(x,y)) \quad (4)$$

There are many different modifications of this basic scheme [1,7,8]. In [1] authors propose to refine depth estimates given by (4) finding a local plane with maximum average sharpness. This improves accuracy in areas where scene surface is not parallel to image plane.



(a)

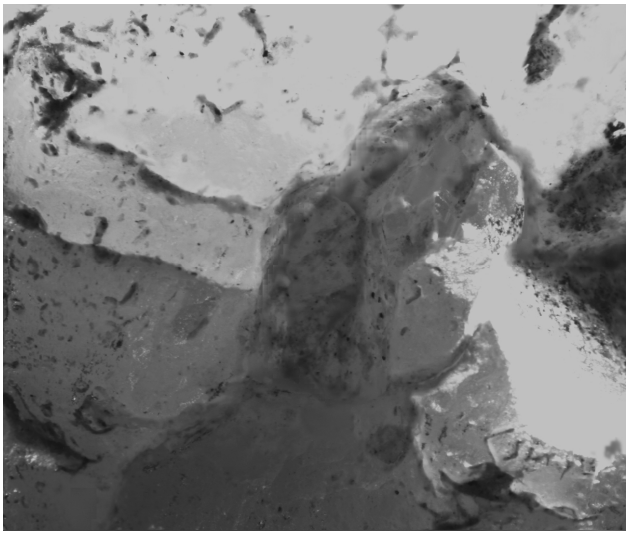


(b)

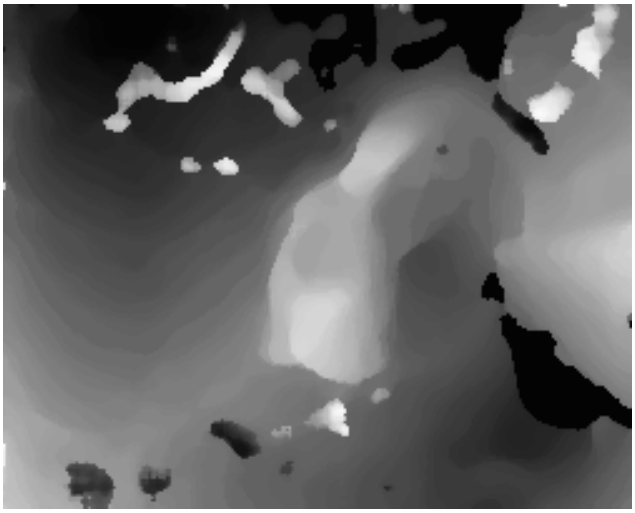
Figure 1. Formation of focused (a) and defocused (b) images

In [7] Ahmad and Choi propose to use dynamic programming to refine results, obtained by (4). This method is less computationally expensive then a brute force approach of [1]. Authors of [8] propose an adaptive averaging scheme that resembles bilateral filtering. This method is computationally efficient and allows obtaining more accurate estimates near sharp depth discontinuities. However this method utilizes an image of the scene taken with large depth of field. Large depth of field image cannot be obtained for strong magnifications of an optical microscope due to physical limitations.

The major drawback of all mentioned SFF algorithms is that they are essentially local. The WTA strategy that searches for a maximum value of sharpness measure fails to produce reasonable results in textureless and underexposed areas and in highlighted regions. In case a textureless region is not very large this can be handled by increasing the evaluation window size. However in general it is impossible to make an evaluation window bigger then the largest textureless region in the scene. Figure 2 shows a typical result of an SFF algorithm on a complex real-world example. As it can be seen the results are unstable in regions without texture. Surprisingly few works on SFF address this issue. Nair and Stewart [6] propose to reject depth estimates in areas where the sharpness measure is below a certain threshold. This results in a sparse but more reliable depth map. Authors of work [9] propose to apply median filter to the resulting depth map to suppress gross errors in reconstruction.



(a)



(b)

Figure 2. Results of a local SFF algorithm of a real world example. Depth map (b) is quite unstable in textureless regions. Multifocus image for this example can be seen in Figure 2(a)

However these simple countermeasures often fail in real-world applications, as it will be shown in our experiments section. To produce stable results in untextured, highlighted and underexposed regions an SFF algorithm should be based on global reasoning. Reliable SFF algorithm should take into account global constraints, along with local sharpness information.

3. PROPOSED METHOD

Our shape from focus method is based on Markov Random Fields theory. MRFs have proven to be an extremely useful tool for many computer vision problems such as stereo correspondence [11] segmentation [10] and image restoration [17]. In this section we show how an SFF problem can be formulated in MRF framework. In this framework the resulting depth map can be found as a maximum a posteriori (MAP) estimate of hidden state of an MRF. This yields an SFF algorithm based on global energy minimization that combines surface shape priors with observed sharpness information from images.

3.1 MRF Formulation

In SFF problem the goal is to estimate the hidden state $D=\{d(x,y)\}$ given the observed data $\hat{I}$. Like other SFF algorithms we would operate on sharpness measure, rater then on original images. Hence the Bayes rule gives us $P(D|\hat{S})=P(\hat{S}|D)P(D)/P(\hat{S})$. To reduce clutter in notation we will write a matrix $D=\{d(x,y)\}$ as a one-dimensional vector d and S_i as vector s_i. Thus a MAP estimate of d can be obtained as follows

$$d_{MAP}=\arg\min_{d}(-\ln(P(\hat{S}|d))-\ln(P(d))) \quad (5)$$

First term in (5) is a likelihood of observing the data given a certain hidden state d and the second term is a prior probability of d. We factorize the likelihood $P(\hat{S}|d)=\prod_i P(\bar{s}_i|d_i)$ where $\bar{s}_i=\{s_1(i),..,s_N(i)\}$. This corresponds to an assumption that values of sharpness measure in certain pixel are independent of depth values in all other pixels. We also assume that d is a Markov Random Field and we assume that only neighboring pixels are statistically dependant. In this case prior in (5) can be factorized using the Hammersley-Clifford theorem and minimum of (5) can be found as (see [12] for detailed derivation) minimum of

$$E(d)=\sum_i (V_i(d_i)+l\sum_{j\in N(i)} U_{\{i,j\}}(d_i,d_j)) \quad (6)$$

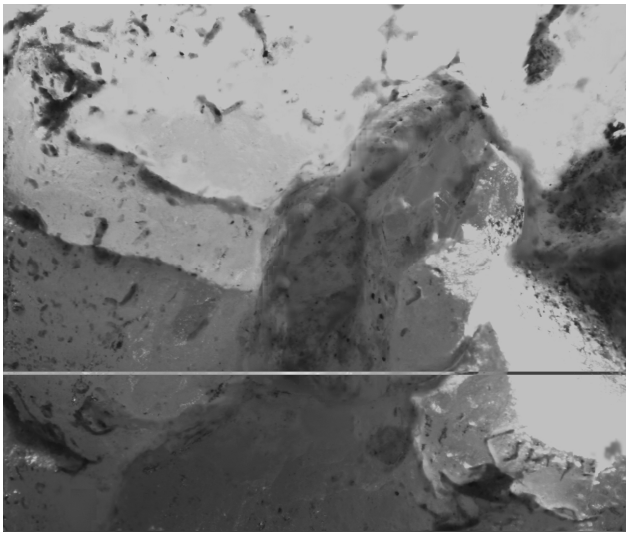
where $N(i)$ defines the 4-connected neighborhood of element i. Function $E(d)$ is called the MRF energy function. This function consists of sums of terms of two types. Terms of first type depend on single element of depth map and are called data terms. Terms of second type depend on two neighboring elements of depth map and are called the smoothness terms. Defining data and smoothness terms appropriately and optimizing energy function in (6) we will obtain d_{MAP} that is a MAP estimate of the scene depth map.

3.2 Data term

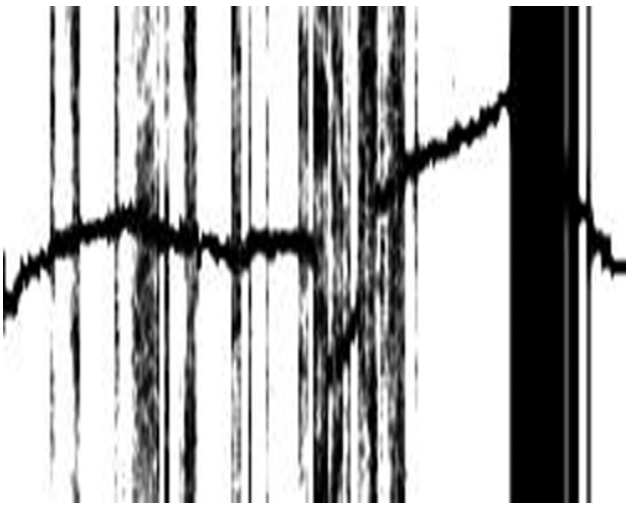
Data term in energy (6) encodes likelihood of observing particular $\bar{s}_i$ given different values of d_i. We propose to define the data term as follows

$$V_i(d_i) = \min((\max(\bar{s}_i) - s_i(d_i))^2, T_d) \quad (7)$$

The data term (6) penalties significant deviations from a maximum value of sharpness measure.



(a)



(b)

Figure 3. Illustration of our data term behavior. Figure 3(a) shows a multifocus image of the set and a position of data term slice. Figure 3(b) shows a slice of our data term. The X axis corresponds to the X axis of images in set, Y axis corresponds to a number of image in set

It can be seen that our data term is minimized by the depth value that corresponds to an image with maximum sharpness. Hence our data term gives preference to depth maps that pass through sharp areas of $\hat{S}$. In case scene area that corresponds to d_i contains no significant texture all the variations in sharpness measure profile $\bar{s}_i$ will be due to camera noise. Hence our data term will give no significant preference to any particular value of d_i. In such areas the surface will be primarily shaped by the smoothness term. However if there is strong texture sharpness profile maximum will be significant and the data term will give clear preference to depth values that correspond to sharp photographs. The behavior of our data term is illustrated in Figure 3. The data term is bounded by T_d to make the method less sensitive to outlier maximums in sharpness profile.

We would like to notice that unlike the existing SFF algorithms we omit the sharpness measure aggregation. This is motivated by the fact that (6) that is minimized by our method contains a sum of data terms for all elements of depth map d. Hence our algorithm implicitly averages the sharpness measure over the entire surface of the reconstructed scene. This choice is verified by our experiments.

3.3 Smoothness term

Smoothness term in (6) is responsible for regularization of the depth map estimate. In scene parts where sharpness measure doesn't provide reliable depth information smoothness prior shapes the depth map to make the results stable. We derive our smoothness term from truncated L2 norm

$$U_{\{i,j\}}(d_i, d_j) = \min((d_i - d_j)^2, T_s) \quad (8)$$

We prefer L2 norm rather then L1 norm or Generalized Potts model (GPM) [12] since it is a better prior for slanted surfaces. Truncation with T_s is necessary to preserve discontinuities. We propose the following smoothness term based on (8)

$$U_{\{i,j\}}(d_i, d_j) = \begin{cases} \min((d_i - d_j)^2, T_s), w_{ij} = 1 \\ (d_i - d_j)^2, w_{ij} = 0 \end{cases} \quad (9)$$

It can be seen that our smoothness term is spatially varying. Binary variable w_{ij} is an indicator of significant texture presence. In case significant texture is present around d_i we truncate the L2 norm with T_s thus allowing sharp jumps in depth map. Otherwise we use non-truncated L2 norm. As a result sharp jumps are not allowed in areas where significant texture isn't present.

We propose to calculate w_{ij} as follows. First we compose a maximum sharpness image

$$H(x, y) = \max(S_1(x, y), ..., S_N(x, y)) \quad (10)$$

After that we threshold $H(x, y)$ and obtain a binary mask of pixels with maximum sharpness greater then T_H. This mask is then processed with mathematical morphology. We apply *opening* with a circular structuring element of radius R to obtain the resulting significant texture presence image. After that we define w_{ij} to be equal to one in case there is significant texture in both pixels that correspond to d_i and d_j.

3.4 Energy optimization

Optimization of MRF energies is an area of active research and there are many different methods that can minimize energy (6). These methods include belief propagation, tree-reweighted message passing (TRW), α -expansion and α, b -swap [13]. In this work we have chosen to use α -expansion. This choice is motivated by the comparison of different MRF energy optimization methods presented in [13]. According to this comparison α -expansion is the fastest energy optimization algorithms algorithm and it produces solutions with very low energy. Although the smoothness term that we have proposed in section 3.3 is non-submodular [18] α -expansion still can be applied to optimize energy (6) as it was shown in [14].

4. EXPERIMENTS

To validate the proposed method and compare it with the existing methods we have implemented it in C++ and tested on a set of real-world example. Unfortunately there is no widely accepted test set for comparison of SFF algorithms. For this reason we have created a test set with data from our application. Inspired by the work of Scharstein and Szeliski on stereo correspondence algorithms evaluation [15] we have hand-labeled the ground truth for our data to evaluate algorithms quantitatively.

4.1 Test images for evaluation

For evaluation of our SFF algorithm we have composed a real-world test set with 27 different image sets. All images in our test set were taken with Leica Z16 APOA optical microscope equipped with a gray-scale digital camera. For our experiments we used 1280x1024 resolution. All images were taken at maximum magnification of the microscope with the depth of field ~30mkm. Each of our 27 test sets contains from 50 up to 200 images. Leica Z16 is motorized and sets of differently focused images have been obtained moving the optical system of the microscope along the viewing direction automatically. Step between two consecutive images for each test set was one-half of the depth of field. Our test set is composed of images of different minerals. Our major application was reconstruction of cavities on the surface of the minerals. Our data set is divers and includes image sets with a lot of texture along with image sets with many untextured regions (see Figure 4).

To evaluate our algorithm quantitatively we have hand-labeled the ground truth for all our 27 test sets. We have implemented a tool that allows browsing a set of differently focused images and placing control points in these images. Control points have to be placed in areas where the sharp features are present. After a number of control points have been placed a ground truth depth map can be obtained via interpolation. For textureless regions near the image border interpolations sometimes fails to produce reasonable depth estimates. Such regions have been hand-labeled and excluded from evaluation. To obtain a high quality depth map lots of control points a required and described procedure becomes very time-consuming. For this reason in the future we are planning to obtain ground truth using an X-ray scanner. Figure 4 demonstrates a number of ground truth depth maps labeled with our tool along with multifocus images of corresponding image sets.

4.2 Comparison results

To evaluate our algorithm we have implemented it in C++. We have also implemented an algorithm presented by Nayar with 10x10 evaluation window [2]. We have implemented this algorithm rather then it's modifications presented in [1,7] since all this modifications use results of Nayar algorithm as an initial estimate. This algorithms refine the initial estimate and if there is a gross error in the initial estimate the refinement result will also be erroneous. We could not implement methods presented in [6,8] for our data, since this algorithms rely on an image of the scene taken with large depth of field. For strong magnifications of a microscope it is impossible to obtain high quality image with a large depth of field due to physical limitations. However for our comparison we have implemented a modified algorithm of Nayar with rejection of unreliable depth estimates described in work on Nair and Stewart [6] followed by linear interpolation. The results of this algorithm have been post-processed with a median filter to suppress gross errors, as proposed by Niederoest et al in [9]. We denote this algorithm as 3N (Nayar, Nair, Niederoest).

To compare the algorithms quantitatively we have used error metric presented in [15]

$$Q(\hat{d},\tilde{d}) = \frac{\sum_{i=1}^{m} q(\hat{d}_i - \tilde{d}_i, T_e)}{m}, q(x,t) = \begin{cases} 1, |x| > t \\ 0, |x| \le t \end{cases} \quad (11)$$

where m is the number of depth map elements. This metric measures the share of depth map elements with estimation error greater then a specified threshold. In our experiments T_e was twice as large as the depth field. We have selected such a large T_e to detect only gross errors since our major concern was robustness of the algorithms. Parameters of 3N and our algorithm have been fine tuned to produce optimal results on our test set. We used the same values of parameters for all image sets in our test base. Results of quantitative evaluation are presented in Table 1.

Method name	Significant errors share (Q)
Nayar	15.7%
3N	6.4%
Proposed method	3.6%

Table 1. Quantitative comparison of proposed method with existing methods

It is clear that in term of error metric (11) our algorithm produces the best results. For visual comparison in Figure 4 we present the results for 3 different image sets from our test base. It can be seen that basic algorithm of Nayar is extremely unstable in textureless regions. This perception is confirmed by our quantitative evaluation. According to it as much as 15% of depth estimate given by method of Nayar are unreliable. Results of 3N algorithm are much more stable then the results of Nayar algorithm. For the second images set (Figure 4(k)) 3N algorithm produced near perfect result. It's worse noticing that the best results for 3N algorithm were obtained with the median filter radius of approximately 50 pixels. This suggests that an algorithm tends to incorporate as much context information as possible to produce stable results. This verifies our idea that a stable SFF algorithm

should be based on global reasoning. However for large untextured regions the 3N algorithm fails to produce reliable depth estimates (Figure 4(j,l)). In contrast, the proposed method handles textureless regions of arbitrary size correctly. It can be seen that the results of our algorithm are smooth and consistent although we omit the sharpness measure aggregation. Optimal results for our algorithm have been obtained with $T_s = 400$ and the resulting depth maps tend to be smooth almost everywhere. Sometimes this results in oversmoothing, as it can be seen from Figure 4(m). We are planning to address this issue in our future work. However on the average our algorithm produces the most reliable result among the compared methods.

5. CONCLUSION AND FUTURE WORK

In this work we have presented a new shape from focus algorithm that is based on Markov Random Fields theory. In contrast to existing SFF methods our algorithm yields globally optimal depth maps with enforced smoothness constraints. Presented experiments verify that our algorithm is able to produce stable depth estimates in untextured, underexposed and highlighted regions, where existing SFF methods fail to produce reliable results.

To evaluate our algorithm we have used the same values of all parameters for all image sets in our test base. However we have noticed that for some image sets this results in suboptimal depth estimates. So in the future we are planning to estimate algorithm parameters, along with the depth map in the unified framework. As it was noticed in the experiments section our algorithm sometimes produces results that are oversmoothed. To handle this realistic priors on sharp discontinuities location should be incorporated in the algorithm. This will allow our method to produce globally optimal depth maps without oversmoothing.

6. REFERENCES

[1] Subbarao M., Choi T. *Accurate recovery of three-dimensional shape from image focus*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995, v. 17, n. 3, pp. 266–274.

[2] Nayar S. K., Y. Nakagawa. *Shape from focus*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1994, v.16, n.8, pp. 824-831.

[3] Jin H., Favaro P. *A variational approach to shape from defocus*, Proc. IEEE European Conference on Computer Vision, 2002, pp. 18-30.

[4] Subbarao M., Surya G. *Depth from defocus by changing camera aperture: A spatial domain approach*, Proc. Computer Vision and Pattern Recognition, 1993, pp. 61–67.

[5] Grossmann P. *Depth from focus*, Pattern Recognition Letters, 1987, v.5 n.1, p.63-69

[6] Nair H.N., and Stewart C.V. *Robust focus ranging*, Proc. Computer Vision and Pattern Recognition, 1992, pp. 309–314.

[7] Ahmad, M.B., Tae-Sun Choi. *A heuristic approach for finding best focused shape*, IEEE Transactions on Circuits and Systems for Video Technology, 2005, v.15, n.4, pp. 566–574.

[8] Aydin T., Akgul Y.S., *A New Adaptive Focus Measure for Shape From Focus* Proc. BMVC 2008

[9] Niederoest M., Niederoest J., and Scucky J. *Automatic 3D Reconstruction and Visualization of Microscopic Objects from a Monoscopic Multifocus Image Sequence*. International Workshop on Visualization and Animation of Reality based 3D Models, 2002

[10] Boykov Y., Jolly M.P. *Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images*. Proc. International Conference On Computer Vision, 2001, pp. 105-112

[11] Boykov Y., Veksler O., Zabih R. *Fast Approximate Energy Minimization via Graph Cuts*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, v.23, n.11, pp. 1222-1239

[12] Boykov Y., Veksler O., Zabih R. *Markov Random Fields with Efficient Approximations.* Proc. Computer Vision and Pattern Recognition, 1998, pp. 648-655

[13] Szeliski R., *et al. A comparative study of energy minimization methods for markov random fields*, In Proc. European Conference on Computer Vision, 2006, v.20, n.6, pp. 1068-1080

[14] Rother, C., Kumar, S., Kolmogorov, V., Blake, A. *Digital tapestry*, Proc. Computer Vision and Pattern Recognition, 2005, pp. 589-596

[15] Scharstein D., Szeliski R. *A taxonomy and evaluation of dense two-frame stereo correspondence algorithms*, International Journal of Computer Vision, 2002, v. 47, pp. 7-42.

[16] Schaper D. *Automated quality control for micro-technology components using a depth from focus approach*, Image Analysis and Interpretation, 2002, pp. 50-54

[17] Geman S., Geman D. *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1984, v.6 pp. 721-741.

[18] Kolmogorov V., Zabih R. *What Energy Functions Can Be Minimized via Graph Cuts?* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, v. 26, pp. 147-159

About the authors

Victor Gaganov is a PhD student at Keldysh Institute of Applied Mathematics of Russian Academy of Science. His research interests include computer vision, 3D reconstruction, machine learning and adjacent fields. His contact e-mail is vgaganov@graphics.cs.msu.ru.

Alexey Ignatenko is a researcher at Computational Mathematics and Cybernetics department of Moscow State University. His research interests include photorealistic 3D rendering, 3D modeling and reconstruction, image-based rendering and adjacent fields. His contact e-mail is ignatenko@graphics.cs.msu.ru.

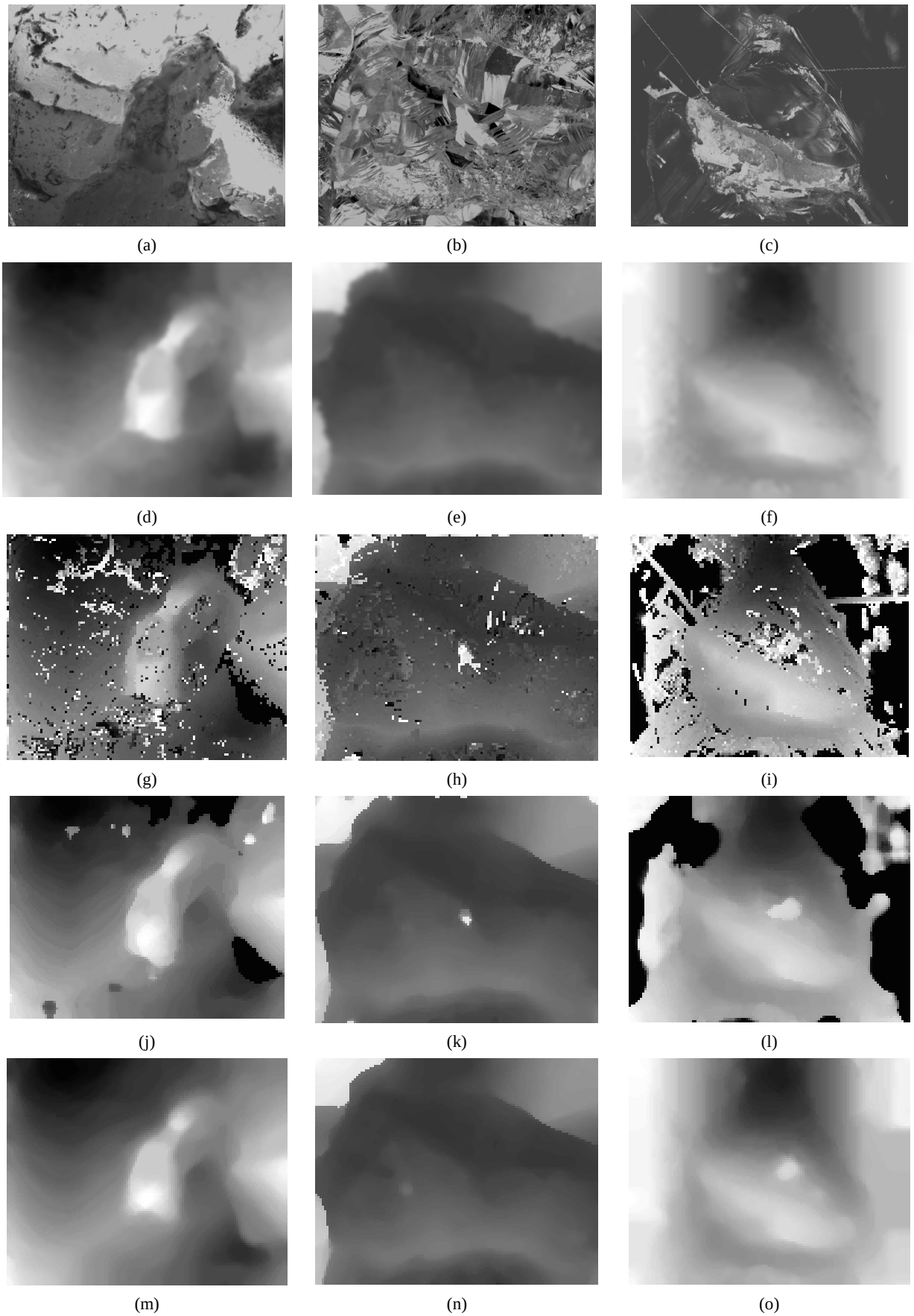


Figure 4. Comparison results for three examples from our test set. First row contains multifocus images for these three sets. Second row is ground truth. The rest three rows are the results of Nayar algorithm, 3N algorithm and the proposed method respectively

Proximity Visualization via Common Fate Luminance Changes

Lior Wolf Chen Goldberg Yehezkel Yeshurun
The Blavatnik School of Computer Science
Tel Aviv University

Multi-Dimensional Scaling

Two frames from the proposed visualization

Figure 1: A comparison between two visualizations of the NASDAQ-100 list. Proximities between stocks were computed based on their 2007 daily prices. (left) The conventional 2D embedding visualization, modified slightly to eliminate occlusions. One panel is the entire space and the other is a zoomed in view. (right) two frames out of the proposed animated visualization. The proposed design depicts several prominent connections that are not depicted by the conventional visualization. For example, SanDisk and Intel's prices were correlated in 2007, as were the prices of Intel and Oracle. However, the prices of Oracle and SanDisk were much less correlated. This can be easily observed in the proposed design, but not in the conventional visualization. The interactive animation is available in the accompanying supplementary material `http://www.cs.tau.ac.il/~wolf/demos/commonfate.zip`.

Abstract

We introduce a new design for visualizing high dimensional data. Points in high dimensional space are often illustrated through 2D or 3D embeddings, which are often cluttered and may-be unaccessible to non-scientific audience. In our design, instead of representing data points as 2D/3D vectors, each data point is represented by a smoothly varying function of time. These smooth functions are used to control the luminance of iconic representations of the data points, where the value of each such function determines the luminance of one icon that is associated with the underlying data point.

The resulting display has a high capacity and is intuitively clear to the viewer. Perceptually, it is based on a fundamental psychophysical principle, namely the Gestalt law of Common Fate. Algorithmically, we present two methods designed to meet the demands of the visualization: a new non-negative matrix factorization method, and a partition-based grid embedding technique.

1 Introduction

Perhaps the earliest record of a dynamic screen-like display is the one of the biblical Chosen (Exodus 28) that was a sacred garment in which 12 precious stones were set in a grid. The Hebrew high priest would communicate with the divine by interpreting the patterns of glowing gems. We propose a visualization design of a similar structure. Icons are placed on a grid, and their individual luminance values change over time. In the proposed design, the patterns of luminance-change convey proximities between the objects represented by the icons. In particular, icons that are lit and unlit together in a synchronized fashion are perceived as related. Thus, correlative information is conveyed to the user intuitively and effortlessly.

Problem formulation In this work we visualize proximity information between tens or hundreds of data points. The proximity can arise from Euclidean distances in the case of vectorial data, from the analysis of a given graph structure, or based on other means of measuring correlations. In all of these cases, the input to our method is the same – the number of data points n, iconic representations of the n points, and a symmetric $n \times n$ affinity matrix A containing positive values. A high affinity value between points i and j (a large $A(i,j)$ entry) indicates that data points i and j are similar. Low values indicate distant, dissimilar points.

In this work we concentrate on visualizations for the non-professional audience, and we adhere to several design requirements. First the data points are required to be effortlessly identifiable. Therefore, we are interested in visualizations that display icons (e.g., text, logos or thumbnails) and not glyphs (e.g., dots), as in most scatter-plots. Second, we wish for the display to be intuitively clear to the user, with no need for elaborate user training. This limits us to visual encoding which is cognitively natural. Third, we wish to have a clean uncluttered display that is visually attractive.

2 Related work

Many times the proximity we wish to visualize is estimated based on high-dimensional data. The visualization of such data is often treated as a Dimensionality-Reduction problem, where each data point i is represented by a vector $x_i \in \Re^m$, for an arbitrary m. The vectors x_i, $1 \leq i \leq n$ are chosen such that they are correlated, up to some error that the method minimizes, according to the pairwise correlations in A. One such method is multidimensional scaling (MDS) [Shepard 1962], which has been used extensively for visualization purposes (e.g., the ThemeScape system [Wise et al. 1995]). In such applications, one recovers $m = 2$ or $m = 3$ dimensions, which are used to determine the screen-location of each data glyph or icon.

Other Dimensionality Reduction methods have also been adapted for visualization purposes. Recently, Venna and Samuel [Venna and Kaski 2007] have compared classical MDS (closely related to PCA), ISOMAP [Balasubramanian et al. 2002],

Curvilinear Component Analysis [Pierre Demartines and Jeanny Herault 1997], Locally Linear Embedding [Roweis and Saul 2000] and Laplacian Eigenmaps [Belkin and Niyogi 2001]. The results show that all embedding methods face difficulties when reducing the dimensionality to as low as two dimensions, and that Curvilinear Component Analysis (CCA) seems to behave better than the other methods. CCA is a weighted version of MDS, which takes into account only distances that are smaller than a threshold.

The capacity of Dimensionality Reduction based scatter plots is limited by the low-dimensionality. Potentially, more dimensions can be added by color-coding or other means, but such combinations seem to be cognitively taxing for the viewers [Treisman and Gormican 1988]. Also, if we choose to display icons and not dots, the icons would many times overlap, creating a cluttered inapprehensive and unattractive visualization. This can be partly avoided at the expense of accuracy by adjusting the locations of the icons.

An alternative methods for the display of multi-dimensional data is the Parallel Coordinates method [Inselberg and Dimsdale 1990]. In this method each coordinate is typically associated with one vertical line, and each data point is drawn as a polyline with vertices on the parallel axes that are positioned according to the coordinates of the point. In Parallel coordinates the variables (less than 10 is most examples) are identifiable and the individual data points are typically not identifiable without interaction. In contrast, our method displays the affinity between the data points and does not assume the existence of an underlying vector representation; Individual data points are represented by large identifiable icons. Parallel coordinates tend to become cluttered when displaying large datasets, and the power of the method is mostly in understanding global patterns. The method has been extended to reduce clutter [Ellis and Dix 2006; Fua et al. 1999], and to create a focus+context display [Novotny and Hauser 2006].

Another successful Focus+context method for displaying large tabular datasets is the Table Lens technique [Rao and Card 1994]. In this method the sizes of the rows and columns of a tabular display change dynamically to allow focus on individual data points (rows) and variables (columns). The Table Lens method is able to allow the exploration of thousands of data points and tens of variables. The problem it solves, however, is quite different from ours. We focus on the display of affinities that might be handed directly, arise from graph structures, from very sparse matrices, or from continuous time data. In all of these cases, a tabular representation may be inappropriate.

An alternative method for displaying affinity information directly (and not tabular information), is the Generalized Association Plots (GAP) method. In this method the correlation matrix of the data is iteratively updated to reflect the correlations between its columns. This iterative process produces correlation matrices of gradually decreasing ranks that can be used to generate a hierarchical clustering of the data points. In addition, it produces a series of elliptical 2D embeddings. Each such 2D embedding induces a 1D embedding of the data points (ordering of the rows and columns of the correlation matrix). This form of visualization can handle extremely large datasets, and does not focus on identifying individual data points.

As a dynamic technique, GAP can be animated and each data point can move along a path governed by its location in the series of elliptical embeddings. Since the correlation matrix is iteratively simplified, the clustering of the data becomes more and more clear as the process progresses. On the other hand, information regarding proximity to points in other clusters, that is not depicted in the 2D embedding of the first iteration, will probably not emerge. In this limited sense the GAP method suffers from the capacity problem of other 2D methods.

Use of animation for visualization Animation is extremely suitable for expressing causality. The work of Michotte [Michotte 1963] shows that with appropriate timing of events, a causal relationship is perceived. It can also express communication by animating messages moving from message sources to message destination objects [Stasko 1990]. Most naturally, animation can be used to visualize dynamic and serial processes. The classical movie "Sorting Out Sorting" uses animation to illustrate a number of computer sorting algorithms [Baecker 1998].

Recently, Heer and Robertson [Heer and Robertson 2007] have studied the use of animated transitions between statistical graphs. The transitions were carefully designed to match the shift between the various aspects of the displayed data, and were shown to increase its apprehension.

More related to our work, Limoges et al. [Limoges et al. 1989] evaluated the use of motion in visualizing properties of data points. They enhanced a conventional scatter plot representation by allowing the points to oscillate sinusoidally, either horizontally or vertically about a center point. The type and amount of motion (frequency, phase and amplitude) were used as visualization properties. The results show that data mapped to phase is perceived as effectively as more conventional properties, such as point size or gray value.

Ware and Bobrow [Ware and Bobrow 2005] show that motion based highlighting is effective for visualizing local neighborhoods in graphs that contain many nodes. Moreover, they showed that motion highlighting and static highlighting can be combined to allow efficient visualization of set intersections. Can an entire graph be visualized by highlighting the neighborhood of each node sequentially? In Section 5 we report from our experience that such a visualization is hard to apprehend.

Using synchronized motion patterns is one of the many manifestations of the well known Gestalt principle of Common Fate. According to this principle, stimulus elements are likely to be perceived as a unit if they move together. However, there is some psychophysical evidence substantiating the view that luminance-based common fate is also very prominent [Sekuler and Bennett 2001]. In our work, the position of points does not change in order to communicate correlations. Instead, information is encoded by synchronization of light patterns, which is related to the concept of phase. Note, however, that the capacity of our method is much higher since the patterns we display evolve over time.

3 The proposed design

In our visualization a fixed grid of icons changes over time by associating a dynamic luminance value to each individual icon. There are two computational questions that need addressing. The first is how to determine the luminance of each icon as a function of time. The second question is how to distribute the icons in space.

The two questions are related. In the datasets that we consider there are more data points than the number of icons that the display window can accommodate at once. This requires us to pan the display over the grid as the animation progresses. Therefore, ideally, all icons that are lit at once are contained in a region of the grid which is small enough to display.

3.1 Embedding in time

In our design, every data point $i = 1..n$ is represented by a positive function $f_i(t)$ that captures its luminance over time. The correlations between these functions should adhere to the given affinity matrix A, which suggests that we minimize a discrepancy score such as

$$\sum_{i,j}(A(i,j) - f_i \cdot f_j)^2$$

where

$$f \cdot g := \int_t f(t)g(t)dt$$

In addition, to create a smooth animation, the luminance should change gradually. Therefore, we also minimize

$$\sum_i ||\partial f_i/\partial t||^2 := \sum_i \int_t (\partial f_i/\partial t)^2(t)dt$$

In the optimization process, we discretize the functions f_i and use vectors $v_i \in \Re^m$ instead. The continuous derivative operator becomes a matrix operator, given by the $m \times m$ matrix H that has a main diagonal of -1, all values of the adjacent upper diagonal equal 1, and the rest of the elements are zero. Furthermore, in order to play the animation in a loop, we modify the first element in the last row of H to be 1. This alteration connects the last frame to the first.

Let V be the matrix whose columns are v_1,v_2,...,v_n. Our optimization procedure minimizes the following energy function for some given parameter λ

$$E = ||A - V^\top V||_F^2 + \lambda ||HV||_F^2$$

Replacing the Frobenius norms with the equivalent traces and eliminating constants, we obtain:

$$E = tr(V^\top V V^\top V) - 2tr(V^\top V A) + \lambda tr(V^\top H^\top H V)$$

Differentiating:

$$\frac{\partial E}{\partial V} = 4VV^\top V - 4VA + 2\lambda H^\top HV$$

We optimize in an iterative manner until convergence, using the gradient projection method (e.g., [Nocedal and Wright 1999]). We initialize a random matrix V^0 and update it by the following rule:

$$V^{s+1} = \max(V^s - \eta \frac{\partial E}{\partial V}, 0) \quad ,$$

where η is the learning rate, and the maximization is performed element by element.

Note that without the smoothness term, our computational problem would have been similar to the problem of (Symmetric) Nonnegative Matrix Factorization (NMF) [Lee and Seung 2001; Li et al. 2007] . NMF solutions are known to be sparse. In our case, this means that most icons would be unlit most of the time, which is desirable since it allows the viewer to focus on a handful of lit icons at every frame.

3.2 Motion and embedding in space

The proposed visualization-technique is designed for hundreds of data-points. However, at any given time only 25-100 can be displayed, depending on the screen size. We solve this by panning a display window such that it is centered at each frame on the center of mass of all lit icons. In other words, let x_i, y_i ($i = 1..n$) be the grid locations of the icons associated with the n data points. The center of the window in time t is placed in coordinates $\bar{x}(t)$, and $\bar{y}(t)$, where

$$\bar{x}(t) = \frac{1}{\sum_{i=1}^n v_i(t)} \sum_{i=1}^n v_i(t) x_i$$

and similarly for $\bar{y}(t)$.

As mentioned above, the main requirement for the spatial grid embedding is that icons that are lit together would fit together in a region small enough to display at once. The computational problem is as follows: given a threshold θ, embed a set of vectors v_i, $i = 1..n$ in a 2D grid, such that for every pair i and j for which there exists a frame index t such that both $v_i(t) \geq \theta$ and $v_j(t) \geq \theta$, the maximal horizontal and vertical distances between the embedding of i and j is not larger than a specified amount.

We are not guaranteed that such an embedding exists. Furthermore, finding such an embedding is an NP-complete problem (by reduction to the Maximal Common Subgraph problem, details omitted). In a sense, this grid embedding challenge is a consequence of the unsuitability of 2D embeddings for the display of high-dimensional data. However, the constraints here are more relaxed since nearby pairs of points that have low affinities are still separated by the luminance modulation.

To heuristically solve the embedding problem we employ the following method, which is based on a quadtree partitioning of the space [Samet 1984]. First, we compute an $n \times n$ co-occurrence matrix O, in which O(i,j) is one if there is a frame index t such that both $v_i(t)$ and $v_j(t)$ are larger than the threshold $\theta = 0.1$, zero otherwise. Next, we embed all n points in $2D$ by applying MDS to the matrix O and obtaining coordinates $(\tilde{x}_i, \tilde{y}_i)$, $i = 1..n$. We then partition the grid horizontally to two sub-grids, followed by a vertical partitions of each sub-grid. At each partitioning stage, we divide the points between the grid partitions. The splitting process repeats recursively until all the partitions contain a single point.

At each stage of the recursion, there is a rectangular $s_x \times s_y$ subgrid in which we need to embed $s_x s_y$ icons. Assume that the next partitioning operation is along the horizontal axis, and that s_x is an even number. We sort all points that are assigned to the sub-grid by their $\tilde{x}_i$ value, and assign the $s_x s_y/2$ points with the lowest $\tilde{x}_i$ values to the left $s_x/2 \times s_y$ partition. The rest are assigned to the right partition of the same size. If s_x is an odd number, we compute the minimum, median, and maximal values of the x-axis embedding $\tilde{x}_i$ of all points that are associated with the grid. We then check if this median is to the left or to the right of the middle of the section between the min value and the max value. The middle grid column is assigned to the left or right half-grid accordingly.

3.3 Interactivity

The proposed design enables one to select data-points by clicking on the corresponding icons. Once there are selected points, the only frames that are shown are frames t for which for every selected datapoint i, the value of $v_i(t)$ is larger than a threshold $\theta = 0.1$.

Some data points, which do not share strong links with the other points, may not be lit in all of the computed frames. Similarly, in the multiple selection case, the set of frames that are lit for all selected data-points may be empty. In such cases the selection frame turns from green to red and the selected icons are all lit up. In the case of an empty multiple selection, a text message appears to explain the situation.

Once a point is selected, a "deselect all" icon appears. Mouse-clicking on this icon returns the display to the normal display mode starting from the last frame that is shown in the selection mode. Note that other forms of selection are also possible based on the application. For example, one might wish to display all frames for which there is at least one icon lit out of the selected icons.

3.4 Limitations

The information-capacity of our visualization is not limited to 2D or 3D, but instead is limited by the length of the animation sequence and by its speed (how fast the luminance of the icons change). Long sequences convey more information, as do faster changing ones. Put differently, their effective dimension as an embedding space is higher. The utilization of the higher dimensionality is bounded, of course, by the memory and attentiveness of the viewer.

Our visualization method is limited in that when no user interaction is present, the user takes a passive observing role following a predetermined animation. In contrast, a 2D-projection type visualization may allow the viewers to freely foveate (interactive zooming may be required for large datasets). Depending on the application, the user may prefer to search around for the desired information, or to *effortlessly* observe an animation sequence.

Parameter		value
m	Target dimension (Sec. 3.1)	n
	Video length (in frames)	$30m$
λ	Smoothness weight (Sec. 3.1)	3
η	Learning rate (Sec. 3.1)	0.01
ε	Convergence Threshold (Sec. 3.1)	10^{-5}
	Max number of iterations (Sec. 3.1)	1000
θ	Visibility threshold (Sec. 3.2)	0.1

Table 1: The set of fixed parameters used throughout the reported experiments and in the online facebook application.

Tversky et al. [Tversky et al. 2002] pointed to possible pitfalls of certain kinds of animations, analyzing animated visualization based on *congruence* between internal and external representations and on the *apprehensibility* of the resulting animation. In congruent visualizations the format and the contents of the graphics match those of the displayed concept; Apprehensive visualizations are those in which the graphical information is perceived appropriately and accurately enough.

We believe that although we use animation not to depict time based dynamics, our animation is congruent since what appears in our mind together is thought of as linked. As for apprehensibility, this property lacks mostly in high paced animations, where transitions are rapid. The smoothness in time of our underlying representation eliminates some of this danger. While the user cannot be expected to extract all the information that exists in the visualization, the capacity of the proposed visualization is much larger than the one of 2D embedding even after considering loses arising from partial apprehension.

4 Results

We implemented our visualization design using the ActionScript programming language, which is used to create Adobe Flash animation. The computational part is performed in Matlab and saved as XML files. This separation between computation and visualization is natural for our design and implies its suitability in the client-server setting. In fact, the facebook visualization below is already implemented as a "facebook application" that was shown on actual facebook user-pages.

We use the design to visualize four different datasets, described below. In each dataset, we embed the data in an m dimensional space of smooth vectors as described in Section 3.1, where m is set to be the number of data points. In the actual visualization we interpolate this data to $30m$ frames by employing linear interpolation between every two consecutive coordinates. We then display the resulting animation at 30 frames per second.

The parameters used to visualize all four datasets are summarized in Table 1. These parameters seem to be appropriate for a wide range of dataset sizes. For example, we have had users for our facebook visualization with as few as 5 friends and users with as many as 800.

For the graph based data (facebook) we use the adjacency graph as the affinity matrix. In the case of vectorial data (all other datasets), the data points are normalized to have a norm of one, and the proximity is computed by considering the dot product between every two data points. Many other alternatives exist in the *kernel methods* literature (e.g., [von Luxburg 2007]). For example a Guassian kernel can be employed which computes affinities as $A(i,j) = \exp(-||x_i - x_j||^2/2\sigma^2)$, and the Matrix Laplacian can be used instead of the original affinity matrix. Our choice to employ the linear dot-product kernel was primarily motivated by our wish to minimize the number of parameters we need to tune during the data representation stage, thus validating the robustness of our method.

Netflix DVD Ratings A dataset containing DVD ratings by over 480,000 users was downloaded from the Netflix Prize website (`www.netflixprize.com`). We selected the subset of the 512 DVDs that were released in 2005, which is the last year for which DVD ratings are included in the dataset. Similarities were computed based on the user-specified ratings only, i.e., from long vectors where each user corresponds to one coordinate. Reviews are specified on a scale of 1 to 5. A rating of 0 is used to indicated no review for a specific DVD by a specific viewer.

The resulting visualization is depicted in Figure 2 and is presented in full (as do the other visualizations) in the accompanying supplementary material available at `http://www.cs.tau.ac.il/~wolf/demos/commonfate.zip`. As can be seen, it clearly displays proximities between related DVDs, and by watching and interacting with the visualization the perception of genres and subgenres emerges. In contrast, MDS, due to its low capacity, places very distinct genres nearby. As a result, one cannot judge reliably whether two DVDs are similar by observing their spatial proximity.

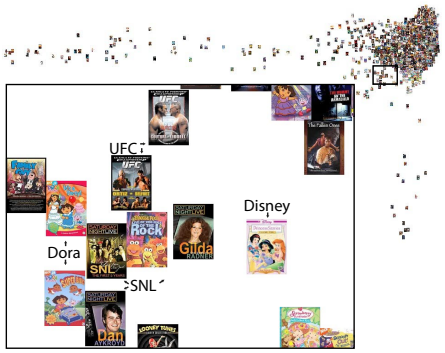
Movie Stars We selected 50 actors and actresses and represented each one by a binary vector which depicts the movies he/she participated in according to The Internet Movie Database (IMDb). Each movie was represented by one coordinate in this representation. The dot product (sans-normalization) between every two such vectors counts the number of movies both actors participated in together. The result, captured in Figure 3, provides a clear view of which actors and actresses played side by side.

While in the MDS-based plots actors that are well separated spatially may have played in the same movie and vise versa, the high capacity of our visualization depicts the relations between the actors reliably.

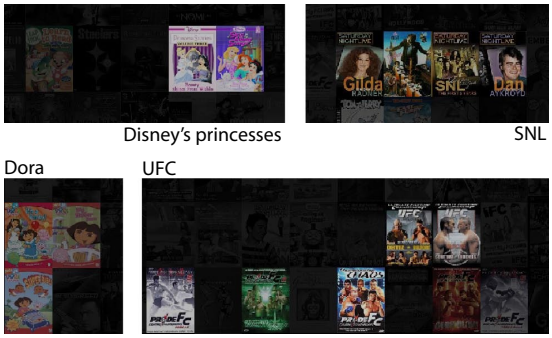
NASDAQ The stocks in the January 2008 NASDAQ-100 list were selected. Each stock was represented by a piece-wise normalized vector of its price changes during 2007 as described in [Gavrilov et al. 2000]. Note that the resulting linear kernel has negative values. We replaced these values by zeros. A number of frames are shown in Figure 1. The visualizaton reveals expected relations, such as the ones between Google and Yahoo, and Sandisk and Intel, as well as some less expected correlations such as the ones between the stocks of Apple and Google, and Autodesk and Starbucks. These correlations exist in the original data and arise from similar price patterns throughout 2007.

Facebook We have created a facebook application that allows users of the social networking website to visualize the connections between their friends. The service was tested by Computer Science students, their friends and a few random facebook users who stumbled across it, a total of 100 users. The service is currently down for the duration of the review period. The loading time of the application is dominated by the retrieval of the photographs of the facebook contacts, which is done in parallel to the computation of the smooth functions and the spatial embedding. Once the computation is done, the remaining images are loaded in the order in which they appear on screen, to allow for the quickest startup possible. The analysis takes a few seconds for users having 50 friends, and less than 2 minutes for users with $500-800$ friends. Due to the dictated structure of facebook applications, this computation is not done beforehand. However, in other (non-facebook) applications this may be possible. Also note that since solutions for nearby networks are expected to be similar, caching the results of previous sessions of the same user can reduce the computation time considerably assuming that the network of friends did not change drastically (this is not implemented yet).

In the facebook application, all the contacts of the user are rep-

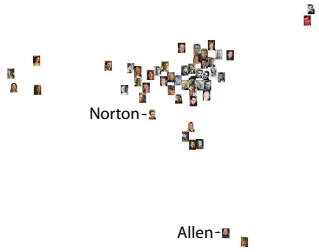


Multi-Dimensional Scaling

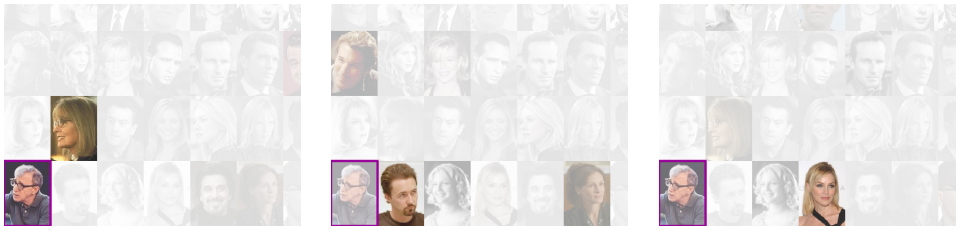


Crops out of the proposed visualization

Figure 2: A comparison between two visualizations of the netflix challenge 2005 DVDs. Proximities are based on the individual users' ratings only. The 2D embedding puts Disney's "Princess Stories", "Ultimate Fighting Championship", "Dora the explorer", and "Saturday Night Live" nearby. The capacity of our visualization (only crops are shown), enables full separation.



Multi-Dimensional Scaling



The proposed visualization

Figure 3: A dataset containing 50 actors and actresses, each associated with the list of movies he played in. Did Edward Norton and Woody Allen play in the same movie? Norton is in the middle of the MDS display. Allen is at the very bottom near Diane Keaton. The displayed animation frames were sampled when Allen's icon was selected.

resented as graph vertices. If person i and person j appear in the contact list of each other we create an edge between graph nodes i and j. As mentioned above, in this application, the resulting binary adjacency matrix is used as the affinity matrix. The resulting visualization for one person with 160 friends is a part of the attached flash-animation, and is captured in Figure 4.

We have received a considerable amount of feedback from the application users. It became evident that without preparation the purpose of the visualization is not clear to most viewers. We had to update the welcome splash screen and make it more appealing by adding an explanatory toy animation based on well known characters from "The Simpsons" television show (please refer to the supplementary material available at `http://www.cs.tau.ac.il/~wolf/demos/commonfate.zip`). The animation was accompanied with one line of text: "Cliques (groups of friends) appear lit together". All users we asked agreed that the structure of cliques among their friends is depicted correctly.

In the initial implementation, the selection could have led to a screen where all icons are unlit. Several users asked why some friends are never lit up. Similarly, two users reported dissatisfaction with the all unlit screen one would get in the initial implementation when selecting multiple people that do not share lit-up frames. To answer these concerns we added a better treatment of isolated graph nodes and the empty intersection problem as explained in Section 3.3 above.

Some users complained that the visualization is too slow, and some that it is slightly too fast (the facebook application did not contain the speed control at first). One user expressed the wish to have the ability to scroll by himself to other parts of the visualization. This feedback was made before the time slider and the data point selection box were added to the application.

4.1 Evaluation of the spatial embedding method

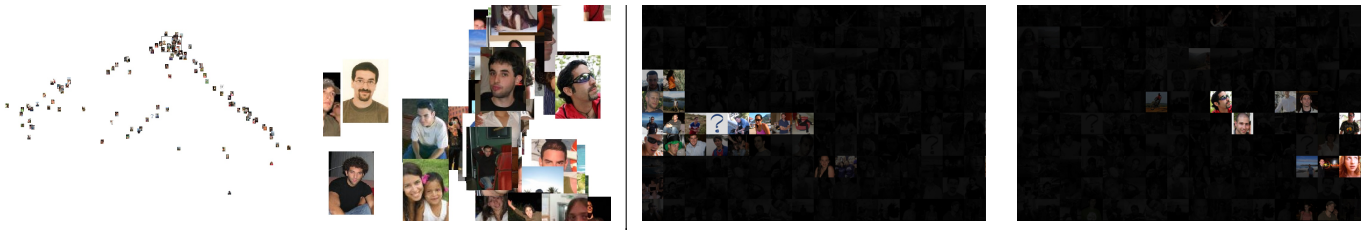
In the four visualizations described above there are two display modes: a windowed display mode of 800×600 pixels and a full screen one (1280×1024 pixels). The sizes of the icons range from 100×40 to 100×140, allowing visible grids of various sizes in each mode. Table 2 depicts the percentage of displayed lit items out of the total number of lit items. For example, in the NASDAQ visualization, there are $3,000$ frames, and a total of $41,161$ icons (counted again each frame) that are lit above a threshold of 0.1. Out of these lit events $39,341$ are displayed on screen during the visualization loop, producing a hit rate of 96 percents. In the full screen mode of the same visualization all icons appear on screen, and the hit rate is 100%. As can be seen from the table, the hit rate is quite high.

5 Alternative Visualizations

Our framework of transforming proximities into smooth functions and displaying those through animation can be extended to other methods and some of the design choices we make can be replaced.

5.1 Alternative embedding methods

The proposed embedding, which associates a smooth positive function with each data point obtained through the method proposed in Section 3.1, can be replaced with other methods. We have con-



Multi-Dimensional Scaling

The proposed visualization

Figure 4: A comparison between MDS and our approach for visualizing the social network of one facebook user who has 160 contacts. Many other users have experimented with the application themselves. The number of friends per user in our experiments was as high as 800, and typically between 50 and 150.

Dataset	n	Windowed mode		Full screen mode	
		Grid size	Hit rate	Grid size	Hit rate
Netflix	512	11×6	77%	18×10	91%
Movie Stars	50	8×4	95%	10×5	100%
NASDAQ	100	9×10	96%	10×10	100%
Facebook	160	10×7	88%	16×10	100%

Table 2: The ratio of lit items displayed on screen out of all lit items in each visualization. The grid size columns depict the size of the visible area of the grid in each dataset for the windowed and the full-screen modes.

ducted experiments with several of these, and attach the results as part of our Multimedia Attachment. To access the visualizations that are based on alternative methods, please scroll on the dataset choice drop-down menu below the first four datasets.

The first such alternative is traversing through the data points, one at a time, and assigning a luminance value to all icons according to the proximities between the underlying data points. We have tried several methods to determine the traversing order, such as employing 1D MDS, and performing hierarchical clustering followed by Optimal Leaf Ordering [Bar-Yoseph et al. 2001]. In all cases the results seem to be too raw to be apprehended by the viewer. The structure of the data is not easy to understand without any automatic consolidation of the data taking place before it is displayed.

On the other hand of the spectrum, one can first cluster the data and then display the clusters one after the other. The order of the display can be determined by employing a 1D embedding method based on the proximities between the clusters. To create a smooth display, the clusters morph from one cluster to the next. This display is easy to grasp, however, it is limited by associating each data point to just one cluster, and has a very limited capacity.

To overcome this, one can attempt to employ a fuzzy clustering algorithm such as the fuzzy c-means algorithm [Bezdek 1981]. The clusters can then be sorted by their similarity and displayed sequentially, changing smoothly from one cluster to the next. The weighted cluster membership value is mapped to the luminance value of each icon.

The experiments conducted exposed several limitations of the fuzzy clustering based method. First, while the smoothness-based functional-embedding method of Section 3.1 is able to assign to each data point a varying amount of total luminance, fuzzy clustering methods assign a normalized value. Trying to relieve the normalization constraint seems to somewhat help. Second, the local nature of the fuzzy c-means algorithm seems to neglect some of the correlations between the data points.

Since the smooth functional embedding method is related to Non-negative Matrix Factorization (NMF), we also try to factorize the proximity matrix using a symmetric NMF algorithm [Li et al. 2007], and to sort the rows of the resulting factor matrix in order to produce a smooth animation. Three ordering methods were tried: a greedy method that starts with the first column, finds the most similar columns and continues iteratively, a 1D MDS method based on the similarities between the rows of the resulting factor matrix, and hierarchical clustering followed by optimal leaf ordering. Despite our efforts, the NMF based methods result in jittery animations, emphasizing the need to combine the smoothness requirement into the initial optimization stage.

As mentioned above, the resulting visualizations for all alternative methods are accessible in the Multimedia Attachment. Figure 5 provides a static view of the various alternatives. Each figure color codes (heatmap) the luminance for every datapoint (rows) for every frame (columns). It can be observed that the results obtained through the method proposed in Section 3.1 are not as cluttered as the Sequential display, are more continuous in time compared to the Symmetric NMF based method and the fuzzy c-means method, and unlike the k-means based method allow each data point to be highlighted more than once.

5.2 Alternative design choices

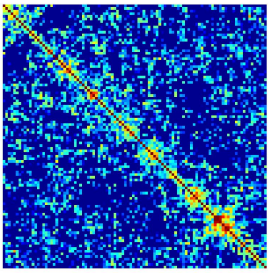
In the current implementation of the proposed visualization we choose to fix the grid locations of the icons associated with the individual data points. We believe that maintaining the spatial location is important when trying to relocate an icon we have seen previously. However, by relaxing this demand and only fixing the locations of icons that are displayed on screen at a given time we may be able to get better screen utilization, and deal with extremely large datasets. These adaptations are left for future implementations.

Throughout the paper we have focused on luminance based common fate. We have also experimented with other variables such as the color saturation (colorfulness), image blurriness or sharpness, and the perceived depth (see Figure 6). We found luminance changes to be much more appealing visually. Saturation and sharpness changes are hard to apprehend, and changes to the perceived depth create cluttered visualizations.

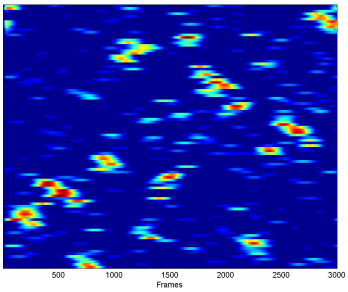
The common fate based luminance changes can be utilized in concert with other visualization methods. For example, it can be used to turn Parallel Coordinate plots into animated plots. On the other hand, other visualization techniques can be incorporated into the proposed grid visualization, for example, in order to display the source of proximity between data points that are selected. Note, however, that such modifications require information that is not contained in the affinity matrix, and may require adaptation for each type of data (graph/tabular based, dense/sparse, few/many variables).

6 Conclusions

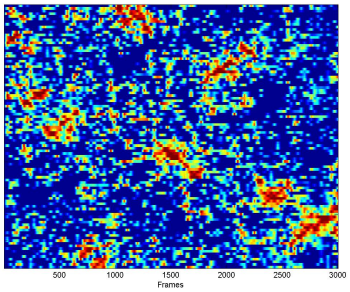
The mapping of one variable to an image axis is a common visualization metaphor. Objects are often sorted, e.g., by size, and portrayed in order. When there are several classes of objects, it is



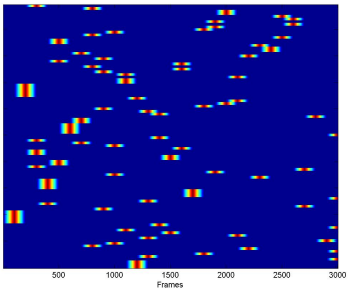
The affinity matrix



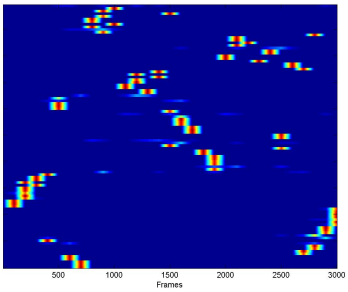
Section 3.1



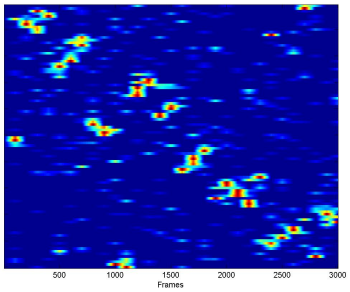
Sequential display



K-means



Fuzzy c-means



Symmetric NMF

Figure 5: A comparison between various methods of embedding the data points in time on the NASDAQ dataset. The datapoints (rows of the images) are ordered, for display purposes, according to the 1D MDS projection. The method of Section 3.1 is the Smooth Non-negative Matrix Factorization method introduced in this paper. Sequencial display is a stretching in time of the original affinity matrix, where non-diagonal elements are further emphasized. The k-means method displays one cluster at a time, and fuzzy c-means works similarly. In Symmetic NMF, the matrix is factored and the rows of the factor matrix are rearranged to create an animation which is as smooth as possible.

Figure 6: A screen capture of an alternative visualization in which the values associated with each data point at each frames were mapped to location in depth rather than to luminance. In this image a small subset containing 12 data points from the NASDAQ dataset is shown. Mapping values to depth information results in cluttered unattractive displays.

also common to group their iconic representation. It is therefore surprising that the graphical depiction of proximity in non-spatial multi-dimensional properties as positioning proximity is mostly a modern Western invention, that played little or no part of the rich visual languages developed since ancient times [Beniger and Robyn 1978]. Moreover, it seems that illustrating correlations as proximity in space is intuitively obvious only to certain educated sections of the population.

Looking for other alternatives for visualization, there is a clear psychophysical evidence, based on the Gestalt principle of Common Fate, that synchronized luminance modulation is a very effective perceptual tool [Sekuler and Bennett 2001]. This is best exemplified by our animation that shows the superiority of luminance common fate over spatial proximity in linking objects.

7 Acknowledgements

This research is supported by the Israel Science Foundation (grants No. 1440/06, 1214/06), the Colton Foundation, and The Ministry of Science and Technology Russia-Israel Scientific Research Cooperation.

References

BAECKER, R. 1998. Sorting out sorting: A case study of software visualization for teaching computer science. In *Software Visualization – Programming as a Multimedia Experience*, The MIT Press, 369–382.

BALASUBRAMANIAN, M., SCHWARTZ, E. L., TENENBAUM, J. B., DE SILVA, V., AND LANGFORD, J. C. 2002. The Isomap algorithm and topological stability. *Science 295*, 5552, 7.

BAR-YOSEPH, Z., GIFFORD, D. K., AND JAAKOLA, T. S. 2001. Fast optimal leaf-ordering for hierarchical clustering. *Bioinformatics 17*.

BELKIN, M., AND NIYOGI, P. 2001. Laplacian Eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, 585–591.

BENIGER, J. R., AND ROBYN, D. L. 1978. Quantitative graphics in statistics: A brief history. *The American Statistician 32*, 1, 1–11.

BEZDEK, J. 1981. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum, New York.

ELLIS, G., AND DIX, A. 2006. Enabling automatic clutter reduction in parallel coordinate plots. *IEEE Transactions on Visualization and Computer Graphics 12*, 5, 717–724.

FUA, Y.-H., WARD, M. O., AND RUNDENSTEINER, E. A. 1999. Hierarchical parallel coordinates for exploration of large datasets. In *VIS '99: Proceedings of the conference on Visualization '99*, 43–50.

GAVRILOV, M., ANGUELOV, D., INDYK, P., AND MOTWANI, R. 2000. Mining the stock market: Which measure is best? In *ACM Int. Conf. Knowledge Discovery and Data Mining*, 487–496.

HEER, J., AND ROBERTSON, G. 2007. Animated transitions in statistical data graphics. In *IEEE Information Visualization (InfoVis)*.

INSELBERG, A., AND DIMSDALE, B. 1990. Parallel coordinates: a tool for visualizing multidimensional geometry. In *IEEE Visualization*, 361–378.

LEE, D. D., AND SEUNG, H. S. 2001. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13*, MIT Press, 556–562.

LI, T., DING, C., AND JORDAN, M. I. 2007. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *ICDM*, IEEE Computer Society, 577–582.

LIMOGES, S., WARE, C., AND KNIGHT, W. 1989. Displaying correlations using position, motion, point size or point colour. In *Graphics Interface '89*, 262–265.

MICHOTTE, A. 1963. *The perception of causality*. Basic Books, New York.

NOCEDAL, J., AND WRIGHT, S. 1999. *Numerical Optimization,*. Springer Verlag.

NOVOTNY, M., AND HAUSER, H. 2006. Outlier-preserving focus+context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics 12*, 5, 893–900.

PIERRE DEMARTINES AND JEANNY HERAULT. 1997. Curvilinear Component Analysis: A Self-Organizing Neural Network for Nonlinear Mapping of Data Sets. *IEEE Transactions on Neural Networks 8*, 1.

RAO, R., AND CARD, S. K. 1994. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, 318–322.

ROWEIS, S. T., AND SAUL, L. K. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science 290*, 5500, 2323–2326.

SAMET, H. 1984. The quadtree and related hierarchical data structures. *ACM Comput. Surv. 16*, 2, 187–260.

SEKULER, A., AND BENNETT, P. 2001. Generalized common fate: Grouping by common luminance changes. *Psychological Science 12*, 6, 437–444.

SHEPARD, R. 1962. The Analysis of Proximities: Multidimensional Scaling with an Unknown Distance Function (Part I). *Psychometrika 27*, 125–140.

STASKO, J. T. 1990. Tango: A framework and system for algorithm animation. *Computer 23*, 9.

TREISMAN, A., AND GORMICAN, S. 1988. Feature analysis in early vision: Evidence from search asymmetries. *Psychological Review 95*, 1, 15–48.

TVERSKY, B., MORRISON, J. B., AND BETRANCOURT, M. 2002. Animation: can it facilitate? *Internation Journal Human-Computer Studies 57*, 4, 247–262.

VENNA, J., AND KASKI, S. 2007. Comparison of visualization methods for an atlas of gene expression data sets. *Information Visualization 6*, 2, 139–154.

VON LUXBURG, U. 2007. A tutorial on spectral clustering. *Statistics and Computing 17*, 4, 395–416.

WARE, C., AND BOBROW, R. 2005. Supporting visual queries on medium-sized node-link diagrams. *Information Visualization 4*, 1, 49–58.

WISE, J., THOMAS, J., PENNOCK, K., LANTRIP, D., POTTIER, M., SCHUR, A., AND CROW, V. 1995. Visualizing the non-visual: spatial analysis and interaction with information from text documents. In *Proc. Info. Visualization*, 51–58.

WOLF, L., AND DONNER, Y. 2008. An experimental study of employing visual appearance as a phenotype. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on 0*, 1–7.

WOLF, L., AND DONNER, Y. 2008. Local regularization for multiclass classification facing significant intraclass variations. 748–759.

Realistic Real-time Underwater Caustics and Godrays

C. Papadopoulos [1,2], G. Papaioannou [1]
[1]Department of Informatics, Athens University of Economics and Business, Athens, Greece
[2]Department of Computer Science, State University of New York at Stony Brook, New York, USA

Abstract

Realistic rendering of underwater scenes has been a subject of increasing importance in modern real-time 3D applications, such as open-world 3D games, which constantly present the user with opportunities to submerge oneself in an underwater environment. Crucial to the accurate recreation of these environments are the effects of caustics and godrays. In this paper, we shall present a novel algorithm, for physically inspired real-time simulation of these phenomena, on commodity 3D graphics hardware, which can easily be integrated in a modern 3D engine.

Keywords: *I.3.7, Three-Dimensional Graphics and Realism, Color, shading, shadowing, and texture.*

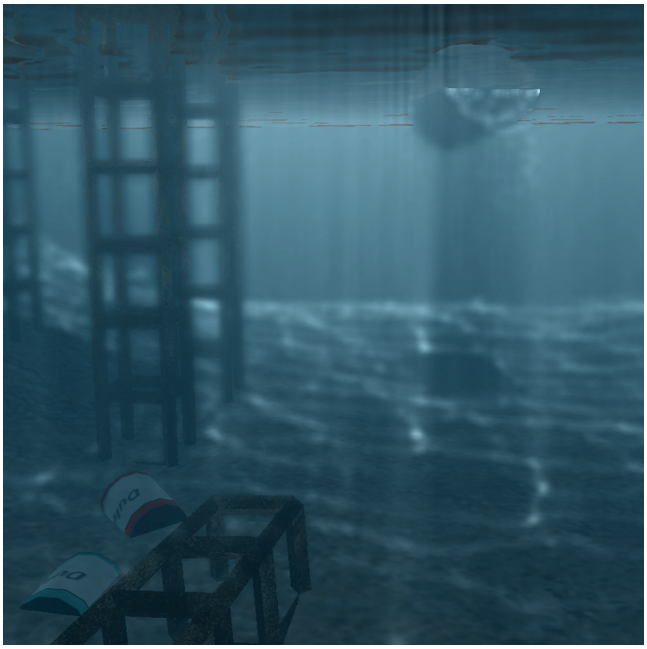


Figure 1: A screenshot of our caustic and godray creation algorithm, running at an excess of 100 fps, at a resolution of 800x800.

1. INTRODUCTION

Caustics are a result of the convergence of light at a single point. The phenomenon occurs when light interacts with a reflective or refractive surface, where rays deviate from their initial directions and focus on certain regions. *Crepuscular rays* or *godrays* are a result of photons outscattering from their path due to the presence of particles in a participating medium. In this paper we shall present a novel method for a physically inspired simulation of these two phenomena in real time (Figure 1).

2. PREVIOUS WORK

Even though the underwater illumination has been the subject of extensive research (mostly in the domain of particle and ray tracing algorithms), real-time solutions to the matter of realistic caustic formation are limited. Methods for the simulation of crepuscular rays (or godrays) are even sparser. In this section, a brief overview of some of the existing work in the area will be presented, which is mainly focused on offline rendering or interactive illumination simulation, but also includes real-time techniques for modeling and displaying caustics and godrays.

Offline solutions can create extremely realistic underwater caustics and godrays, at a significant computational cost. One of the first methods that allowed for caustic creation is forward ray tracing [1], which differs from conventional ray tracing in the sense that rays are cast from the light towards the scene. Rays that intersect the camera clipping plane contribute to the intensity of the respective pixel. An alternative (and faster) method is bidirectional ray tracing [2],[3], that traces rays both from the camera and the light source and 'connects' the ray paths to find the radiance contribution to the corresponding pixel. Photon mapping, a global illumination solution presented in [4], utilizes forward ray tracing along with probabilistic techniques, importance sampling and an illumination gathering step to achieve relatively high performance and high quality caustics. It also permits the creation of godrays in participating media (via ray marching). However, despite the improvements when compared to conventional forward ray tracing, photon mapping is also limited to off-line applications.

One of the first proposals for real-time caustic creation was by [5] and involved the projection of a precomputed texture onto the scene geometry using additive blending. Animation was achieved via texture coordinate perturbation. Obviously, while being fast, this method did not produce physically correct caustics that animated in tandem with the water surface. Another approach, by [6] assumes that caustics are formed by rays emanating from the water surface directly above the point of interest and uses Snell's law to produce refracted rays. It modulates intensity based on the direction of these rays and alternatively uses the vectors to index a light map. Obviously the caustics produced by this method are not physically correct, as a result of the above assumption.

Shah et al. [7] presented a novel method for creating refractive caustics in image-space. This method involves creating a vertex grid out of the refractive geometry and then splatting the vertices onto the scene using an image-space ray-scene intersection algorithm. However caustic quality is directly related to the tessellation of this vertex grid. Furthermore, this method does not provide a solution to the issue of godrays that is part of the same illumination problem.

Research concerning godray creation falls into two main categories, the first being the modeling of the rays as light shafts. [8] presented a method that modeled lightshafts as the front-facing surfaces of 'illumination volumes'. However, the computational cost for this method was very high and, at its time, it did not provide a real time solution. Lanza [9] proposed a similar method that used parallelepipeds, which are transformed via a vertex shader program

so as to animate in tandem with the water surface. This method, while producing very 'dramatic' results, requires a high density parallelepiped grid in order for the effect to be realistic, which in turn results in a very high fill rate cost. Furthermore, the final intensity computation for the godrays does not account for individual ray attenuation, as it is done in a post-processing pass, during which, information for each individual ray is not available.

The second category of godray creation algorithms focuses around the sampling of the visible distance in front of the viewer by rasterizing planar surfaces parallel to the near clipping plane. Dobashi et al. [10] uses precomputed integral values (saved in 2D textures) in order to compute the intensity value at each plane, which are then accumulated in the frame buffer. Jensen [11] performs a per-fragment projective texture read from a caustics map, for each fragment on a sampling plane. Both of these methods present one main drawback, since a large number of sampling planes is required in order to avoid artifacting. As these planes are represented by screen-space quads, there is a significant fill-rate premium to pay for their rasterization. Furthermore, in the case of Dobashi et al., there is no direct way for the algorithm to control animation of the light shafts (since it would require recomputation of the 2D lookup textures), while as far as Jensen's method is concerned, the result is not physically accurate. In addition, the non-linear nature of these projection transformations on the view-oriented planes produces non-uniformly spaced samples and may result in curved godrays.

Another type of method that can handle caustics and scattering due to complex refractive objects involves ray-marching through a voxel grid of the refractive geometry. 'Eikonal rendering' [12] is such a technique, and while it allows for real-time framerates during viewport changes, if the refractive object or the light position are altered, the lighting distribution has to be recalculated (a process that can take several seconds) making it unsuitable for simulating underwater effects due to their constantly shifting nature. Another technique in this area is presented by Sun et al.[13] who use an oct-tree data structure to store voxel data along with an adaptive photon-tracing step to recompute the radiance volumes at interactive frame rates (approximately 8 fps). However, while successfully dealing with arbitrary and non-uniform refractive geometries, this method also is too computationaly intensive to be used in a real-time generic 3D application.

Finally, Krüger et al. [14] proposed a method for generic caustic simulation that utilizes the rasterization of lines compacted as texture rows to compute the intersection points of photon rays with scene geometry. While this method can be quite accurate when all occluders lie inside the camera frustum and can also handle multiple bounces via depth peeling, it fails to calculate intersection points for off-frustum occluders, which is frequently the case in underwater scenes. Furthermore, the brute-force texture-space intersection algorithm imposes a high fill-rate cost and requires multiple passes to calculate an adequate number of particle collisions, as the data for each particle intersection estimation effectively occupies one texture row (texture space scan-line).

Our method is a real-time approximation of the photon mapping algorithm for underwater caustics (and volumetric caustics) generation. It utilizes the image-space ray-scene intersection method by Shah et al. [7], but in contrast to the original technique, it decouples the effect accuracy from the refractive geometry by tracing uniformly distributed photons from the light source towards the scene. Caustics are modeled as point primitives of variable size created on the intersection points, while scattering due to the participating medium is simulated using line rasterization between the water surface and these points. Intensity calculations are made on a per-photon basis (allowing for realistic attenuation and outscattering). Finally the equivalent of the photon mapping gathering step is a filtering pass that spreads the particle contribution over a certain area and reduces aliasing. Our technique achieves very high framerates on commodity graphics accelerators and can be easily integrated within any modern 3D engine.

3. METHOD OVERVIEW

Our method realistically approximates caustic creation, by casting photons from the light source evenly distributed over a grid. These photons are intersected against the scene geometry, and point primitives of variable size are created at the intersection points. For godrays, the intersection points are discovered in a similar way, but line primitives are spawned instead. Our method utilizes a deferred rendering approach and makes extensive use of render-to-texture and programmable shader capabilities of modern graphics hardware (vertex/fragment and geometry shaders). Following is a high-level overview of the rendering pipeline:

1. Frame preparation
    a. Calculate the new position and look-at vector for the rendering pass from the light's perspective.
2. Shadow map creation
    a. *(In light-space)* Render the shadowmap.
    b. *(In light-space)* Optionally, render a mask representing the water surface into the shadowmap's color attachment, allowing for the algorithm to test photons against the water's surface only and therefore eliminate those that do not enter the water mass.
3. Render scene geometry
    a. *(In camera-space)* Render the scene geometry into a render target, from the camera's point of view.
4. Optional reflection/refraction rendering passes
5. Caustics Rendering
    a. *(In light-space)* Calculate intersection points of rays with the scene geometry.
    b. *(In camera-space)* Emit point primitives and rasterize them with additive blend.
    c. *(In camera-space)* Filter caustics.
6. Godray Rendering
    a. *(In light-space)* Calculate intersection points of rays with the scene geometry.
    b. *(In camera-space)* Emit line primitives and rasterize them with additive blend.
    c. *(In camera-space)* Filter godrays.
7. Compose final image
    a. *(In camera-space)* Use additive blending to compose the final image using input from steps 3,4,5 and 6.
    b. *(In camera-space)* Optionally apply other screen-space post processing effects like ambient occlusion and motion blurring.

Since our goal was to focus around underwater caustics and godrays, the algorithm supports the tracing of a single refraction. It can however be extended to support multiple refractive interfaces via depth peeling ([7], [14]). A visual representation of the algorithm process that demonstrates how results from each step are combined, is presented in Figure 2.

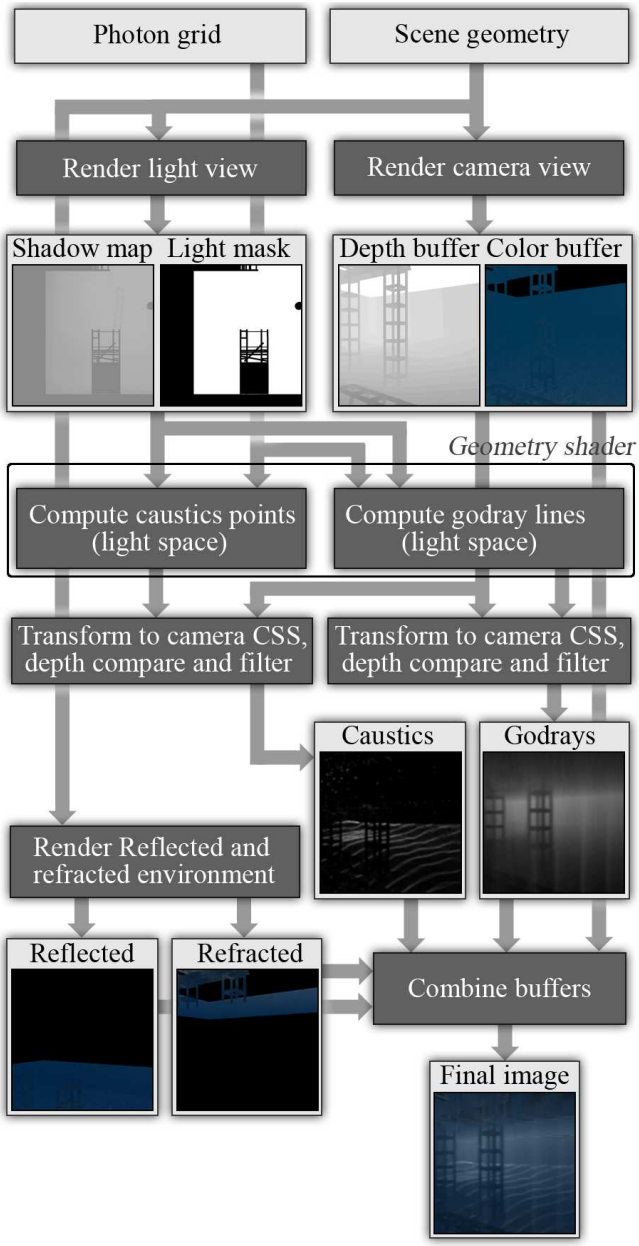


Figure 2: Overview of the godray and caustic creation algorithm.

3.1 Frame Preparation

Since we cast a concentrated but small number of photons onto the geometry, we must ensure that most of them intersect the visible portion of the scene. Therefore, the light frustum (and the photon grid) is bound to follow the camera frustum.

For directional (infinite) light sources, their position $\mathbf{p}_{light}$ tracks the camera frustum, whereas for point lights, the position remains fixed. In both cases though, the look-at vector $\vec{\mathbf{l}}$ points at the middle of the camera view frustum $\mathbf{p}_{mid}$.

$$\vec{\mathbf{l}} = \mathbf{p}_{mid} - \mathbf{p}_{light}$$

$$\mathbf{p}_{mid} = \mathbf{p}_{viewer} + \frac{z_{near}+z_{far}}{2} \cdot \vec{\mathbf{l}}_{viewer}$$

where z_{near}, z_{far} are the camera's near and far clipping distances and $\vec{\mathbf{l}}_{viewer}$ is its look-at vector.

3.2 Render Scene Geometry

In this step we render the scene geometry into an off-screen buffer. If necessary, fog calculations can also be applied here. Furthermore, the Z-buffer information from this pass is captured, as it is required in the following steps to depth-test the caustics and godrays.

3.3 Caustics Rendering

We have mentioned that our process involved casting a photon grid onto the scene. In this way, the emission of photons from the light source towards the scene is simulated. The grid is modeled in the light's canonical screen space as an array of points, with the required tessellation. During this step of the algorithm, a low resolution grid is sent to the GPU for rendering. Then, using a geometry program, each grid cell is subdivided to produce the desired number of points, ensuring a dense photon distribution. An alternative adaptive subdivision scheme has been proposed by Wyman et al. in [15]. However, usage of this scheme is not justified in our method, since it requires feedback during each subdivision step and is better suited for generic caustics simulations in which the refractor has limited screen-space coverage and possesses an arbitrary shape. The geometry shader also performs the refraction, point splatting and godray modeling operations. The grid point currently being calculated is unprojected from light canonical screen space coordinates into world coordinates and modeled as a ray ($\vec{r_i}$), which is then intersected against the water surface. With the surface intersection point $\mathbf{p_s}$ and normal $\vec{\mathbf{n_s}}$ of the water surface known, $\vec{\mathbf{r_i}}$ is refracted to produce the refraction direction $\vec{\mathbf{r_t}}$. Both $\vec{\mathbf{r_t}}$ and $\mathbf{p_s}$ are passed to the image space intersection algorithm by Shah et al. [7], along with the shadow map texture. The algorithm uses the Newton-Rhapson (NR) derived iterative method to approximate the solution to a function $f(d)$ with d being the distance to intersection point $\mathbf{p_i}$ from $\mathbf{p_s}$ along $\vec{\mathbf{r_r}}$. In order to do so, an estimated intersection point $\mathbf{p_e} = \mathbf{p_s} + d_{estimate} * \vec{\mathbf{r_t}}$ is assumed, along with its projection into light-space $\mathbf{p_{proj_{pe}}}$ and the algorithm approximates the solution to $f(d) = \mathbf{p_e} - \mathbf{p_{proj_{pe}}}$. In [7], it is shown that this NR-derived estimator tends to converge to a value of d so that $\mathbf{p_s} + d * \vec{\mathbf{r_t}}$ is the surface intersection point $\mathbf{p_i}$. At $\mathbf{p_i}$ we then emit a point primitive. A schematic overview of the photon casting process can be seen in Figure 3.

These primitives are transformed to camera space and rendered using additive blending with their size corresponding to their screen-space coverage. Consiquently, point size is actually a significant factor in computation of the final caustics intensity. If all points emitted have a constant size then the projection of the distant point primitives would overlap on the view plane, resulting in much brighter caustics than the ones close to the camera. On the other hand, if the point size is not large enough, photons rasterized close to the viewer do not superimpose each other, leading to inadequate splatting that cannot simulate the gathering stage of particle tracing and produces noise artifacts. The solution to this issue is to regulate the size of the points based on their distance from the camera (see Figure 4). The final point size is calculated as follows:

$$s_{final} = a + b/d_{pointFromViewer}$$

with $a = s_{max} - \frac{z_{far} \cdot (s_{max} - s_{min})}{z_{far} - z_{near}}$

and $b = \frac{z_{near} \cdot z_{far} \cdot (s_{max} - s_{min})}{z_{far} - z_{near}}$

S_{max} and S_{min} are minimum and maximum point sizes.

The emitted point primitives are then rasterized (with additive blending) in a high-accuracy render target and Z-tested against the camera's depth buffer (already available from step 3). The final

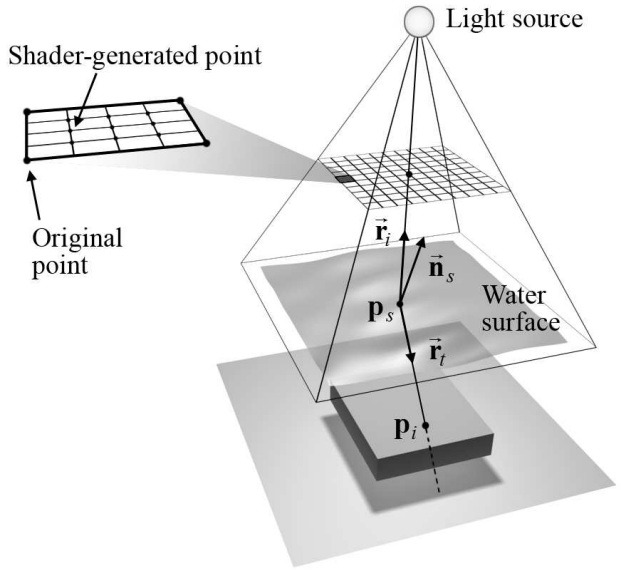


Figure 3: The photon casting process.

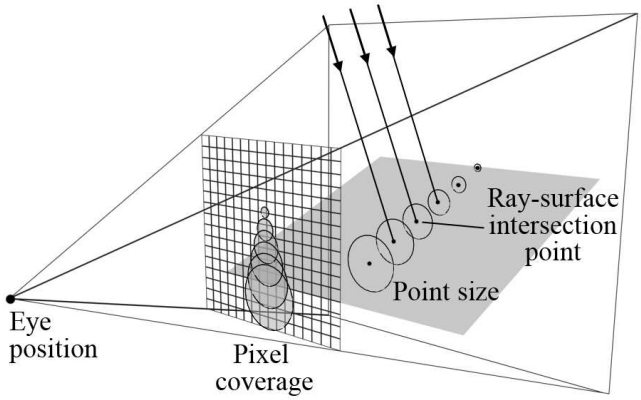


Figure 4: Distance-regulated point size to account for the non-projective hardware-based drawing of point primitives.

intensity value is produced from this formula:

$$I_{final} = I_{photon} \cdot e^{-\gamma \cdot d_{fromSurface}}$$

where γ is the medium attenuation parameter and $d_{fromSurface}$ is the distance of the photon from the water surface. This formula only takes $d_{fromSurface}$ into account, since the attenuation based on the distance from the viewer is intrinsically handled via the point size regulation described above.

3.4 Godray Rendering

In this step, a grid of rays is cast and intersected with the scene (similar to step 5). The main difference lies in that, instead of emitting a single point primitive at the ray-scene intersection point $\mathbf{p_i}$, a line primitive that starts at water surface $\mathbf{p_s}$ and ends at $\mathbf{p_i}$ is emitted. The line primitives are transformed to camera-space, then rasterized with additive blending in a high-accuracy render target and Z-tested against the camera's depth buffer. The formula that produces the final intensity value per godray fragment is the following:

$$I_{final} = I_{photon} \cdot Mie(\theta) \cdot e^{-\gamma \cdot d_{fromViewer}} \cdot e^{-\gamma \cdot d_{fromSurface}}$$

$Mie(\theta)$ is a Mie scattering phase function with θ being the angle between the ray direction and the vector from the viewer to the fragment. $d_{fromSurface}$ is the distance of the fragment from the water surface.

3.5 Filtering and Composition

In both of the above steps, the result can display some aliasing in the low intensity areas. To counter this issue, after calculating the effects, we apply a multi-tap low pass filter with a rotating sampling kernel that reduces noise. Comparison between the filtered and unfiltered results can be see in Figure 5.

Finally, the intermediate images (color/reflection/refraction/filtered godray/filtered caustics buffers) from the above steps are combined into the end result. At this point, the water surface geometry is rendered as well. The final product of our algorithm can be seen in Figures 6,7,8.

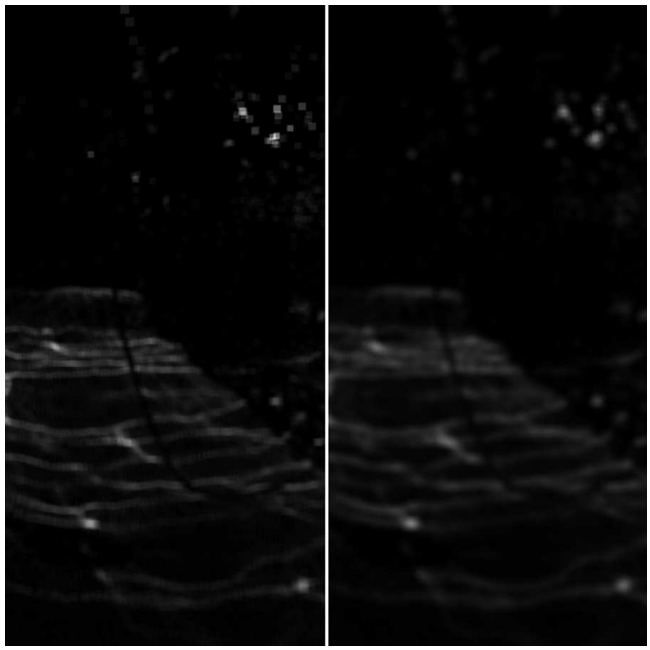


Figure 5: Detail demonstrating the differences between unfiltered (left) and filtered caustics (right).

Figure 6: Our algorithm, rendering a small port, running at a resolution of 1440×850, with a framerate of 60+ fps.

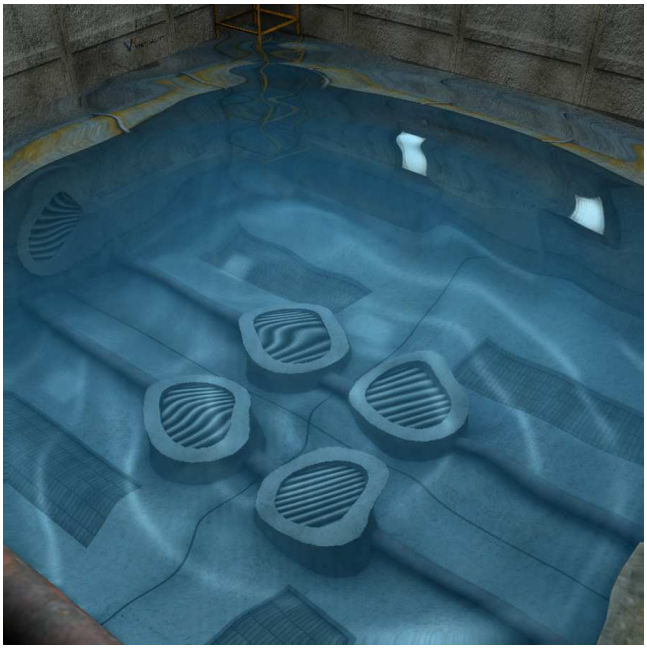


Figure 7: A water tank scene rendered with our algorithm. The scene runs at more than 120 fps in a window of 800×800 pixels.

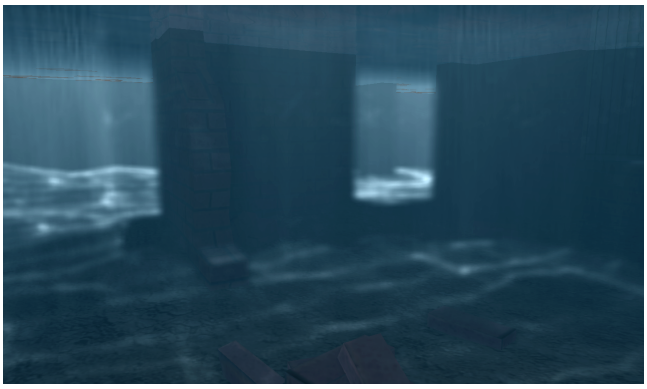


Figure 8: A flooded storage area scene rendered with our algorithm in a 1440×850 resolution. The scene runs at more than 60 fps.

4. IMPLEMENTATION AND RESULTS

4.1 Implementation Details

We have implemented the caustic and godray formation algorithm using OpenGL and GLSL. The render targets are implemented as OpenGL Frame Buffer Objects with textures of various sizes bound to the respective attachment points.

In order to improve performance, we regulate the sizes of the render targets to a fraction of the final frame buffer resolution. Specifically, in our demo application the caustic render target resolution is 512×512 (compared to the 800×800 default resolution for the main viewport), whereas the godray render target is 128×128 pixels. This results in a significant performance boost (since the godray rendering process is fill-rate intensive) with only a small cost in accuracy. If the application viewport is resized, the back-buffers are also resized to preserve this scale factor.

In our implementation, the photon grids are initialized on application startup and rendered during each frame. The volumetric nature of the godray effect requires a smaller amount of cast photons when compared to the caustic effect in order to achieve satisfactory detail. Thus, two separate photon grids are created, with the first one being 200×200 points large, and the second one being 100×100 points. Inside the geometry shader, these points can be further tessellated in order to improve detail. Dynamic tesselation is possible but greatly impacts peformance (by as much as 50%) as the shader compiler is unable to optimize the shader code by unwinding the primitive generation loop.

For our test cases (and the respective demo application) we have created a single light positioned at a very large heigth above the scene in order to simulate a directional light source with an intensity of $9.2\frac{w}{sr}$. The exponential attenuation factor γ used is $0.13m^{-1}$. Also, we have utilized a simplified version of the Henyey-Greenstein phase function [16] in order to compute the $Mie()$ term.

We mentioned in the previous section that a crucial part of caustic production is the regulation of the size of the generated points. In our implementation we have set a minimum point size of 5 and a maximum of 20. These numbers can be adjusted according to the resolution of the cast photon grid (with lower resolutions requiring larger point sizes in order to compensate). Similarly, the width of the lines spawned in the godray portion of the algorithm has to be modified to compensate for a reduction in grid resolution.

In our tests, we provide two different models for the procedural generation of water surface elevation and normals. The first one is a simple circular cosine function with a small noise contribution that is read off a Perlin Noise [17] texture, that provides a familiar and easily comparable effect. The second model consists of a cosine value along one of the world axes with a significant Perlin Noise contribution, approximating the turbulent waves of a water surface. In both models, the amplitude of the water surface can be regulated by the user during run time.

4.2 Results / Tests

We conducted our tests on a system with an Intel Core 2 Duo E4500 processor running at 2.2 Ghz, with 2 gigabytes of ram and an NVidia GTX260 GPU with 216 stream processors. With a viewport resolution of 800×800 for the main window, with a 512×512 caustics buffer, a 128×128 godray buffer and grids for the caustics and godrays of $200 \times 200 \times 4$ and $100 \times 100 \times 4$ photons respectively, the frame rate exceeds 110 fps. At a viewport resolution of 1440×850, while maintaining the same grid resolutions, the frame rate still remains highly interactive, exceeding 60 fps.

If the grid resolutions are increased to $500 \times 500 \times 4$ (for a total of 1000000 photons) for the caustics portion and to $300 \times 300 \times 4$ (for a total of 360000 photons) for the godrays portion, the framerate still remains interactive on our test system (ranging from 20 to 30 fps). At these resolutions, aliasing effects are no longer noticeable, making the filtering passes unnecessary (as can be seen for the godray portion of the algorithm in Figure 9).

In order to test how the algorithm scales with respect to back-buffer resolution, we performed measurements with a viewport resolution of 1440×850, while increasing the godray back-buffer to 460×272 (a 4-fold increase in resolution compared to the default setting). To compensate, we also increased the screen-coverage of the line primitives to 12 pixels. Finally, we augmented the grid resolutions to $400 \times 400 \times 4$ for caustics generation and $200 \times 200 \times 4$ for godrays and regulated the point primitive size (smallest 2 pixels, largest 10 pixels) to sharpen the caustics. Despite the increase in back buffer resolution and fill-rate cost, the frame-rate still remains highly interactive (20 frames per second). Again, with this increase

in back-buffer size and grid resolutions, filtering of the effects becomes unnecessary. Results of this configuration can be seen in Figure 10.

The screen captures presented in this paper have had their color balance and gamma slightly enhanced in order to improve legibility on printed media.

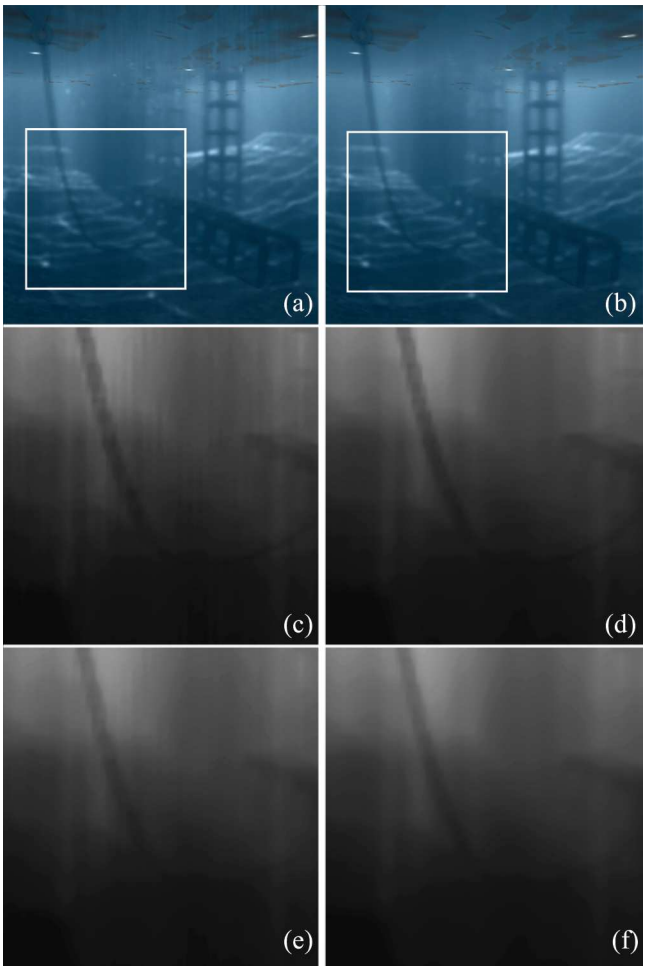


Figure 9: Figure demonstrating the results of the godray portion of the algorithm with different grid resolutions. Left column: (a) Final result with 40000 photons. (c) unfiltered godrays. (e) post-filtered godrays. Right column: (b) Final result with 360000 photons. (d) unfiltered godrays. (f) post-filtered godrays.

5. CONCLUSIONS

We have presented a novel algorithm for the creation of real-time underwater caustics and godrays. The algorithm achieves realistic results at high framerates on consumer graphics hardware. Due to the utilization of rendering stages and buffers encountered in modern graphics engine implementations, our method can be easily integrated into any 3D engine.

6. ACKNOWLEDGEMENTS

The work presented in this paper is funded by the Athens University of Economics and Business Special Account for Research Grants (EP-1600-10/00-1).

7. REFERENCES

[1] J. Arvo, "Backwards ray tracing," in *SIGGRAPH Course Notes*, 1986, vol. 12, p. 100.

[2] E. P. Lafortune, "A theoretical framework for physically based rendering," *Computer Graphics Forum*, vol. 13, pp. 97–107, 1994.

[3] E. Veach and L. J. Guibas, "Bidirectional estimators for light transport," in *Eurographics Rendering Workshop 1994*, 1994, pp. 147–162.

[4] H. W. Jensen, *Realistic image synthesis using photon mapping*, A. K. Peters, Ltd., Natick, MA, USA, 2001.

[5] J. Stam, "Random caustics: Wave theory and natural textures revisited," in *Visual Proceedings of SIGGRAPH 1996*, 1996, p. 151, Available at: `http://www.dgp.toronto.edu/people/stam/INRIA/caustics.html`.

[6] J. Guardado and D. Sanchez-Crespo, "Rendering water caustics," in *NVidia GPU Gems*, pp. 31–44. Addison-Wesley, 2004.

[7] M. A. Shah and J. Konttinen, "Caustics mapping: An image-space technique for real-time caustics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 2, pp. 272–280, 2007, Member-Pattanaik, S.

[8] K. Iwasaki, T. Dobashi, and T. Nishita, "An efficient method for rendering underwater optical effects using graphics hardware," in *COMPUTER GRAPHICS forum*, 2002, vol. 21, pp. 701–711.

[9] S. Lanza, "Animation and rendering of underwater godrays," in *SHADERX 5*, pp. 315–327. Charles River Media, 2007.

[10] Y. Dobashi, T. Yamamoto, and T. Nishita, "Interactive rendering of atmospheric scattering effects using graphics hardware," in *Graphics Hardware*, 2002, pp. 99 – 108.

[11] L. Jensen, "Deep-water animation and rendering," 2001, Available at: `http://www.gamasutra.com/view/feature/3036/deep_water_animation_and_rendering.php`.

[12] I. Ihrke, G. Ziegler, A. Tevs, C. Theobalt, M. Magnor, and H. P. Seidel, "Eikonal rendering: efficient light transport in refractive objects," in *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, New York, NY, USA, 2007, p. 59, ACM.

[13] X. Sun, K. Zhou, E. Stollnitz, J. Shi, and B. Guo, "Interactive relighting of dynamic refractive objects," in *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, New York, NY, USA, 2008, pp. 1–9, ACM.

[14] J. Krüger, K. Bürger, and R. Westermann, "Interactive screen-space accurate photon tracing on gpus," in *Rendering Techniques (Eurographics Symposium on Rendering - EGSR)*, June 2006, pp. 319–329.

[15] C. Wyman and G. Nichols, "Adaptive caustic maps using deferred shading," *Computer Graphics Forum*, vol. 28, no. 2, pp. 309–318, 2009.

[16] L. G. Henyey and J. L. Greenstein, "Diffuse radiation in the Galaxy," *Annales d'Astrophysique*, vol. 3, pp. 117, 1940.

[17] K. Perlin, "Improving noise," in *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 2002, pp. 681–682, ACM.

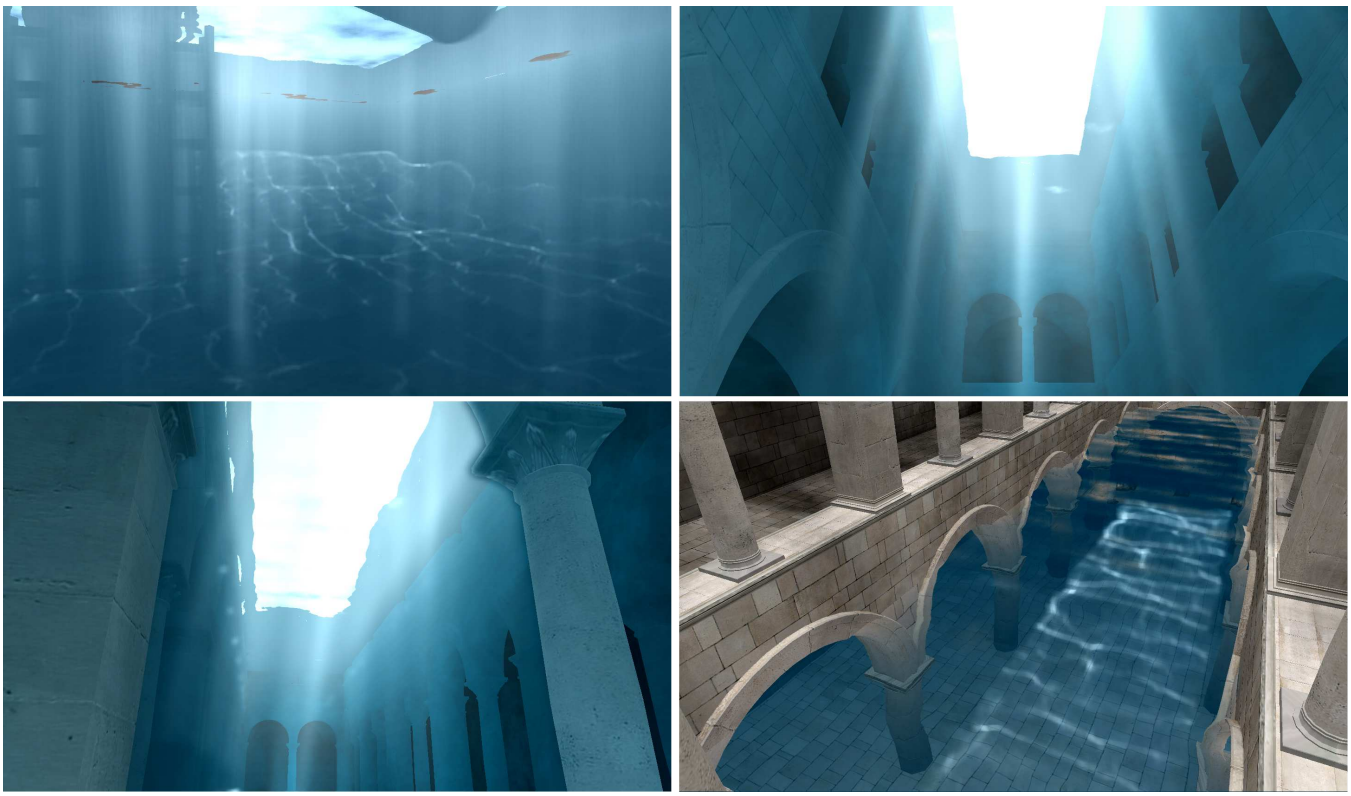


Figure 10: Final results of our algorithm with high grid resolutions, an increased godray back-buffer resolution, regulated primitive sizes and screen-space ambient occlusion approximation. For all these test cases, the method still allows for interactive frame-rates.

ABOUT THE AUTHORS

Charilaos Papadopoulos received a BSc in Informatics from the Athens University of Economics and Business and commenced his PhD studies in Computer Science at the State University of New York at Stony Brook in the Fall of 2009. His research and teaching interests focus around computer graphics, real-time photo-realistic rendering and visualization. His contact email is `papado@aueb.gr` and his personal website can be found at `http://graphics.cs.aueb.gr/users/papado`.

Georgios Papaioannou received a BSc in Computer Science in 1996 and a PhD degree in Computer Graphics and Pattern Recognition in 2001, both from the University of Athens, Greece. From 2002 till 2007 he has been a virtual reality systems engineer and developer at the Foundation of the Hellenic World. Dr. Papaioannou has been teaching elementary and advanced computer graphics, programming and human-computer interaction courses since 2002. He is currently a lecturer at the Department of Computer Science of the Athens University of Economics and Business and his research is focused on real-time computer graphics algorithms, photorealistic rendering, virtual reality and three-dimensional shape analysis. He is a member of IEEE, ACM and SIGGRAPH. His contact email is `gepap@aueb.gr`.

Edge Tracking of Textured Objects with a Recursive Particle Filter

Thomas Mörwald* Vienna University of Technology
Michael Zillich† Vienna University of Technology
Markus Vincze‡ Vienna University of Technology

Abstract

This paper proposes a new approach of model-based 3D object tracking in real-time. The developed algorithm uses edges as features to track, which are easy and robust to detect. It exploits the functionality of modern highly parallel graphics boards by performing hidden face removal, image processing, texturing and particle filtering. Using a standard 3D -model the tracker requires neither memory and time -consuming training nor other pre-calculations. In contrast to other approaches this tracker also works on objects where geometry edges are barely visible because of low contrast.

Keywords: Object tracking, edge detection, 3D tracking, real-time, image processing, feature matching

Best to read in color.

1 Introduction

Tracking the pose of a three dimensional object from a single camera is a well known task in computer vision. What seems to be simple for humans turns out to be significantly more complicated for computers. While humans are able to perform highly parallel image processing, even modern *central processing units (CPUs)* have problems calculating the pose of an object with sufficient accuracy, robustness and speed. This leads to the idea of using modern graphics cards, which also work in parallel, to solve this problem. This approach exploits the parallel power of *graphics processing units (GPUs)* by comparing edge features of camera images and a 3D -model.

Graphics boards are designed to render virtual scenes as realistically as possible. The main idea is to compare those virtual scenes with an image captured from reality. Texturing is a common method of simulating realistic surfaces. In this paper, the edges of those textures are used for comparison. Fast progress in computer science will soon allow the inclusion of more and more optical effects like shadows, reflections, shading, occlusions or even smoke, fire, water or fog.

1.1 Related work

One of the first successful approaches of tracking objects by their edges was RAPiD [Harris 1992]. It uses points on edges and searches for correspondence to its surroundings along the edge gradient. However this method lacks robustness and several improvements were applied to overcome this problem as in [Drummond and Cipolla 1999; Philipp Michel 2008; Luca Vacchetti and Fua 2004; Klein and Drummond 2003].

Another approach is to globally match model primitives with those from the camera image [Lowe 1992; Gennery 1992; D. Koller and Nagel 1993; Kosaka and Nakazawa 1995; A. Ruf and Nagel 1997]. This method has been used for robot and car tracking applications, but was later replaced by improved versions based on RAPiD.

*e-mail: moerwald@acin.tuwien.ac.at
†e-mail: zillich@acin.tuwien.ac.at
‡e-mail: vincze@acin.tuwien.ac.at

[Lucie Masson and Jurie 2004] also uses edges and textures for tracking. They extract point features from the texture and use them together with the edges to calculate the pose. This turns out to perform very fast and robust against occlusion. Our approach not only uses patches but the whole texture, which usually lets the pose converge very quickly to the accurate pose. Since the algorithm runs on the GPU, it is as fast as the method in [Lucie Masson and Jurie 2004].

The work presented in [M. Vincze and Zillich 2001] uses edge features to track but does not take into account texture information. This makes it less robust against occlusion. Since the search area in that approach is very small, it is also less robust against fast movement and getting caught in local minima.

The work presented in this paper is based on [Klein and Murray 2006] where they take advantage of graphics processing by projecting a wireframe model into the camera image. Then a particle filter with a Gaussian noise model is used to evaluate the confidence level with respect to the pose.

Figure 1: *Edges from geometry vs. edges from texture*

Our approach not only uses geometry edges but also edge features from textures which extends the class of trackable models by those that have curved surfaces as illustrated on the right of Figure 1. This is because in a standard 3D -model curvature is approximated by triangles and quadrangles which would produce virtual edges which do not correspond to the actual edges as shown on the left of Figure 1. The particle filter is extended by using it in a recursive design that evaluates a single pose estimate given by the most likely particles.

1.2 Overview

The idea of this approach is to

- extract the edges from the incoming camera image,
- extract the edges from the textured 3D -model,
- generate hundreds of slightly different views of the model relative to a pose estimate,
- calculate the most likely pose of the model by matching the edges of the camera image and the 3D -model.

The algorithm developed is separated into two parts. The *preprocessing* in Section 2, where all possible pre-calculations are made

and the *recursive particle filtering* in Section 3, which generates several hundred poses and for each of them evaluates the confidence level. Therefore the second part is very time -crucial. A *linear Kalman filter* described in Section 4 is applied for smoothing the resulting trajectory. Section 5 gives some hints on how to implement the proposed methods. The *results* in Section 6 and the *conclusion* in Section 7 summarize the advantages and strengths of the presented tracker.

2 Preprocessing

In the preprocessing stage of the algorithm the edge image I_C^e of the incoming camera image I_C is extracted and stored for comparison later. The color surface S of the object to track is *projected* into the incoming image I_C and the edges are calculated again. This edge image of the object I_S^e is then used to *re-project* to the geometry in world space which results in the corresponding edge map of the original surface S^e.

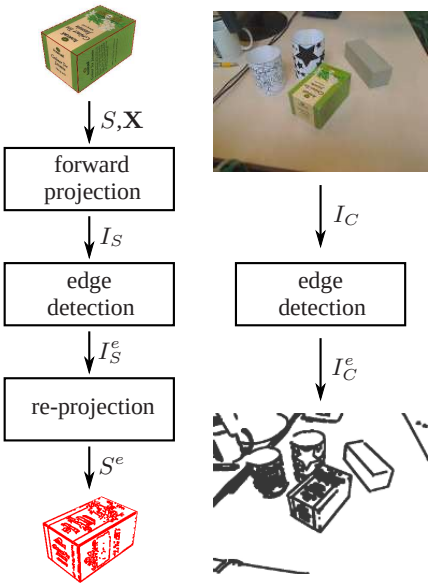


Figure 2: *Flow chart of preprocessing*

2.1 Edge detection

The edges I^e are found by convolving the original image I with a Gaussian smoothing $\mathbf{G}$ and the two Sobel kernels $\mathbf{H}_{s,x}$ and $\mathbf{H}_{s,y}$ as described in [Burger and Burge 2008].

$$I^e = \begin{pmatrix} I_x^e \\ I_y^e \end{pmatrix} = \begin{pmatrix} \mathbf{H}_x * \mathbf{G} * I \\ \mathbf{H}_y * \mathbf{G} * I \end{pmatrix} \quad (1)$$

Furthermore, the result is improved by applying thinning and spreading algorithms. Note that for the gradient calculation in Section 3.2 the x and y values are stored separately.

Figure 3 shows the different results of edge detection where the x- and y-components of the gradient are stored in the red and green color channel. The detection tolerance can be influenced by applying spreading, which broadens the edges by a specific number of pixels according to the tolerance level. This means that instead of searching for edge pixels close to each other, the line width of the edge is raised as shown in Figure 3, which broadens the matching area.

2.2 Forward projection

The 3D -model is projected into the camera image I_C as defined by Equation (2). Using the camera image I_C takes into account that edges are not visible when there are similar light and color conditions in the background. Then the edges of the image are extracted using Equation (1).

The transformation of the model from world space to image space is performed by the following matrix operations:

$$\begin{aligned} \mathbf{u}_S &= \mathbf{T}_p \mathbf{X} \mathbf{v}_S \\ I_S(u,v) &= \begin{cases} S(\mathbf{v}_S) & \text{if} \quad (u,v) \in \boldsymbol{\mathcal{U}} \\ I_C(u,v) & \text{else} \end{cases} \end{aligned} \quad (2)$$

where I_S is the camera image with the projected model. $\boldsymbol{\mathcal{U}}$ defines the geometry of the object in image space with

$$\mathbf{u}_S = [u_S, v_S] \in \boldsymbol{\mathcal{U}}$$

$\mathbf{T}_p$ denotes the projection- and $\mathbf{X}$ the model view or world transformation matrix which defines the pose of an object to track with a rotational and translational term $\mathbf{R}$ and $\mathbf{t}$.

$$\mathbf{X} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}$$

The geometry of the object in world space $\boldsymbol{\mathcal{V}}$ is represented by its vectors

$$\mathbf{v}_S = [x_S, y_S, z_S] \in \boldsymbol{\mathcal{V}}$$

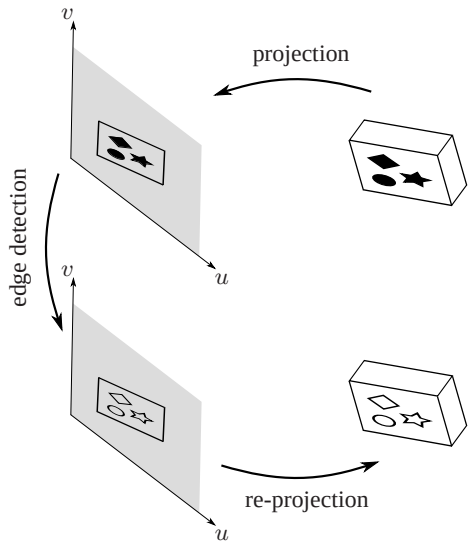


Figure 4: *Forward projection and re-projection*

2.3 Re-projection

The idea of re-projection is to replace the color surface of the 3D -model with the corresponding edge map. Note that it is not possible to do the image processing on the color surface S of the object directly, as the edge features get distorted and thinned out when they are projected to image space and therefore wrongly fail the edge matching test. Comparing the edges of the model with the camera image requires the same methods applied to the same point of view and also the same scaling of the edge width.

Using a particle filter requires drawing the model several hundred times at different poses $\mathbf{X}_i$ with $i = 1...N$. Replacing the surface

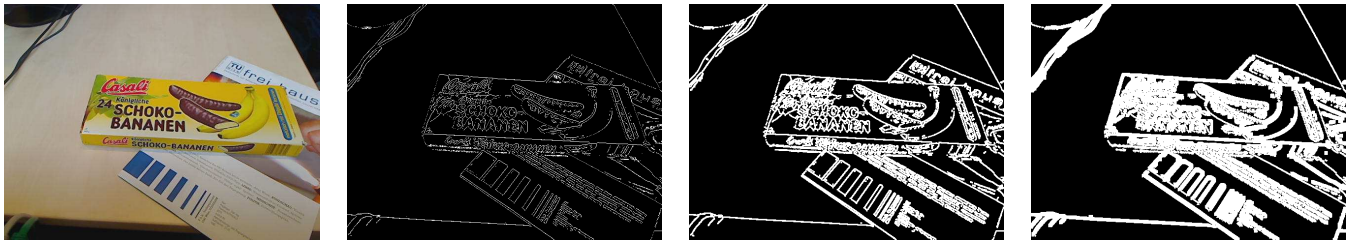


Figure 3: *Edge detection results from left to right: original, no spreading, one time spreading, three times spreading*

and running the edge detection algorithm for each particle would cause the tracker to be far away from real-time capability. For this reason, the surface of all particles is replaced by only one edge map I^e_S, calculated by using the prior tracking result $\mathbf{X}^+$. This, in principle causes the same problems as mentioned above, but assuming that the motion of the particles $\mathbf{X}_i$ remains small within one tracking pass, the distortions and thinning out can be disregarded.

3 Recursive particle filtering

For each tracking pass the recursive particle filtering executes the methods shown in Figure 5. First the particles $i = [1 \dots N]$, representing the pose of the object, are generated using Gaussian noise. Then the confidence level of each particle i is evaluated by matching it against the edge image of the camera I^e_C. If there is still processing time t_f remaining, then a further recursion step of particle generation and evaluation of the confidence levels is performed with different parameters as described in Section 3.4. Otherwise the maximum likely particle is passed to the next step. The linear Kalman filter, including a physical motion model, is attached to the outcome of the recursive particle filter to fine tune the result and remove remaining jitter. As this additional filter is not part of the recursion it is explained in Section 4.

The reason for this setup is to benefit from the robustness and speed of a particle filter. For higher accuracy, the standard deviation of the noise is reduced in each recursion. The linear Kalman filter is attached just for fine tuning as mentioned above.

3.1 Particle generation

The prior pose $\mathbf{X}^-_i$ of each particle is calculated by perturbing the posterior $\mathbf{X}^+$ with Gaussian noise $\mathbf{n}(\boldsymbol{\sigma}^2)$ with a standard deviation scaled by the prior confidence level of the pose w_m, a scaling factor for motion effect $\mathbf{f}_m$ and a scaling factor $\mathbf{f}_c$ set by each recursion step:

$$\begin{aligned} \mathbf{X}^-_i &= \mathbf{X}^+ + \mathbf{n}(\boldsymbol{\sigma}^2(w_m, \mathbf{f}_m, f_c)) \\ i &= [1 \dots N] \end{aligned} \tag{3}$$

The standard deviation is evaluated by

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}_I \mathbf{f}_m \mathbf{f}_c . (1 - w_m) \tag{4}$$

where the confidence level of the prior pose w_m is multiplied to the initial standard deviation $\boldsymbol{\sigma}_I$ so that the particle distribution narrows with higher confidence. The motion effect $\mathbf{f}_m$ takes into account that motion in world space along the camera viewing axis causes less change in image space then the same motion orthogonally to the viewing axis. $\mathbf{f}_c$ becomes smaller with each recursion step in the particle filter. $\boldsymbol{\sigma}_I$ is implemented as a parameter to be set by the user, but should be evaluated automatically in the future regarding the tracking conditions.

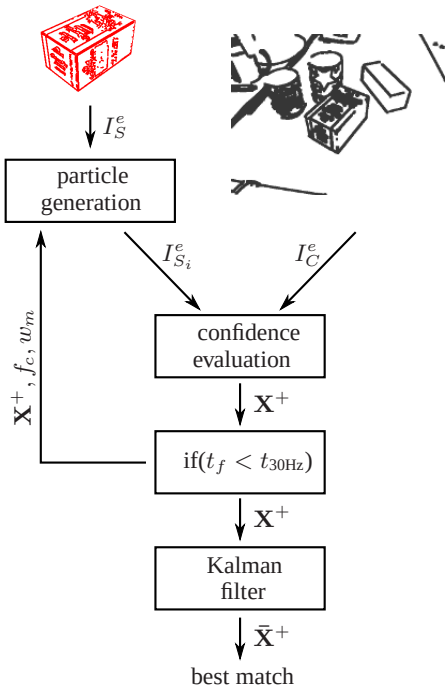


Figure 5: *Block scheme of motion with recursive particle filter and Kalman filter*

3.2 Confidence level

Each particle is tested against the camera image and a confidence level is calculated. Therefore the correlation between the gradients of the edges $\mathbf{g}_{S_i}(u,v)$ and $\mathbf{g}_C(u,v)$ is evaluated by comparing the direction of the edges at each image point (u,v).

$$\begin{aligned} \mathbf{g}_{S_i}(u,v) &= \begin{pmatrix} I^e_{S_i,x}(u,v) \\ I^e_{S_i,y}(u,v) \end{pmatrix} \\ \mathbf{g}_C(u,v) &= \begin{pmatrix} I^e_{C,x}(u,v) \\ I^e_{C,y}(u,v) \end{pmatrix} \end{aligned}$$

The angles between those vectors are calculated, producing the edge correlation image Φ_i:

$$\begin{aligned} \phi &= \arccos(\mathbf{g}_{S_i} . \mathbf{g}_C) \\ \Phi_i(u,v) &= \begin{cases} 1 - \frac{2\phi}{\pi} & \text{if} \quad \phi < \pi/2 \\ 1 - \frac{2|\phi - \pi|}{\pi} & \text{if} \quad \phi > \pi/2 \\ 0 & \text{if} \quad (u,v) \notin \mathbf{v}'_S \end{cases} \end{aligned} \tag{5}$$

Note that it is assumed that the result of the $\arccos()$ function lies within 0 and π. The image Φ_i now contains the degree of correlation between the pose suggested by the particle i and the camera

image. The angular deviation of the edge angles Φ_i is scaled to the range of 0 to 1.

The confidence level w_i is evaluated as follows:

$$w_i = \left(\frac{m_i}{n_i} + \frac{m_i}{n_{max}} \right) \frac{1}{W} \tag{6}$$

with

$$\begin{aligned} m_i &= \int_u \int_v \Phi_i(u,v) dv du \\ n_i &= \int_u \int_v |I^e_{S_i}(u,v)| dv du \\ W &= \sum_{i=1}^{N} w_i \\ n_{max} &\propto \max(n_i) \end{aligned}$$

The first term within the brackets is the percentage of matching edge pixels m_i with respect to the total visible edge pixels n_i. Calculating the confidence level only with this term would cause the tracker to lock when only one side of the 3D object is visible. When at this special pose the object is rotated slightly, another side of the object becomes visible. The particle matching this rotation would be equal or most often less then the prior front facing particle. The reason for this is that the number of matching pixels m_i grow less than the total visible pixels n_i when rotating the object out of the front side view. This effect amplifies when taking into account that edge detection for strongly tilted faces is very faulty.

The second term allocates more weight to the total number of matching pixels m_i which is intrinsically higher for the rotated particle. n_{max} which are the maximum visible edge pixels in the actual area scales the pixels to the proper range. As this summation would lead to confidence levels higher than 1, it is divided by the sum of confidence levels W.

This differs from the function used in [Klein and Murray 2006]

$$\text{Likelihood}\left(\mathbf{X}_i^-\right) \propto \exp\left(k \frac{d_i}{v_i}\right)$$

which we experienced to lock very fast at the local minima mentioned above. Here d_i denotes the number of matching edge pixel, v_i the total edge pixels of the wireframe model, where the hidden edges are removed and k is a constant for distributing the likelihood.

3.3 Determining the pose

As explained in the sections 2.2 and 2.3 for projection and reprojection of the model, a single pose $\mathbf{X}^+$ has to be defined. This is where the approach suggested in this paper differs from usual, straight forward particle filters, where the whole propability density function

$$\begin{aligned} w_i &= P(\mathbf{X}_i^-) \\ i &= [1 \ldots N] \end{aligned}$$

of the previous timestep, is carried over into the next estimation step.

The pose $\mathbf{X}^+$ is evaluated using the mean of the top most likely particles M. $\mathbf{X}_k^-$ in Equation (7) denotes the particles sorted by the confidence level w_k in descending order.

$$\mathbf{X}^+ = \frac{1}{w_m} \sum_{k=1}^{M} \mathbf{X}_k^- . w_k \tag{7}$$

with

$$w_m = \frac{1}{M} \sum_{k=1}^{M} w_k$$

Experiments have shown that increasing the number of most likely particles M to consider in the mean confidence, while lowering the standard deviation in Equation (4) by f_c and the edge width (see Figure 3) for each further recursion obtains good results.

3.4 Recursion

The methods described in Section 3.1 and 3.2 are performed for each of the hundreds of particles. The idea of recursion is to take advantage of the information gain when calculating. This means that the N particles are divided into subranges $R_1, R_2, R_3, \ldots$ and so forth. For every range R_k, the pose estimate $\mathbf{X}^+$ and confidence level w_m of the previous particle filtering R_{k-1} is used. The standard deviation for the particle generation is reduced by the scaling factor $\mathbf{f}_c$, which narrows the *search area* of the filter. Therefore Equation (3) becomes

$$\begin{aligned} \mathbf{X}_i^- (R_k) &= \mathbf{X}^+ (R_{k-1}) + \mathbf{n}\left(\boldsymbol{\sigma}^2\right) \\ i &= [1 \ldots N_k] \end{aligned}$$

with

$$\boldsymbol{\sigma} = \boldsymbol{\sigma_I} \mathbf{f}_m f_c(R_{k-1}).(1 - w_m(R_{k-1}))$$

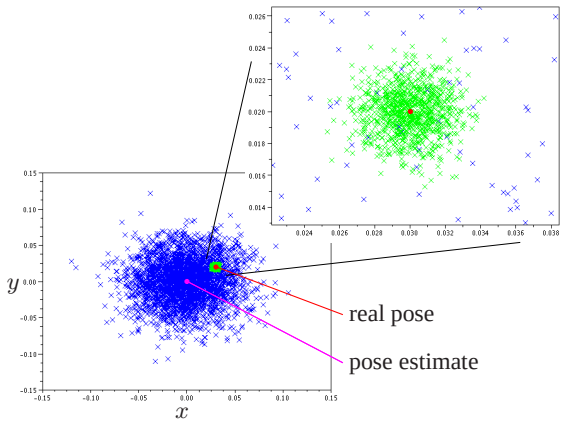


Figure 6: *Recursive particle filter*

Figure 6 shows the principal idea of recursive particle filtering in 2D. In the lower left graph the particles are perturbed using a high standard deviation of the Gaussian noise of the particles, covering a large area around the prior pose estimate. The mean of the top most likely particles is used to evaluate the rough pose of the real object. The upper left graph shows particles with lower standard deviation, where this time the most likely particles measure the real pose much more accurately.

The example in Figure 6 is drawn with 1500 particles with high and 500 with low standard deviation. This method allows the tracker to respond both quickly and accurately without increasing the tracking time, because it does not require any more particles than before.

4 Linear Kalman filtering

The discrete Kalman filter implemented for this approach uses a motion model for all six degrees of freedom of the object. The

reason why the motion model is not applied in the particle filter is because this would reduce the speed and accuracy of the tracker, as modelling the velocity of the object would rise the degree of freedom from 6 to 12. This would mean that the particles have to cover 6 more dimensions. Therefore the Kalman filter is attached only to smooth the resulting trajectory and remove remaining jitter.

Time Update - "Predict"

$$\begin{aligned} \mathbf{x}_k^- &= \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_k \\ \mathbf{P}_k^- &= \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q} \end{aligned}$$

Measurement Update - "Correct"

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_k^-\mathbf{H}^T\left(\mathbf{H}\mathbf{P}_k^-\mathbf{H}^T + \mathbf{R}(w_m)\right)^{-1} \\ \mathbf{x}_k &= \mathbf{x}_k^- + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{H}\mathbf{x}_k^-\right) \\ \mathbf{P}_k &= \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}\right)\mathbf{P}_k^- \end{aligned} \tag{8}$$

where

$$\mathbf{x}_k = \left[x_k, y_k, z_k, \alpha_k, \beta_k, \gamma_k, \dot{x}_k, \dot{y}_k, \dot{z}_k, \dot{\alpha}_k, \dot{\beta}_k, \dot{\gamma}_k\right]$$

denotes the state of the Kalman filter containing the pose and velocity of the six degrees of freedom.

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} & \text{diag}(\Delta t) \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$

is the *state matrix*,

$$\mathbf{B} = [\text{diag}(0)]$$

the *input matrix*,

$$\mathbf{H} = [\mathbf{I}, \text{diag}(0)]$$

the *output matrix*, $\mathbf{P}_k$ the *estimate error covariance*, $\mathbf{Q}$ the *process noise covariance*, $\mathbf{R}$ the *measurement noise covariance*, $\mathbf{I}$ the unity matrix and $\mathbf{K}_k$ the *Kalman gain*. Please refer to [Welch and Bishop 2004] for more details on Kalman filtering.

The process covariance matrix $\mathbf{Q}$ defines the noise of the physical model, where no information is available. However, in contrast to $\mathbf{R}$, this matrix is set to a fixed value.

The measurement covariance matrix $\mathbf{R}$ depends on the confidence level of the pose as follows:

$$\mathbf{R}(w_m) = \begin{bmatrix} \text{diag}((w_{t-3})^3) & 0 \\ 0 & \text{diag}((w_{t-3})^3) \end{bmatrix}$$

As shown in Equation (8), this means that $\mathbf{R}$ rises proportionally to the delayed confidence level of the measurement from the particle filter. Therefore, the Kalman gain drops, giving less weight to the measurement $\mathbf{z}_k$ and more weight to the motion model, which smooths the result. This means that jitter is removed when the confidence is high, for example when the object to track does not move.

On the other hand, the lag behind the real object, caused by the motion model during acceleration, is decreased when the confidence falls, which usually happens when the object to track moves fast. In this case, the Kalman gain increases the weight of the measurement, increasing the speed but also allowing jitter which is barely visible when the object is moving anyway.

Experimants have shown that a cubic function yields to a smooth tracking behaviour. The delay of the confidence level w_{t-3} removes overshooting of the motion model when the object suddenly stops moving.

5 Implementation notes

The implementation of the algorithm requires discretization which is denoted by bold letters for images in this section.

5.1 Notes for preprocessing

The preprocessing is done once per tracking pass and is not as time critical as the recursive particle filtering in the following section. However, the overall performance of the tracker needs to be as fast as possible, so this part is also implemented using the graphics processing unit. Therefore, the image received by the camera is sent to the graphics board where it is stored as texture. The convolution with the Gaussian and Sobel kernels are applied by shaders as well as thinning and spreading. The edge image is again stored as *RGB* texture where the *R*- and *G*-channels are used for the x- and y-components of the image gradient.

As the channels only allow values between 0 and 1, the normalized gradients ranging from -1 to 1 need to be adjusted.

The surface of the object to track is made up of vertices which form triangles and quadrangles, also called primitives. The projections described in Equation (2) and (9) are performed for these vertices only, because the surface points within a primitive can be determined by linear interpolation, which is optimized by the graphics adapter.

$\mathbf{T}_p$ and $\mathbf{X}$ denote the projection and model view matrix, which can be queried from OpenGL. The six degrees of freedom are stored in the vector

$$\mathbf{x}_i = [x_i, y_i, z_i, \alpha_i, \beta_i, \gamma_i]$$

Instead of transforming the edge map I_S^e back to world space by solving Equation (2) with respect to S, the coordinates for the edge map in image space are evaluated. This has to be done only once, since those coordinates do not change for the other particles.

$$\begin{aligned} \mathbf{u}_S^+ &= \mathbf{T}_p\mathbf{X}^+\mathbf{v}_S \\ I_{S_i}^e(u,v) &= I_S^e(\mathbf{u}_S^+) \end{aligned} \tag{9}$$

The particles i are represented by the pose matrices $\mathbf{X}_i^-$. The vectors $\mathbf{u}_{S_i}$ that are used to find the corresponding point in the camera image I_C^e are calculated with

$$\begin{aligned} \mathbf{u}_{S_i} &= \mathbf{T}_p\mathbf{X}_i^-\mathbf{v}_S \\ I_{C_i}^e(u,v) &= I_C^e(\mathbf{u}_{S_i}) \end{aligned}$$

At this point for each particle i the model $I_{S_i}^e$ and camera image $I_{C_i}^e$ are ready for comparison which is described in Section 3.2.

5.2 Notes for recursive particle filtering

The calculations described in this section are very critical with respect to optimized programming, since every single line of code is called several hundreds of times. In particular evaluating the confidence level using the *NVIDIA Occlusion Query* is definitely a bottle-neck in the algorithm. This OpenGL extension is responsible for counting the pixels of the whole edge map. The matching pixels m_i and total edge pixels n_i are evaluated by summing up the correlation image $\mathbf{\Phi}_i$ and the edge map $\mathbf{I}_{S_i}^e$:

$$\begin{aligned} m_i &= \sum_{u,v} \mathbf{\Phi}_i(u,v) \\ n_i &= \sum_{u,v} \mathbf{I}_{S_i}^e(u,v) \end{aligned}$$

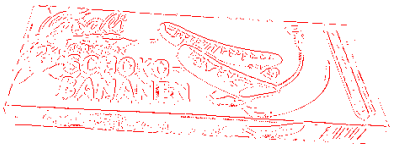
As this extension only supports counting pixels disregarding the value, they only can be marked to be rendered or not. Therefore, the displacement image is thresholded by the angle ϵ:

$$\begin{aligned} \phi &= \arccos\left(\mathbf{g}_{S_i}(u,v).\mathbf{g}_C(u,v)\right) \\ \mathbf{\Phi}_i &= \begin{cases} 0 & \text{if} \quad \phi < \epsilon \\ 1 & \text{if} \quad \phi \geq \epsilon \end{cases} \end{aligned} \tag{10}$$

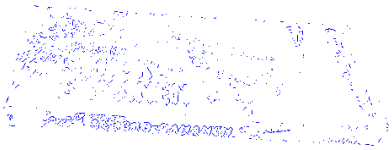
where ϵ denotes the angular threshold. For the match evaluation shader 0 means that the pixel is discarded when rendering and does not increase the counter, whereas 1 means the pixel is drawn and therefore increases m_i.

Figure 7b) shows the pixels m_i successfully passing the match evaluation. 7c) are the pixels which fail the match test of Equation (10) and 7d) is the total number of pixels n_i of the edge image of the object. Note the missing pixels on the very upper edge, as a result of the background having the same color (*yellow*) as the object. The mismatch of the edges on the front side of the box is caused by the inaccuracy of the placement of the texture on the geometry of the model which is produced manually by a 3D modeling tool.

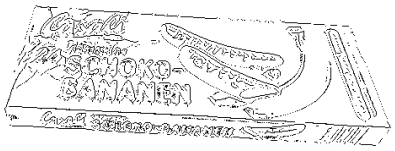
a) Object to track with cluttered background



b) Matched edges m_i



c) Mismatched edges



d) Total edges of model n_i

Figure 7: *Edge matching*

6 Results

As the tracker requires fast parallel calculations, the focus of the system is on the graphics board where it is implemented. It has been tested on an NVIDIA GeForce GTX 285 with a fill rate of 50 billion pixels per second, and an Intel Core2 Quad CPU Q6600 with 2.4 GHz. To fulfill a minimum frame rate of 30 FPS (frames per second), the time for one tracking pass is 33 ms. Within this limits it is possible to draw 1500 particles for a box (6 faces textured with a 900x730 pixel image) like in Figure 7a). With the cylinder model shown in Figure 1 with 16 faces and a 900x400 pixel texture image, the tracker achieves 580 particles within the same time and 100 particles with the cylinder with 64 faces.

Figure 8 shows some results of a video sequence. The first row demonstrates the robustness of the tracker. The whole top surface and the edges of the geometry are covered by the hand and there are reflections of the checkerboard pattern on the front face and a cluttered background, but the pose of the box still can be estimated. Of course at some degree of occlusion, the accuracy of the pose drops until it cannot be determined at all.

The second row shows the fast and reliable convergence. Although the deviation of the estimated from the actual pose is very high, the tracker finds the correct alignment within a second. The third row illustrates the concept of recursive particle filtering. Especially in the second image from the left, where the speed of the moving object is high, the benefit is clearly visible (compare with Figure 6).

7 Conclusion

We presented a method for fast and robust object tracking. It converges fast to the correct pose and is able to handle relative large deviations, for example when initializing. Partial occlusion, reflections, light changes, shadows and cluttered background are no problem for the tracker, as long as enough features are visible to determine the pose. Exploiting the power of a graphics processing unit with a particle filter in a recursive design allows high tracking speed with sufficient accuracy.

However, there are several improvements possible. Firstly the mismatch of the 3D -model to the real object, as described in Section 5.2, can be learned and corrected with constraints that prevent the model from strong distortion. Secondly corner detection, color matching and so forth can be implemented which would most likely further improve the accuracy and robustness of the tracker.

The correlation map Φ_i in Equation (5) is simplified to Equation (10), because the *NVIDIA Occlusion Query* only counts visible pixels disregarding the information about the angular displacement stored within. A future work would be to implement precise evaluation of the confidence level as described in Equations (5) and (6).

The bottle-neck, with respect to the frame time of this approach, is definitely the particle filter with its evaluation of the confidence level using the OpenGL extension. The tracking error e directly correlates with the standard deviation σ and number of particles N as follows:

$$e \propto \frac{\sigma}{N}$$

with

$$N \propto t = t_{33ms}$$

This means that the tracking error e can be significantly reduced by lowering the standard deviation σ for each degree of freedom independently. When for example the z position of an object to track is known, because it lies on a table the accuracy can be increased by lowering the standard deviation for this degree of freedom.

There are several points of the algorithm where further investigation needs to be done, like finding the optimal boundary conditions and functions for the recursive particle filter and designing a better function for particle generation. Or more specifically, designing better functions for calculating the standard deviation of the Gaussian noise in Equation (4).

A further problem that needs to be solved is, that the tracker cannot supply information if tracking fails when it locks into a local min-

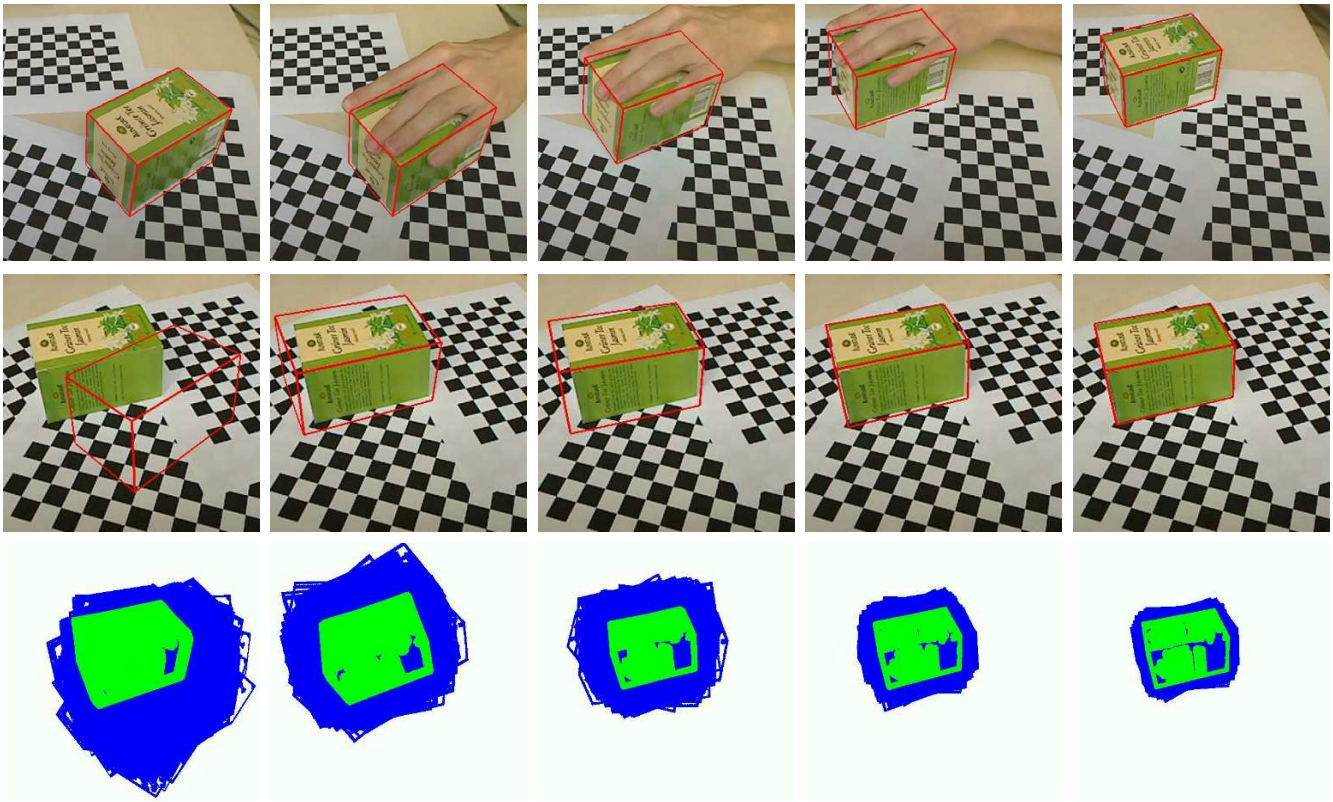


Figure 8: *First row: robustness against occlusion, reflections and background clutter; Second row: fast and robust convergence; Third row: particle distribution with three recursions*

imum. There, the confidence evaluation returns sometimes values as high as at the correct tracking pose.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement No. 215181, CogX.

References

A. RUF, M. TONKO, R. H., AND NAGEL, H.-H. 1997. Visual tracking by adaptive kinematic prediction. *Proceedings of International Conference on Intelligent Robots and Systems*.

BURGER, W., AND BURGE, M. J. 2008. *Digital Image Processing, An Algorithmic Introduction Using Java*. Springer.

D. KOLLER, K. D., AND NAGEL, H.-H. 1993. Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*.

DRUMMOND, T., AND CIPOLLA, R. 1999. Real-time tracking of complex structures with on-line camera calibration. 574–583.

GENNERY, D. 1992. Visual tracking of known three-dimensional object. *International Journal of Computer Vision*.

HARRIS, C. 1992. Tracking with rigid objects. *MIT Press*.

KESSENICH, J. 2008. *The OpenGL Shading Language, Version 1.30*.

KLEIN, G., AND DRUMMOND, T. 2003. Robust visual tracking for non-instrumented augmented reality.

KLEIN, G., AND MURRAY, D. 2006. Full-3d edge tracking with a particle filter. *British Machine Vision Conference Proc 17th*.

KLEIN, G., AND MURRAY, D. 2007. Parallel tracking and mapping for small ar workspaces. *Proc International Symposium on Mixed and Augmented Reality (ISMAR)*.

KOSAKA, A., AND NAKAZAWA, G. 1995. Vision-based motion tracking of rigid objects using prediction of uncertainties. *International Conference on Robotics and Automation*.

L. VACCHETTI, V. L., AND FUA, P. 2004. Stable real-time 3d tracking using online and offline information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

LOWE, D. G. 1992. Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*.

LUCA VACCHETTI, V. L., AND FUA, P. 2004. Combining edge and texture information for real-time accurate 3d camera tracking.

LUCIE MASSON, M. D., AND JURIE, F. 2004. Robust real time tracking of 3d objects.

M. VINCZE, M. AYROMLOU, W. P., AND ZILLICH, M. 2001. Edge-projected integration of image and model cues for robust model-based object tracking. *The International Journal of Robotics Research*.

MUSTAFA ÖZUYSAL, MICHAEL CALONDER, V. L., AND FUA, P. 2009. Fast keypoint recognition using random ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

P.A. SMITH, I. R., AND DAVISON, A. 2006. Real-time monocular slam with straight lines. *Proc 17th British Machine Vision Conference* (sept).

PHILIPP MICHEL, J. C. E. A. 2008. Gpu-accelerated real-time 3d tracking for humanoid autonomy.

R. KOCH, K. KOESER, B. S., AND EVERS-SENNE, J.-F. 2005. Markerless image-based 3d tracking for real-time augmented reality applications. *WIAMIS* (april).

ROST, R. J. 2006. *OpenGL Shading Language*, vol. Second Edition. Addison-Wesley.

SEGAL, M., AND AKELEY, K. 2008. *The OpenGL Graphics System: A Specification, Version 3.0*.

SUN-KYOO HWANG, M. B., AND KIM, W.-Y. 2008. Local discriptor by zernike moments for real-time keypoint matching. *IEEE Congress on Image and Signal Processing*.

VINCENT LEPETIT, JULIEN PILET, P. F. 2004. Point matching as a classifiaction problem for fast and robutst object pose estimation. *Conference on Computer Vision and Pattern Recognition* (june).

WELCH, G., AND BISHOP, G. 2004. An introduction to the kalman filter.

Классификация теней с помощью деревьев решений в задаче мониторинга транспортного потока

Илья Попов, Александр Бовырин
Компания Интелливижн,
Нижний Новгород, Россия,
{ilyap, alexb} @ intelli-vision.com

Abstract

This work aimed to shadow classification problem. Due to increasing traffic jams number automatic road traffic monitoring is becoming important application of computer vision. It is known that shadows of vehicles can dramatically decrease accuracy of monitoring. We developed shadow recognition algorithm which significantly decrease influence of shadows. This algorithm is based on extraction of specialized set of features and CART method usage for further classification. This shadow detection method has low computational load and this has been developed to be ported to DSP architectures.

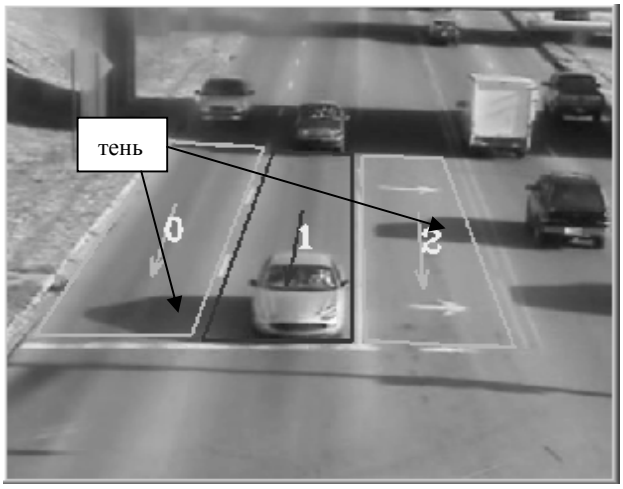
Аннотация

Данная работа посвящена проблеме распознавания теней автомобилей при анализе транспортного потока. Автоматический учёт проезжающих автомобилей по видео данным является важным приложением компьютерного зрения. Известно, что тени от транспортных средств негативно влияют на корректность полученной статистики. В статье предложен алгоритм распознавания теней, который позволяет значительно уменьшить влияние теней на работу системы. Этот алгоритм основан на оригинальном наборе признаков и применении метода CART для дальнейшей классификации. Особенностью алгоритма является то, что он не требует больших вычислительных ресурсов и может быть реализован на DSP-архитектурах.

Ключевые слова: *классификация теней, анализ транспортного потока, деревья решений.*

1. ВВЕДЕНИЕ

В настоящее время системы автоматического мониторинга загруженности дорог особенно актуальны в связи с всё увеличивающимся транспортным потоком. Было предложено много работ по автоматическому подсчёту транспортных средств (ТС) на полосе движения, оценки их средней скорости и их классификации [1], [2]. Большинство авторов [2]-[4] отмечают, что одной из основных проблем влияющих на точность анализа является наличие теней от проезжающих автомобилей на соседних полосах движения, которые и по форме и по движению похожи на автомобили и они, таким образом, генерируют ложные срабатывания детектора автомобилей, привнося серьёзные ошибки в общую статистику.
На изображении 1 представлен типичный случай тени на соседней полосе движения.



Изображение 1:
типичный случай тени на соседней полосе движения.

Во многих работах проблема теней решается с помощью классификации каждого пикселя на изображении, учитывая факт, что в теневой области оттенок цвета пикселя фона не сильно меняется, а интенсивность уменьшается. Обзор таких методов можно найти в работе [4]. Однако попиксельная классификация недостаточно хорошо работает на чёрно-белых камерах видеонаблюдения и часто “вырезает” большие тёмные области на ТС, что может привести к ошибкам детектирования.
В работе [3] задача распознавания теней решалась с помощью использования набора вейвлетов и применения обученной нейронной сети в качестве классификатора.

Следует отметить, что нейронные сети не позволяют прямо оценить, какие переменные наиболее важны для распознавания и таким образом для решения задачи требуется вычислять все признаки.

Авторы работы [10] предложили использовать ребра изображения как вспомогательную информацию для точного отсечения тени от ТС. Однако применяемые в этой работе методики (детектор рёбер Канни, преобразование Хафа) требуют значительных вычислительных ресурсов.

Поскольку использование систем автоматического мониторинга транспортного потока чаще всего подразумевает их портирование на архитектуру DSP [5], то важно экономить вычислительные ресурсы и вычислять только те признаки, которые наиболее значимы для распознавания.

Для этих целей в данной работе предложено использовать метод CART (classification and regression tree) предложенный

в [6]. CART позволяет оценить важность признаков участвующих в распознавании.

В работе также предложен оригинальный набор признаков, по которым производится построение дерева решений и дальнейшее распознавание.

В данной работе не представлены описания используемых нами методов обучения фона, методов извлечения движущихся объектов на видео и методов слежения за ТС. Примеры таких методов можно найти в [7], [1].

2. ИЗВЛЕЧЕНИЕ ВЕКТОРА ПРИЗНАКОВ

В рамках поставленной задачи было введено понятие полосы движения – четырехугольная область, заданная на интересующей пользователя части дороги (см. изображение 1).

Для обеспечения возможности детекции и распознавания объектов для каждой полосы была реализована модель фона (BG). Эта модель в дальнейшем использовалась для выделения области (FG), в которой находится появившийся объект – транспортное средство (ТС), тень, пешеход и т.п.

В рамках работы было рассмотрено более 20 признаков, из которых в процессе отбора с использованием деревьев принятия решений рассмотренных ниже в финальной версии алгоритма оставлено 10 наиболее значимых. Большинство из них в качестве исходных данных используют информацию об области FG. Отбор признаков проводился, как по критерию качества классификации тени, так и по критерию трудоемкости вычислений. Причем, трудоемкость вычислений отчасти имела превалирующую роль в связи с тем, что алгоритм распознавания теней должен был работать в составе основного программного комплекса на одном вычислительном устройстве одновременно для нескольких потоков видеоданных в условиях значительных ограничений по объему доступной памяти и производительности процессора (например, процессоры серии TI DSP C62xx, 8MB RAM, 300 MHz).

Из наиболее интересных среди значимых признаков можно выделить:

1) Число угловых точек [8] внутри области тени. Поскольку изображение ТС контрастнее покрытия дорожной полосы, то обычно число угловых точек внутри теневой области значительно меньше, чем в области ТС.

$$nFPCount = \sum_{p \in FG} 1,$$

где FG - множество внутренних пикселов маски объекта, p – угловая точка с координатами (x, y).

2) Площадь не фоновых пикселей обнаруженных на полосе движения;

$S = \sum_{c=1}^{w} \sum_{r=1}^{h} M(c,r)$, где M(c,r) – маска FG, содержащая значение 1 в области объекта, иначе – 0, w – число столбцов в изображении, h – число строк в изображении.

3) Средняя магнитуда градиентов в области объекта. Тень дает минимум новых градиентов, причем в основном на ее краях;

$$nEdgesXY = \frac{\sum_{p \in FG} \sqrt{(I_x^2(p_i) + I_y^2(p_i))}}{S},$$

где Ix_i и Iy_i - значения градиентов в точке p_i, S – площадь объекта определенная, как признак (2)

4) Среднеквадратическое отклонение значений интенсивности в области объекта.

$$nStd = \sqrt{\frac{\sum_{p_i \in FG} (I(p_i) - I_{avg})^2}{S - 1}},$$

где I_{avg} – среднее значение интенсивности для множества пикселов определенных FG, S – определена признаком (2) . Тень должна иметь меньшее значение nStd, чем ТС.

5) Энтропия значений интенсивности в области объекта определяется как:

$$E = -\sum_{i=0}^{255} w_i \log(w_i), w_i = \frac{Hist(i)}{S},$$

где Hist[i], i = 0..255 – вектор значений хранящий гистограмму распределения значений интенсивности в области FG, S – определена признаком (2). В теневой области признак должен иметь малое значение.

6) Процент числа пикселов от общего числа пикселов объекта, у которых значение интенсивности уменьшилось по сравнению с данными модели фона;

$$nOffDown = \frac{\sum_{c=1}^{w} \sum_{r=1}^{h} M(c,r) \cdot f(c,r)}{S},$$

$$f(c,r) = \begin{cases} 1, & p_{c,r} < b_{c,r} \\ 0, & p_{c,r} >= b_{c,r} \end{cases}$$

где $p_{c,r} \in P$ – исходное изображение, $b_{c,r} \in BG$ –изображение модели BG, параметры w, h, M(c,r), S – определены в признаке (2).

7) Средняя интенсивность изображения фона в области, где маска FG содержит ненулевые значения.

В области тени средняя интенсивность должна быть меньшей, чем в той же области на изображении фона.

$$nBGAvgInt = \frac{\sum_{p \in FG} BG(p)}{S}$$

где BG – изображение фона, S – значение площади не фоновых пикселов, p – точка определяющая координаты пикселов, где маска FG принимает ненулевые значения.

8) Средняя интенсивность пикселов изображения фона.

$$nBGOnlyAvgInt = \frac{\sum_{p \in BG} BG(p)}{N},$$

где BG – изображение фона, p – точка принадлежащая полосе, N – число пикселов на анализируемой полосе движения.

Правила классификации теней могут меняться, когда интенсивность пикселов фона мала. Например, видеоданные соответствуют позднему вечеру, ночи и т.п.

Классификация обнаруженного объекта на классы: тень/не тень само по себе является нетривиальной задачей. В данном случае задача дополнительно осложнялось требованием минимизации ошибок распознавания ТС как тени, так как в данном случае алгоритм переставал детектировать ТС, что значительно ухудшало точностные характеристики программного комплекса в целом. В качестве основного подхода примененного на начальном этапе решения данной задачи была сформирована база данных значений признаков с целью определения их статистических характеристик, таких как дисперсия, средние и медианные значения. Данный подход дал положительные результаты только на начальном этапе разработки алгоритма, когда в наличии было не более 2-3 часов видео с примерами теней. Когда объем тестовых видеоданных превысил 40 часов статистические данные о распределении признаков теней и ТС стали иметь значительное пересечение, что привело к проблемам с фильтрацией теней и ложным распознаванием ТС, как тени. Таким образом, независимый учет признаков по отдельности не смог обеспечить качественного решения задачи.

3. ИСПОЛЬЗОВАНИЕ ДЕРЕВЬЕВ РЕШЕНИЙ

Для построения решающих правил принятия решения о классификации объекта на тень или ТС мы использовали алгоритм построения дерева решений CART (Classification And Regression Tree) предложенный в работе [6].

Результатом работы CART является бинарное дерево решений. Каждая вершина этого дерева представляет собой правило классификации ассоциированное с одной переменной, которое наилучшим образом разделяет тренировочное подмножество данных на два класса.

Существуют различные критерии выбора наилучшей переменной при формировании вершины дерева.

Нами было опробовано два критерия:

1. Энтропийный критерий

$$I_E(i) = -\sum_{j=1}^{m} f(i,j) \log_2 f(i,j)$$

2. Gini-критерий

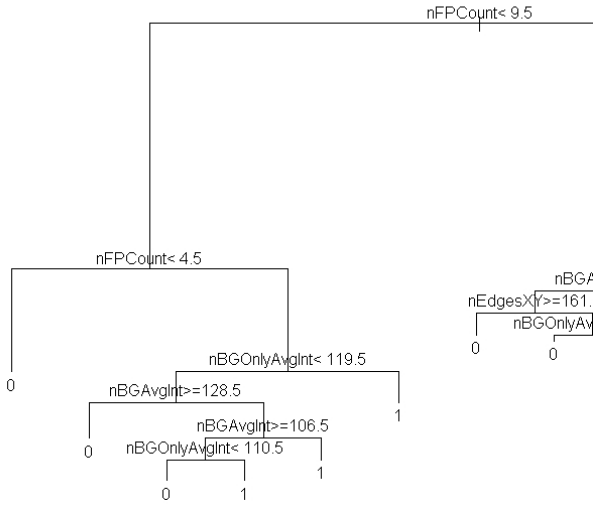
$$I_G(i) = 1 - \sum_{j=1}^{m} f(i,j)^2$$

где m – число классов (в нашем случае m=2), *f(i,j)* - частота класса j в вершине i.

В ходе построения дерева решений последовательно выбираются переменные минимизирующие один из этих критериев в производных вершинах. Например, если есть переменная i, которая идеально разбивает множество на классы, то $I_G = 0$ в производных вершинах.

Эксперименты показали, что для нашей задачи наилучший результат даёт тренировка дерева с помощью энтропийного критерия. При этом было замечено, что увеличение глубины дерева решений, начиная с шести, мало влияет на качество распознавания (например, использование дерева с глубиной семь показало уменьшение ошибки распознавания менее чем на один процент), поэтому глубина финального дерева была выбрана равной 6.

В рамках формата данной статьи затруднительно привести изображение всего дерева решений, поэтому мы приводим лишь фрагмент.



Изображение 2: фрагмент дерева решений (метка 0 соответствует классу “тень”, 1 –“ТС”).

Значительным достоинством применения CART является то, что в качестве результата работы формируется «прозрачное» для восприятия и анализа дерево решений, построенное с использованием заранее подготовленного обучающего множества, причем решающее задачу с заданными точностными характеристиками. В качестве набора данных для обучения использовались признаки, полученные по примерно 8 часам видеоданных.

В ходе построения дерева решений оказалось, что некоторые вычислительно тяжёлые признаки такие как, например, энтропия расположены далеко от корня дерева и мало влияют на результат распознавания. Поэтому эти признаки могут быть опущены при использовании метода на DSP архитектурах без значимого ухудшения результатов распознавания.

Следует отметить ещё один аспект, почему из множества алгоритмов классификации (машина опорных векторов, бустинг, нейронные сети,... [9]) был выбран именно алгоритм CART. Дело в том, что при классификации с помощью дерева решений нет необходимости вычислять все выбранные

признаки, поскольку часто путь от корня до листа дерева содержит меньшее число признаков.

4. РЕЗУЛЬТАТЫ

Тренировка дерева была осуществлена на тренировочной базе состоящей из 1194573 векторов признаков соответствующих случаем тени на полосе движения и 2082016 примерам соответствующих наличию транспортных средств на полосе.

Тестирование метода осуществлялось на тестовой базе состоящей из 4300462 позитивных примеров (тени) и 7911660 негативных примеров (ТС). Тестовая база не включала примеры из тренировочной базы.

Применение описанного выше метода распознавания теней дало следующие результаты:

- 79% правильно распознанных теней.
- 0.8% ТС были распознаны как тень

Важно, что процент ТС распознанных как тень не велик, поскольку в реальных условиях применения системы лучше классифицировать тень как машину, чем пропустить машину. Для того чтобы получить требуемые точностные характеристики распознавания можно варьировать параметры функции потерь при построении дерева (потери при ошибки классификации ТС должны быть больше, чем потери при неправильной классификации тени).

Отметим, что с целью дальнейшего уменьшения ошибок классификации проводилась временная фильтрация результатов классификаций.

5. ЗАКЛЮЧЕНИЕ

Применение предложенного набора признаков и метода CART для распознавания теней ТС на соседних полосах движения показало существенное уменьшения случаев ложных срабатываний детектора автомобилей на тестовом наборе данных. С помощью этого метода удалось достичь уровня ложных срабатываний равного 0.8%, что является приемлемым уровнем для применения в коммерческих системах автоматического мониторинга трафика.

Следует отметить, что классификация с помощью CART и предложенного набора признаков не требует больших вычислительных ресурсов и может быть реализована на DSP, что практически значимо для систем подобного рода [5].

6. ЛИТЕРАТУРА

[1] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati,. “Statistical and knowledge-based moving object detection in traffic scene", in Proceedings of IEEE Int'l Conference on Intelligent Transportation Systems, Oct. 2000, pp. 27-32.

[2] Yoneyama, A., Yeh, C.H., Kuo, C.C.J., “Robust Vehicle and Traffic Information Extraction for Highway Surveillance”, JASP(2005), No. 14, 2005, pp. 2305-2321.

[3] Chao, T, Lau, B, Park, Y "Vehicle Detection and Classification in Shadowy Traffic Images Using Wavelets and Neural Networks", Transportation Sensors and Controls: Collision Avoidance, Traffic Management, 1996, SPIE Proc. Vol 2902, pp 136-147.

[4] A. Prati, I. Mikic, C. Grana, and M. M. Trivedi, “Shadow detection algorithm for traffic flow analysis: a comparative study,” in Proc. IEEE Conference on Intelligent Transportation System (ITSC ’01), pp. 340–345, Oakland, Calif, USA, August 2001.

[5] Michael Bramberger, Roman P. Pflugfelder, Arnold Maier, Bernhard Rinner, Bernhard Strobl, Helmut Schwabach “A Smart Camera for Traffic Surveillance”, In Proceedings of the First Workshop on Intelligent Solutions in Embedded Systems, 2003

[6] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification* and Regression trees. Wadsworth Inc., 1984.

[7] C. Stauffer andW. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” in Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’99), vol. 2, pp. 246–252, Fort Collins, Colo, USA, June 1999.

[8] C. Harris and M.J. Stephens. A combined corner and edge detector. In Alvey Vision Conference, pages 147–152, 1988.

[9] Richard O. Duda, Peter E. Hart, and David G. Stork (2001). *Pattern Classification*. Wiley.

[10] Angie W. K. So, Kwan-Yee Kenneth Wong, Ronald H. Y. Chung, Francis Y. L. Chin: Shadow detection for vehicles by locating the object-shadow boundary. Signal and Image Processing, Proceedings of the IASTED International Conference, 2005.

Compensation of illumination inhomogeneities in multi-resolution image acquisition in confocal microscopy

Zoltan Tomori[1,2], Martin Capek[3,4], Jan Michalek[3], Lucie Kubinova[3]
[1] Institute of Experimental Physics, Slovak Academy of Science, Kosice, Slovakia
[2] Dept. of Cybernetics and Artificial Intelligence, FEI, Technical University Kosice, Slovakia
[3] Dept. of Biomathematics, Institute of Physiology, Academy of Sciences of the Czech Republic, Prague, Czech Republic
[4] Czech Technical University in Prague, Faculty of Biomedical Engineering, Kladno, Czech Republic
tomori@saske.sk, {capek, michalek, kubinova}@biomed.cas.cz

Abstract

A multi-resolution application in modern confocal microscopy is based on a set of objectives with different resolution and motorized XY stage. An image captured through a low-resolution objective serves as navigation map where the user points to positions or areas of interest in which a high-resolution image will be captured. This allows e.g. automated acquisition in time lapse experiments, generating random, or systematic random positions for stereological counting frames, image tiling etc. However, captured image series representing a volume of interest suffers from illumination inhomogeneities.

Algorithms described in this paper compensate both uneven illumination of captured images and light attenuation with depth. The first problem was solved by using modified morphological operation called the upper Lipschitz cover. To compensate the light attenuation with depth, a novel, very fast algorithm based on matching of distribution functions has been proposed.

Keywords: *Confocal microscopy, Light attenuation, Distribution function matching*

1. INTRODUCTION

A great number of image processing algorithms are based on a hierarchical top-down principle where the knowledge obtained in the higher (coarse) level is exploited in the lower (fine) level. A cooperation between the levels results in a significant increase of efficiency. Specific data structures (pyramids, quadtrees) simplify implementation of such multi-resolution algorithms in image processing.

A microscope itself represents a multi-resolution device because it is supplied with a set of objectives with different resolution. A typical task exploiting a coarse/fine resolution is navigation when the image captured using a low-resolution objective serves as a map where the user points at positions of interest from which a high-resolution image is to be captured. The system must be calibrated to determine the correspondence between the screen-coordinates and the real units required by the motorized stage. The navigation algorithms allow various applications like time lapse experiment, generation of stereological probes or image tiling and stitching.

2. PROBLEMS

Each objective has unique mechanical and optical properties leading to inhomogeneous illumination. A confocal laser scanning microscope (CLSM) is able to scan images of optical sections from different depths of a sample, which represents additional source of inhomogeneities.

Alignment of objectives. The exchange of objectives (manual or motorized) is usually accompanied by a misalignment of views. The image captured by a high-resolution objective represents an affine transform of the image captured from the same position by a low-resolution objective. This transform includes the coefficient of magnification and also a shift caused by imprecisely aligned objectives, which can be easily compensated by software.

Light attenuation with depth. It is well known that in image acquisition by CLSM, images obtained from deep layers of a specimen are often darker than images from the top layers due to absorption and scattering of both excitation and emitted fluorescent light. This effect causes problems in subsequent analysis of biological objects. Thus, an algorithm for the compensation of the light attenuation with depth has to be applied.

Uneven illumination. We assume that image regions corresponding to the same concentration of fluorescent material in the specimen should be mapped to the same grayscale levels. In practice, unfortunately, even ideally homogeneous specimens like artificial calibration grids exhibit darkening of image edges, lightening of the centre etc.; the effects are yet more pronounced in images of real biological specimens. A host of reasons account for the effect: optical imperfections of the objective (e.g. vignetting), surface of the specimen being non-perpendicular to the laser beam which makes the rays bend depending on the angle of incidence, locally curved surface of the specimen which concentrates or fans out the light, non-planar, curved, surface of focus, rather than a plane, which may miss a thin fluorescent layer, and the like. All these contribute to spatially uneven illumination of the specimen producing eventually the aforementioned artefacts. Besides making CLSM images unappealing, a spatially varying greyscale map complicates image post-processing. If, e.g., in image stitching of neighboring fields of view, overlapping regions possess spatially varying grayscale maps, similarity measures that assume spatial independence of the grayscale relationship may fail, with the result that a stitching algorithm sticks to a wrong spot.

Both, compensation of the light attenuation with depth and the light correction in the focal plane are solved in the next chapters.

2.1 Compensation of the light attenuation with depth

Image intensities from a CLSM suffer from light attenuation with depth caused by light loss due to two main effects—light aberration and photobleaching. As a result, images captured from deep layers of the specimen are darker than images from superficial layers, which make subsequent image analysis, segmentation, and 3D visualization of biological objects difficult.

Some methods for compensation of this effect has been addressed lately [1], [2]. They are based on rather complicated heuristics. However, a simple method with straightforward implementation

giving comparable results is often preferred in real laboratory practice.

In this article we present an efficient and fast method for compensation of the light attenuation based on matching the brightness of an image to be enhanced $\bar{I}$ with the apriori chosen reference image I_{ref}, using grey-level distribution functions ("cumulative histograms") of both images. While histogram matching [3] leads to a difficult optimization problem solved, e.g., by dynamic programming, matching distribution functions exploits their monotonicity resulting in simple optimization and fast execution. The brightest image of the stack should be chosen as a reference. If no satisfactory image is found (e.g. the whole stack is too dark) then a user-chosen image enhanced by a suitable software serves as a reference one. A short description of the algorithm follows.

Definition 1. Distribution function. Let

$$h = \{h_0, h_1, ..., h_{b_{\max}}\} \tag{1}$$

be the numbers of occurrence of respective image greyscale values $\{0,1,...,b_{max}\}$. We define the distribution function of an image as a mapping

$$S : b \in \{0,1,...b_{\max}\} \to \{0,1,..., n_{pixels}\} \tag{2}$$

$$S(b) = \sum_{i \le b} h_i \,. \tag{3}$$

Obviously, the distribution function is monotonic non-decreasing. From here on, we will assume that the reference image and the image being matched are of the same size. This implies that the respective distribution functions have the same final values. These two properties simplify the brightness matching task: for any value of S, we simply need to shift the corresponding brightness value until the value of S coincides with the value of S_{ref}.

Definition 2. Brightness matching of images. We will say that the brightness of an image $\bar{I}$ matches the brightness of an image I_{ref} within the resolution of the distribution function S_{ref} if, for any two brightness levels $\bar{b}_1, \bar{b}_2$, the number of pixels of $\bar{I}$ whose brightness lies between $\bar{b}_1, \bar{b}_2$ differs from the number of pixels of the reference image I_{ref}, whose intensities lie between $\bar{b}_1, \bar{b}_2$, by no more than the sum of two histogram values of I_{ref} at, or neighboring to, $\bar{b}_1, \bar{b}_2$.

Definition 3. Brightness matching of distribution functions.

Let S_{ref} be an image distribution function,

$$S_{ref} : b \in \{0,1,...b_{\max}\} \to \{0,1,..., n_{pixels}\} \tag{4}$$

.

Define an auxiliary boundary value $S_{ref}(-1) = 0$. Let S be an image distribution function. We define a brightness mapping g of the image distribution function S onto S_{ref},

$$g : b \to \bar{b}\ ,\ b, \bar{b} \in \{0,1,...b_{\max}\} \tag{5}$$

by the relationships

$$\begin{aligned} S(b) = S_{ref}(b) \ &\Rightarrow\ g(b) = b \\ S(b) < S_{ref}(b) \ &\Rightarrow\ g(b) = b_{ref} : S_{ref}(b_{ref} - 1) < S(b) \le S_{ref}(b_{ref}) \\ S(b) > S_{ref}(b) \ &\Rightarrow\ g(b) = b_{ref} : S_{ref}(b_{ref}) \le S(b) < S_{ref}(b_{ref} + 1). \end{aligned} \tag{6}$$

2.2 Compensation of illumination inhomogeneities

Inhomogeneous light is a frequently solved problem in image stitching [4]. We developed a specific solution of this problem based on estimating a spatially variable gain which models the adverse effects of uneven illumination, and correcting images by inverting the estimated gain. The proposed approach exploits a morphological operation called the upper Lipschitz cover [5].

In confocal laser scanning imaging, the image brightness is proportional to the concentration of the fluorescent agent in the specimen; for specimen regions having the *same concentration* of the fluorescent agent, corresponding regions of the resulting image should be *equally bright*, since the brightness obeys the formula [6]

$$I_{em}(x, y) = I_{ex}(x, y).Obj(x, y), \tag{7}$$

whereby I_{em} denotes the intensity of the emitted light, I_{ex} stands for the excitation intensity, and *Obj* represents the concentration of the fluorescent agent at the location *(x,y)* of the specimen. If constant, excitation intensity can be assumed across the specimen, emitted light intensity for object regions having the same fluorescent concentration should *in an ideal case* be the same.

In order to be able to treat the problem of compensation of illumination inhomogeneities, we will make additional assumptions pertaining to CLSM imaging: First, the excitation intensity is constant across the whole specimen. Second, the emitted light at different pixel positions *(x,y)* is amplified or attenuated by diverse effects which we will summarize into a single function called *gain(x,y)* yields the relationship between the fluorescent concentration and the recorded light intensity I_{rec}:

$$I_{rec}(x, y) = gain(x, y).I_{em}(x, y) = gain(x, y).Obj(x, y).I_{ex}. \tag{8}$$

If we knew *gain(x,y)*, we could easily correct the *recorded image* I_{rec} to obtain the *emitted light* I_{em}, which is what we are really interested in:

$$I_{corr}(x, y) = (1/gain(x, y)).I_{rec}(x, y) = I_{em}(x, y). \tag{9}$$

Therefore, we will attempt to get an *estimate* of *gain(x,y)*, the function that summarizes all adverse effects on the recorded greyscale values, without even knowing the exact physical causes of these effects in detail.

I_{ex} is constant by assumption; what we see in a recorded image is the product of the fluorescent concentration in the object *Obj(x,y)* and the distortions *gain(x,y)*. If arbitrary types of objects are admitted, the functions *Obj(x,y)* and *gain(x,y)* cannot generally be distinguished. However, real CLSM objects show some features that are not to be expected in the empirical gain. So, for example, the gain changes slowly, is continuous and therefore has only a small number of minima or maxima, while the concentration of the fluorescent dye in the microscope specimen changes abruptly

on tissue boundaries, which in turn produces discontinuities and consequently high frequencies.

Our estimate of the *spatially variable gain* is based on the following reasoning: If the *concentration* of the fluorescent dye were *constant* (*e.g.* maximal) over the whole scanning layer of the object, the grayscale distribution of the captured image would possess form identical (up to a scaling constant) to that of the *gain* function. We will assume that local maxima (or at least some of them) in the acquired image correspond to specimen regions with the highest fluorescent dye concentration, and try to pad sub-maximal regions with this maximal concentration. Of course, this cannot be done physically on the specimen - instead, we *modify the brightness of the acquired image* (which frequently violates the condition of a slow change due to discontinuities at tissue boundaries and the like) so as to make it look like an image of an *object* with constant fluorescent dye concentration.

Because the form of the modified image should reflect the form of the *gain*, it must comply with the condition that *the speed of change* is slow. This requirement can be cast into the formula:

$$|gain(x_1, y_1) - gain(x_2, y_2)| \le K.|(x_1, y_1) - (x_2, y_2)| \qquad (10)$$

for any two pixels (x_1, y_1), (x_2, y_2) (Lipchitz condition). "Filling" the acquired image so as to satisfy the previous equation can be done very fast by subjecting the captured image to a *morphological operator* called "the upper Lipschitz cover". The *upper Lipschitz cover* of an image $I(x,y)$ is the infimum of functions $L(x,y)$ satisfying the conditions:

$$|L(x_1, y_1) - L(x_2, y_2)| \le K.|(x_1, y_1) - (x_2, y_2)|,$$
$$L(x, y) \ge I(x, y). \qquad (11)$$

3. SOFTWARE

We developed a CLSM software package which can be used by the operator to capture images having different resolutions. The light compensation algorithm can be applied to captured stacks. The whole system is controlled by main dialog with a tree-like control. A manually entered (clicked) position on left image can be saved which adds a new node to a child level of a tree control. After entering the child nodes, the SCAN process moves the stage to saved positions and captures corresponding images. Any node can be defined as a new parent and the whole DEFINE/SCAN process can continue on the next level acquiring images at a higher magnification.

It is up to operator's decision what parents and how deep in the tree structure will he continue. Some nodes evidently represent regions that are not interesting and where the higher resolution image acquisition makes no sense. Automatically generated positions are very useful in applications like image tiling, stitching, generation of stereological probes, etc. Software was developed in C++ as a plug-in module of Ellipse image processing software package (ViDiTo Systems, Slovakia).

4. RESULTS

The images captured by CLSM from different depths of the same specimen suffer from the light attenuation with depth as well as the darkening of pixels in the edges of the fields of view. Compensation of both inhomogeneities is demonstrated on Fig. 1 using two different physical slices.

In our practical experiments we used a postcranial part of a 17-day-old Wistar rat embryo embedded in paraffine. The embryo was cut into thin, 30 μm thick, physical slices by a microtome. Slice1 was captured by CLSM in 3x5 neighboring positions with overlap 10%, and then a new image was created as a mosaic. Since CLSM is capable to focus through the physical slices, each mosaic consists of five images acquired from different depths (7.5 μm apart) of the same physical slice. Slice1 in 3D consists of 3x5x5=125 stitched images (first row on Fig.1). Smaller Slice2 consists of 3x4x5 images (third row). The second and fourth rows of images show the results of applying the respective two image enhancement algorithms. It is shown that both mentioned effects were removed.

The C implementation of the algorithm is very fast, it needs approximately 0.02 s to match a single pair of 512x512 8 bit images.

5. CONCLUSION

Despite the huge progress in technology, using microscopy for quantitative analyses is still very time-consuming. Most of the samples in biological research are too complicated to be acquired without the human interaction and therefore fully automated multi-resolution microscopy is successful only in very specific areas.

The presented paper solves problems of the light correction in confocal microscopy as a part of software for interactive multi-resolution data acquisition. The proposed solution allows automated data acquisition from interactively selected positions having different resolutions. Acquired images have reduced light inhomogeneities in both axial and lateral axes.

This work was supported by the research grants from the Slovak Grant Agency VEGA No. 0164, APVV 0682-07, SK-CZ 0030-07, by Slovak Academy of Sciences in frame of CEX NANOFLUID and by the European Regional Development Fund, Project no. 26220120021. In the Czech Republic was this work supported by grants GAČR 102/08/0691 and MSM6840770012.

6. REFERENCES

[1] LEE, S.C. AND BAJCSY, P. 2006. Intensity correction of fluorescent confocal laser scanning microscope images by mean-weight filtering. *Journal of Microscopy-Oxford* ***221***, 122-136.

[2] GOPINATH, S., WEN, Q., THAKOOR, N., LUBY-PHELPS, K. AND GAO, J.X. 2008. A statistical approach for intensity loss compensation of confocal microscopy images. *Journal of Microscopy-Oxford* ***230***, 143-159.

[3] CAPEK, M., JANACEK, J. AND KUBINOVA, L. 2006. Methods for compensation of the light attenuation with depth of images captured by a confocal microscope. *Microscopy Research and Technique* ***69***, 624-635.

[4] JIA, J.Y. AND TANG, C.K. 2008. Image stitching using structure deformation. *Ieee Transactions on Pattern Analysis and Machine Intelligence* ***30***, 617-631.

[5] STENCEL, M. AND JANACEK, J. 2006. On calculation of chamfer distance and Lipschitz covers in digital images. In *Proceedings of the S4G*, Prague, Czech Republic, 517-522.

[6] HEINTZMANN, R. 2008. Advanced optical microscopy: challenges and opportunities (Part II, indirect methods), ISBI Paris

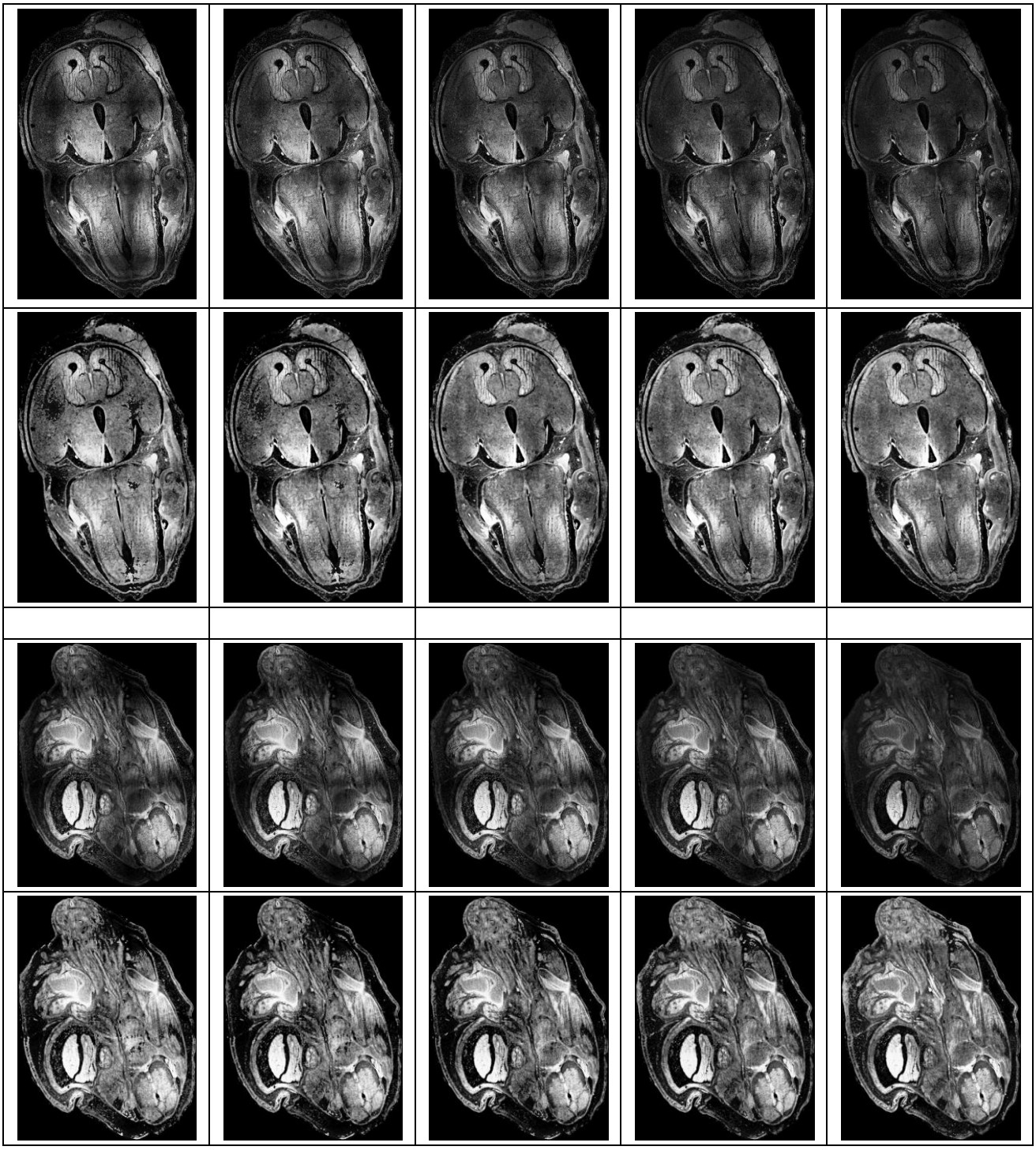


Fig. 1. Compensation of light in CLSM demonstrated on a rat embryo sample. Each image is created by stitching high-resolution images together (3x5 images from slice1 and 3x4 images from slice2). Rows 1 and 3 show the increasing light attenuation from left to right caused by increasing depth of acquisition and also the light inhomogeneities inside the individual images. Rows 2 and 4 represent the light compensation after the application of both algorithms.

Local Contrast Enhancement for Improving Screen Images Exposed to Intensive External Light

D.S. Lebedev[1], G.I. Rozhkova[1], V.A. Bastakov[1], C.-Y. Kim[2], S.-D. Lee[2]

[1] Kharkevich Institute for Information Transmission Problems, Russian Academy of Sciences, Moscow, Russia
[2] Samsung Advanced Institute of Technology, Samsung Electronics Co., Republic of Korea

Abstract

When a mobile device (notebook, smartphone, etc.) is used, the images on the monitor screen may be significantly distorted by a strong external light (for instance, sunlight) illuminating the screen surface. This additional illumination reduces all the local contrast values in the displayed images and leads to fading and deterioration of the subjective visual images thus impeding image recognition and analysis. A technique of processing the input digital video signal in such a way that, in presence of an additional external illumination, the local contrast values of the displayed image are made as close as possible to the original ones is described. The significant image enhancement due to this technique has been demonstrated in experiments with the illuminated monitor screen and has been estimated mathematically via the distributions calculated for local contrast values before and after processing video signal.

Keywords: *Contrast enhancement, Displayed images, Local contrast, External illumination, Ambient light correction*

1. INTRODUCTION

The images on the monitor screen may be distorted by a strong light illuminating the screen surface. Such a situation occurs rather frequently when a mobile device (a notebook, a smartphone, etc) is used under conditions of bright natural light or artificial illumination of high intensity levels. This additional external illumination reduces all the local contrast values in the displayed images relative to those in the original images observed in the absence of such illumination. Since successful contrast detection is one of the most important conditions for an adequate visual perception, decrease of the local contrast values could result not only in a general deterioration of subjective image quality but also in failures and errors in visual performance.

Evidently, to improve recognition of details in the displayed image, it is necessary to compensate somehow for the loss of the perceived image quality. One of the possible ways is to employ some method of contrast enhancement. A lot of procedures for contrast enhancement have been proposed during the last thirty years, especially for the purposes of aviation, medical digital radiography, infrared imaging systems [Beghdadi and Negrate 1989, Dash and Chatterji 1991, Weeks 1996, Chang and Wu 1998, Zhu et al. 1999, Mukhopadhyay and Chanda 2000, Kim et al. 2001, Cheng et al. 2003, Branchitta et al. 2008] where the initial images could be of a poor quality. These procedures have been classified into two groups: direct methods implying modification of calculated contrast values by some functions, and indirect methods using modification of histograms, in particular, histogram equalization**.** In fact, many of these procedures are equivalent to low-pass and high-pass filtering for separating smooth and detail areas of an image and enhancing the detail components.

Despite essential differences, all these methods are so-to-say universal in the sense that they could be applied to any original image independently of the cause of its low quality. The task of image processing is formulated rather generally: to obtain the better image and, at the same time, to avoid over-enhancement and under-enhancement. Our study concerns a partial case, when the quality of the displayed image is poor due to the external illumination of the monitor screen surface. For this case, several variations of the general approaches could be found in the literature available: gamma correction with a lower value of gamma than in the absence of ambient light [Ware 2000], a novel remapping function including the measured value of the ambient light [Delvin et al. 2006], high-pass and low-pass filtering depending on the ambient light [O'Dea et al. 2004/2009]. However, in the case of distortion caused by the external illumination the task of image enhancement can be formulated more precisely.

Proceeding from the assumption that the quality of the original image is good enough and taking into account the crucial role of local contrasts in human perception, we suppose that to improve the quality of the displayed image washed-out by the ambient light, it is rational to find a transformation of the original input digital video signal into the signal that make the local contrast values as close as possible to those of the original image. A straightforward algorithm of such a transformation is described below. In this paper, we restricted ourselves to the simplest case: a *monochrome* original image and *smoothed* additional luminance of the screen.

2. THEORETICAL APPROACH TO FINDING THE LOCAL CONTRAST ENHANCEMENT ALGORITHM

2.1. Formulating the Problem

Let an original monochrome image be described by the luminance distribution on a monitor screen surface, $U(x, y)$, where x, y are the coordinates of the surface points. Suppose that the coordinates are of discrete (integer) values and set in a rectangular region, $\mathbf{R}$: $x = 0,..., N-1$, $y = 0,..., M-1$. Then the image may be presented as the luminance array $\mathbf{U} = \{U(x, y),\ (x, y) \in \mathbf{R}\}$ consisting of NM pixels. The original image is determined by *the input digital video signal*, that is, the array of brightness code words, $\mathbf{c}_{\mathbf{U}} = \{c_{\mathbf{U}}(x, y), (x, y) \in \mathbf{R}\}$, where the brightness code word, $c_{\mathbf{U}}(x, y) = 0, 1,..., 255$. The luminance at the point $(x, y) \in \mathbf{R}$ is an increasing function of the corresponding brightness code:

$$U(x, y) = F(c_{\mathbf{U}}(x, y)).$$

The function $F(\ldots)$ is the *tone characteristic* of the monitor. Using this function, the original image may be expressed as

$$\mathbf{U} = \{U(x, y) = F(c_{\mathbf{U}}(x, y)),\ (x, y) \in \mathbf{R}\}.$$

Suppose that the monitor screen is illuminated by an external light source. It results in an additional luminance, $U_0(x, y)$, at every screen point. The new luminance distribution,

induced *by the same input signal* in the presence of the external illumination of the screen, is the distorted image,

$$\mathbf{V} = \{V(x,y) = F(c_{\mathbf{U}}(x,y)) + U_0(x,y),\ (x,y) \in \mathbf{R}\}.$$

Due to the external illumination the visual quality of the distorted image on the monitor screen is lower than of the original one. Our task is to find a transformation of the original input video signal into another signal,

$$\mathbf{c}_{\text{opt}} = \{c_{\text{opt}}(x,y),\ (x,y) \in \mathbf{R}\},$$

inducing the screen image,

$$\mathbf{U}_{\text{opt}} = \{U_{\text{opt}}(x,y),\ (x,y) \in \mathbf{R}\},$$

that is *subjectively* as close as possible to the original image in spite of the presence of the external screen illumination.

2.2. Local Contrast

The measure of similarity of the two images used in this study was based on the differences of their local contrast values. It is natural to define the local contrast at the point (x,y) of the original image, $\mathbf{U} = \{U(x,y),\ (x,y) \in \mathbf{R}\}$, as

$$(1)\quad C_{\mathbf{U}}(x,y) = \frac{U(x,y) - \bar{U}(x,y)}{\bar{U}(x,y)},$$

where $\bar{U}(x,y)$ is a weighted average luminance of the points surrounding a given point (x,y), that is

$$(2)\quad \bar{U}(x,y) = \sum_{i,j=-d}^{d} a_{i,j} \cdot U(x-i, y-j).$$

In this equation the parameter *d* characterizes the size of the surrounding area, and $a_{i,j},\ i,j = -d,...,d,$ are positive normalized weight coefficients such, that $\sum_{i,j=-d}^{d} a_{i,j} = 1$. The image

$$\bar{\mathbf{U}} = \{\bar{U}(x,y),\ (x,y) \in \mathbf{R}\}$$

may be treated as a smoothed copy of the given image.

It is obvious that Eqs (1) and (2) may be applied to any image. In particular, the local contrast at the point (x,y) of the distorted image $\mathbf{V}$ is

$$C_{\mathbf{V}}(x,y) = \frac{V(x,y) - \bar{V}(x,y)}{\bar{V}(x,y)}.$$

Since $V(x,y) = U(x,y) + U_0(x,y)$, then $\bar{V}(x,y) = \bar{U}(x,y) + \bar{U}_0(x,y)$. In the case of smoothed additional luminance of the screen, the following approximation is valid: $\bar{U}_0(x,y) \approx U_0(x,y)$. Therefore,

$$C_{\mathbf{V}}(x,y) \approx \frac{U(x,y) - \bar{U}(x,y)}{\bar{U}(x,y) + U_0(x,y)}.$$

That is,

$$(3)\quad C_{\mathbf{V}}(x,y) \approx C_{\mathbf{U}} \cdot \frac{\bar{U}(x,y)}{\bar{U}(x,y) + U_0(x,y)}.$$

Thus, the additional illumination of the screen *decreases the local contrast value at any point.* The darker is the image region, the more is contrast decreasing (cf. **Figure 1**, A, and B).

2.3. Algorithm

We define the similarity of two images, $\mathbf{U}$ and $\mathbf{W}$, as the square of *Euclidean distance* between the corresponding arrays of local contrasts:

$$\rho(\mathbf{U},\mathbf{W}) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (C_{\mathbf{U}}(x,y) - C_{\mathbf{W}}(x,y))^2.$$

Using the definition of the local contrast (Eq (1)) one may rewrite this expression as

$$(4)\quad \rho(\mathbf{U},\mathbf{W}) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (C_{\mathbf{U}}(x,y) - \frac{W(x,y)}{\bar{W}(x,y)} + 1)^2$$

It is evident that the *optimal enhanced image* $\mathbf{U}_{\text{opt}}$ is a solution of the following equation:

$$\rho(\mathbf{U},\mathbf{U}_{\text{opt}}) = \min_{\mathbf{W}} \rho(\mathbf{U},\mathbf{W}).$$

The straightforward technique of solving the problem is 'the simple iteration method'. Let one know the *m*-th approximation to the enhanced image,

$$\mathbf{U}_{\text{opt}}^{(m)} = \{U_{\text{opt}}^{(m)}(x,y),\ (x,y) \in \mathbf{R}\}.$$

The first step is calculating the *m*-th average luminance according to Eq (2):

$$(5)\ \bar{U}_{\text{opt}}^{(m)}(x,y) = \sum_{i,j=-d}^{d} a_{i,j} \cdot U_{\text{opt}}^{(m)}(x-i, y-j),\ (x,y) \in \mathbf{R}.$$

Substituting $\mathbf{U}_{\text{opt}}^{(m)}$ and $\bar{\mathbf{U}}_{\text{opt}}^{(m)}$ into Eq (4) one obtains

$$\rho(\mathbf{U},\mathbf{U}_{\text{opt}}^{(m)}) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (C_{\mathbf{U}}(x,y) - \frac{U_{\text{opt}}^{(m)}(x,y)}{\bar{U}_{\text{opt}}^{(m)}(x,y)} + 1)^2.$$

The next approximation $\mathbf{U}_{\text{opt}}^{(m+1)}$ should decrease the distance $\rho(\mathbf{U},\mathbf{U}_{\text{opt}}^{(m)})$. It is reasonable to find the approximation as the solution of the following equation:

$$\sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (C_{\mathbf{U}}(x,y) - \frac{U_{\text{opt}}^{(m+1)}(x,y)}{\bar{U}_{\text{opt}}^{(m)}(x,y)} + 1)^2 = 0.$$

It is evident that the solution is

$$(6)\ U_{\text{opt}}^{(m+1)}(x,y) = (C_{\mathbf{U}}(x,y) + 1) \cdot \bar{U}_{\text{opt}}^{(m)}(x,y),\ (x,y) \in \mathbf{R}$$

If the conditions $\text{Min} \le U_{\text{opt}}^{(m+1)}(x,y) \le \text{Max},$ where $\text{Min} = U_0(x,y)$, $\text{Max} = \text{Min} + F(255)$, are violated, then it is necessary to replace the calculated value (6) with one of the extreme values:

$$(7)\ U_{\text{opt}}^{(m+1)}(x,y) = \begin{cases} \text{Min}, & \text{if calculated value less than Min} \\ \text{Max}, & \text{if calculated value more than Max} \end{cases}.$$

As an initial approximation, $\mathbf{U}_{\text{opt}}^{(0)}$, one could use the distorted image $\mathbf{V}$.

After obtaining the final (*n*-th) approximation, $\mathbf{U}_{\text{opt}}^{(n)}$, it is necessary to find the transformed input digital signal

$$\mathbf{c}_{\text{opt}} = \{c_{\text{opt}}(x,y),\ (x,y) \in \mathbf{R}\}.$$

Since $U_{\text{opt}}^{(n)}(x,y) = F(c_{\text{opt}}(x,y)) + U_0(x,y)$, then

$$(8)\quad c_{\text{opt}}(x,y) = F^{-1}(U_{\text{opt}}^{(n)}(x,y) - U_0(x,y)),$$

where the function $F^{-1}(...)$ is the inverse function to the tone characteristic of the monitor. An example of the enhanced image is shown in **Figure 1**, C.

It is evident that without external light video signal $\mathbf{c}_{\text{opt}}$ would produce the image

$$\mathbf{U}^* = \{U_{\text{opt}}^{(n)}(x,y) - U_0(x,y),\ (x,y) \in \mathbf{R}\}.$$

The local contrast of this image exceeds the original one at any point (see **Figure 1**, D).

The iteration procedure described by Eqs (5)-(7) converges. However, the convergence is rather slow. Nevertheless, it is possible to obtain considerable enhancement of the displayed image after a reasonable number of iterations. The further improving of the similarity by means of increasing the iteration number is not often noticed by observers.

Some results of the algorithm application to the real pictures are presented in the next section.

3. EXPERIMENT

As it follows from the theoretical analysis presented in Section 2, to apply the proposed algorithm for contrast enhancement under conditions of the external illumination, one has to know: (i) the tone characteristic of the monitor, $F(\ldots)$, and (ii) the level of the equivalent luminance $U_0(x, y)$ produced by the external illumination of the screen surface.

3.1. Monitor Tone Characteristic

Description of the tone characteristic might be presented in the form of the tables containing screen luminance values for all values (0-255) of the input digital video signal. Such tables should be created for all the monitor settings selected by the user. Instead of the tables, one could use appropriate mathematical formulas.

The tone characteristic of the LCD monitor ACER AL1751W used in our experiments was based on the measurements of the screen luminance for the discrete signal values separated by 10 units. In the course of measurements, it is not rational to use smaller intervals since the luminance increments/decrements will be too small. However, for the purposes of signal processing it is necessary to know luminance for each input signal value. Taking into account natural assumption about a good smoothness of monitor tone characteristic, it is reasonable to use some kind of interpolation instead of measurements of the luminance values for each of 256 input values. In our study, the simplest cubic spline interpolation technique was implemented.

3.2. Equivalent Luminance Produced by the External Illumination

In order to evaluate the effect of the external illumination on the perceived luminance of the monitor screen, it was divided into two equal parts and one of them was *uniformly* illuminated by the external source. Thus, the luminance of one part was determined solely by input video signal while the luminance of the adjacent second part was determined both by the input signal and the external illumination. The physical luminance of the screen was varied by varying input signals designed to both parts of the screen and the level of the external illumination. Since the optical density of the liquid crystal elements depend on input signal, a priori, one could not exclude the possibility that, in the LCD monitor, the effect of illumination would be dependent on the screen luminance.

The task of the subject was to make the two screen parts visually identical in perceived luminance by controlling the input signal to the screen part that was not illuminated. The results were presented as the relationship between the two input signals. Comparing these data with corresponding tone characteristic, one could estimate the equivalent constant luminance added to the own screen luminance due to certain external illumination. Since the details of the physical measurements have no direct relation to the subject matter of the conference, they are omitted here.

The experiments with luminance matching made us sure that the equivalent luminance due to an external illumination was not dependent on the own luminance of the LCD monitors. In other words, in order to find the equivalent luminance caused by a given external illumination, it is really enough to determine the input signal value c_e that provides the screen luminance equal to the luminance caused by the external illumination alone (i. e. when the input signal value applied to the illuminated part is equal to 0).The value of the equivalent luminance U_0 that has to be used in calculation of contrast (see Section I) should be $U_0 = F(c_e)$

3.3. Application of the Algorithm to Real Pictures

The algorithm described by Eqs (5) - (8) was applied to some real pictures. The additional luminance, $U_0(x, y)$, was set equaled to 100 cd/m^2 at every screen point. It corresponds to ca. 5000 lx of the screen external illumination. The normalized weight coefficients in Eq (2) were chosen as follows:

$$a_{i,j} = a(i)\cdot a(j),\ \ i, j = -2,\ldots 2;\ \ a(0) = 6/16,$$

$$a(-1) = a(1) = 4/16,\ \ a(-2) = a(2) = 1/16.$$

The luminance of the original pictures was converted to the range [2..275] cd/m^2 so that the luminance of distorted pictures belong to the range [102..375] cd/m^2. For the used monitor ACER AL1751W with selected brightness/contrast settings these ranges correspond to the video signal ranges [0..223] and [145..255].

In **Figure 1,** A**,** one of the original images, “Cupola” (512x635 pixels, http://www.photoforum.ru/photo/315333, author: Alexander Kolesnikov), is depicted. The image in **Figure 1,** B is faded due to the external light. The image enhanced by two iterations is shown in **Figure 1**, C. One could see that many details of the original which had disappeared from the faded image reappeared again after the enhancement (pay attention to the dark areas). The last image (**Figure 1,** D) is reproduced by means of the transformed video signal $\mathbf{c}_{opt}$ defined by Eq.(8) in the absence of the external light, and looks as somewhat over-enhanced.

3.4. Quantitative Estimation of Image Fading and Image Enhancement

The external illumination inevitably leads to some narrowing the range of the local contrast values since all of them decreases in the absolute magnitude (see Eq.(3)). On the other hand, application of any effective algorithm for image enhancement should automatically increase the contrast range of the image.

Taking this into account, we propose to estimate image fading/enhancement in parallel with image processing. As a measure of the image contrast range we propose to use the standard deviation of the local contrast values. **Figure 2** illustrates this approach. It contains 3 histograms of contrast values obtained for the images shown in **Figure** 1, A, B, C. The overall range of the contrast values [-1, 1] is divided in bins of 0.02 width and the percentages of the values in each bin are calculated. For each histogram corresponding peak value (the zero bin percentage) and the local contrast standard deviation are indicated. A significant narrowing of the contrast distribution due to the external illumination and its essential widening due to processing the video signal for image enhancement are well expressed.

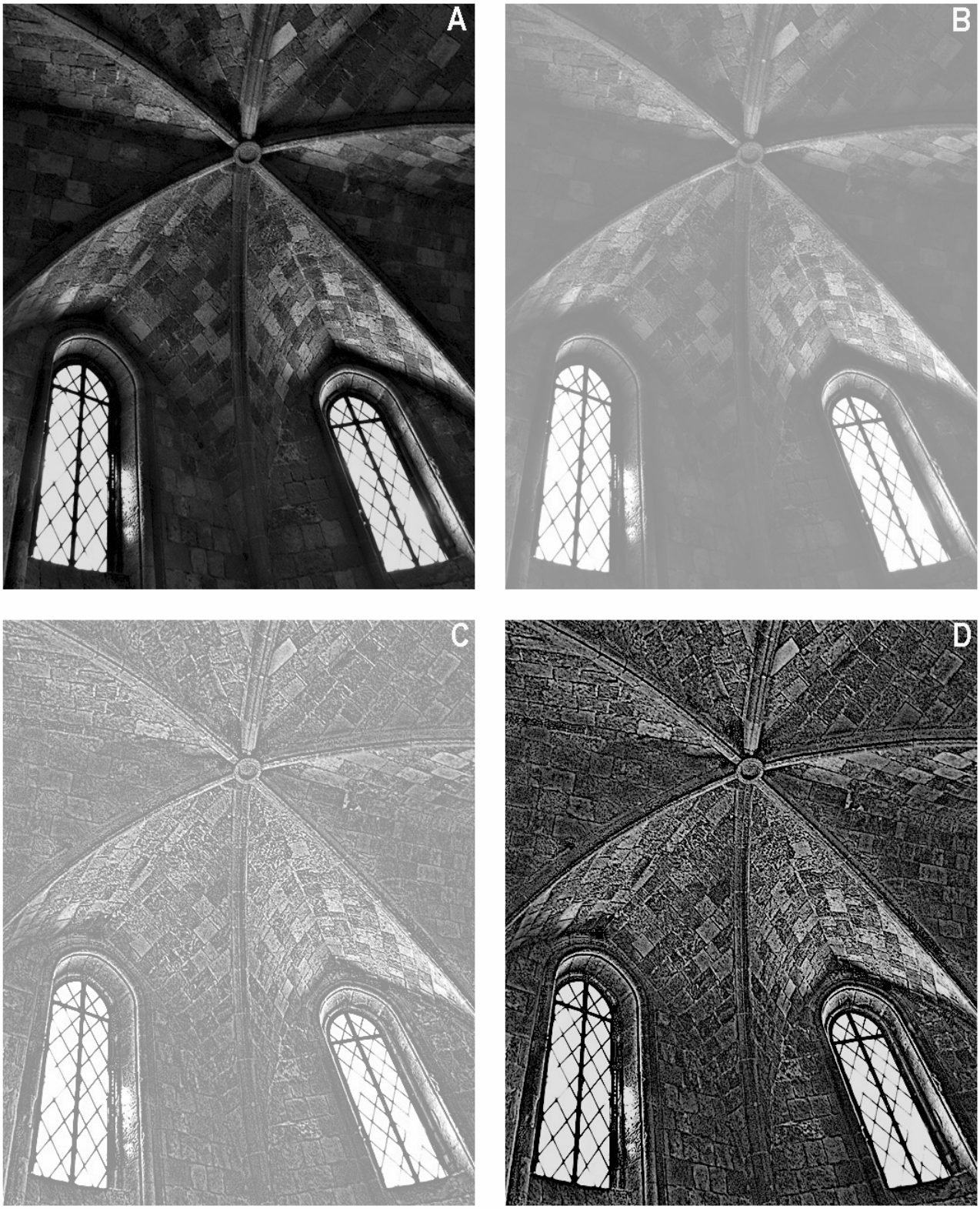


Figure 1: Result of the local contrast enhancement of a real image. A – Original image “CUPOLA” (512 x 635 pixels); B – the image distorted by external illumination; C, D - the enhanced images *with* (C) and *without* (D) external illumination.

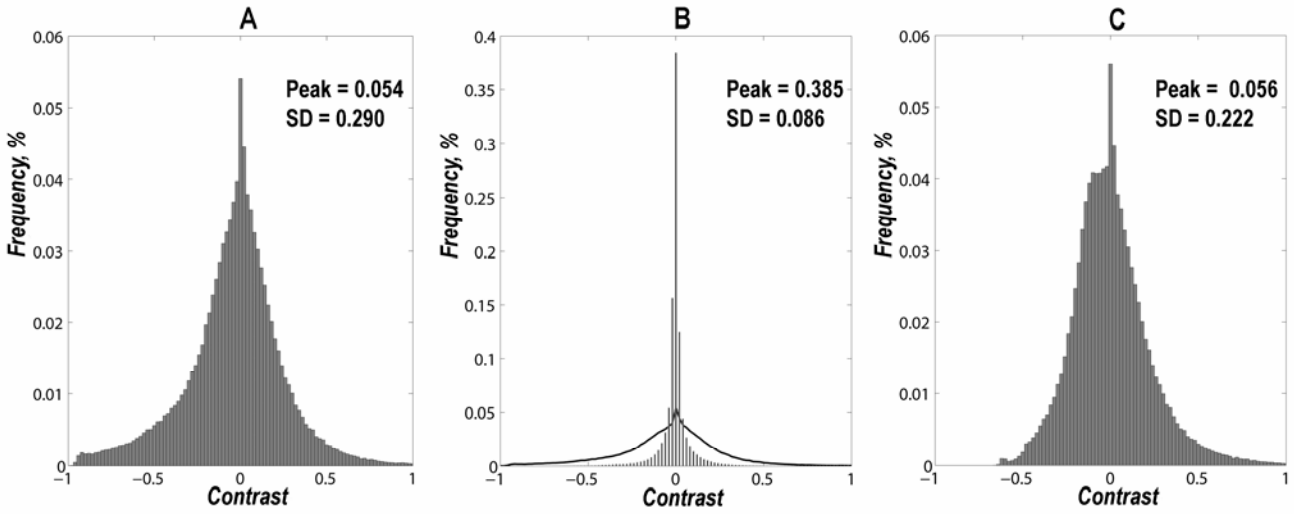


Figure 2: The histograms of the local contrast values. A – for the original image; B – for the image distorted by external illumination (the black line is the histogram A envelope curve); C – for the enhanced image with external illumination.

4. CONCLUSIONS

An effective method of image improvement has been proposed for a partial case when a low quality of displayed images is caused by intensive ambient illumination of a screen.

The method is based on restoration of the local contrast values and implies a transformation of the original video signal depending on the monitor tone characteristic and the level of the external illumination. However, the method is rather robust and does not require knowing of the precise values of these parameters.

The proposed approach to restoration of the local contrast values may find wide application for enhancing variety images of different types (including X-rays, infrared and so on).

The investigation has been carried out in the framework of the Research and Development Agreement (2004) between Institute for Information Transmission Problems RAS and Samsung Electronics Co. Ltd.

Our patent application to US, **Method of, and Apparatus for Image Enhancement Taking Ambient Illuminance Into Account,** was accepted 2006.02.15. Application number: 20060187242.

REFERENCES

Beghdadi A., Negrate A.L.1989. Contrast enhancement technique based on local detection of edges, *Comput. Vision Graphics Image Process*. 4, 6 162-174.

Branchitta F., Diani M., Corsini G., and Porta A. 2008. Dynamic-range compression and contrast enhancement in infrared imaging systems. *SPI Opt. Eng. 47*, 076401-076414

Chang D., Wu W. 1998. Image contrast enhancement based on a histogram transformation of local standard deviation, *IEEE Trans. Med. Imaging 17(4)*, 518-531.

Cheng H.D., Xue M., and Shi X.J. 2003. Contrast enhancement based on novel homogeneity measurement. *Pattern Recognition, 36*, 2687-2697.

Dash L., Chatterji B.N. 1991. Adaptive contrast enhancement and de-enhancement, *Pattern Recognition, 24*, 289-302.

Delvin K., Chalmers A., and Reinhard E. 2006. Visual calibration and correction for ambient illumination. *ACM Transactions on Applied Perception 3(4)*, 429-452.

Kim J-Y., Kim L-S., and Hwang S-H. 2001. An advanced contrast enhancement using partially overlapped sub-block histogram equalization. *IEEE Trans. Circuits and Systems for Video Technolog, 11(4).* 475-484.

Mukhopadhyay S., and Chanda B. 2000. A multiscale morphological approach to local contrast enhancement. *Signal Processing 80*, 685-696.

O'Dea S. R., O'Connell M., and Berardi W. Enhancing contrast. 2009. US Patent 7,545,397 June 9, 2009.

Ware C. 2000. *Information Visualization: Perception for Design*. Morgan Kauffman, San Francisco, CA.

Weeks A. 1996. *Fundamentals of Electronic Image Processing*. SPIE/IEEE Press.

Zhu H., Chan F.H.Y., Lam F.K. 1999. Image contrast enhancement by constrained local histogram equalization. *Comput. Vision Image Understanding, 73 (2)*, 281-290

About the authors

Dmitry S. Lebedev is a Doctor of Science, Professor, Sheaf Scientist of Laboratory of Sensory Information Processing (LSIP), Institute for Information Transmission Problems (IITP) RAS, email: lebedev@iitp.ru.

Galina I. Rozykova is a Doctor of Science, Professor, Leading Scientist of LSIP, IITP RAS, email: gir@iitp.ru.

Vladimir A. Bastakov is a PhD, Professor, the Head of LSIP, IITP RAS, email: bastakov@iitp.ru.

Chang-Young Kim is the Vice President, Samsung Advanced Institute of Technology (SAIT), the Director, Multimedia Lab., SAIT, Samsung Electronics Co. email: cykim@samsung.com.

Seong-Deok Lee is the Vice President SAIT, the Head of Digital Imaging Group, Multimedia Lab., SAIT, Samsung Electronics Co. email: lsdlee@samsung.com.

Automatic Photo Selection for Media and Entertainment Applications

Ekaterina Potapova, Marta Egorova, Ilia Safonov
Moscow Engineering Physics Institute (National Nuclear Research University), Moscow, Russia
ep.slip at gmail.com, marta.egorova at gmail.com, ilia.safonov at gmail.com

Abstract

We propose an algorithm for automatic photo selection for media and entertainment applications like photobook and slide-show. The technique comprises three main steps: photo quality estimation and elimination of poor-quality photos, adaptive quantization of survived photos in time-camera plane, and selection of the most appealing photos from each quantized group. For detection of low-quality photos complex classifier comprising of two AdaBoost classifiers committees is created. Photos with exposure defects, such as over- and underexposed, backlit, blurred photos as well as images affected by strong JPEG artifacts are detected confidently. For quantization of photos the method similar to median-cut color quantization is proposed. The appealing photos are selected basing on novel scheme via comparison of visual salience among several images as well as face detection. Our method of identification of the most salient photo among others is based on Itti-Koch-Niebur algorithm of saliency map building. Obtained results of selection as well as time performance issues are discussed. The majority of observers were pleased with the results of the algorithm.

Keywords: *low-quality photo detection, automatic photo selection, visual salience.*

1. INTRODU CTION

At present time almost all persons have digital photo camera and capture a thousand photos. Frequently some individuals use several cameras simultaneously, for example, Digital Still Camera (DSC), camera in mobile phone, camcorder. Browsing and viewing very large photo collections are exhausting work. How to keep order in such huge heap of photos? How to select good photos for slide-show, printed photobook and similar applications? Manual selection from huge set of photos takes a long time and it is tiresome. Researchers from HP Labs [1] remark the following: "people do not create as many photobooks as they would like to, one of the reasons being the image selection process is too painful in the current digital photography landscape, in which hundreds or even thousands of photos are taken in one single event". Accordingly automatic photo selection for media and entertainment applications is a topical task.

Traditional album with printed photos, printed photobook, Web-album, slide-show for PC or digital photo frame, DVD slide-show and so forth relate to media and entertainment applications, which telling about events such as travelling or party with friends and relatives. It is required for such applications to select M photos from N, where $N>M$, sometimes $N>>M$ in order to target a specific final photo count, while preserving a good coverage of the event as well as selection of high-quality photos only. In given paper we solve the problem of selection about 100-200 photos among collection from 500-1000 photos or selection of 20-30 photos from several hundreds. In the scope of the task additional sub-tasks occur for example selection of several representative photos for cover of photobook or DVD, selection of K photos for each page of photobook, which are related to the same event, etc. Fig. 1 demonstrates collection from 30 photos. We use the collection for demonstration of automatic selection of 10 photos by proposed algorithm. The selected photos are outlined by red dot line.

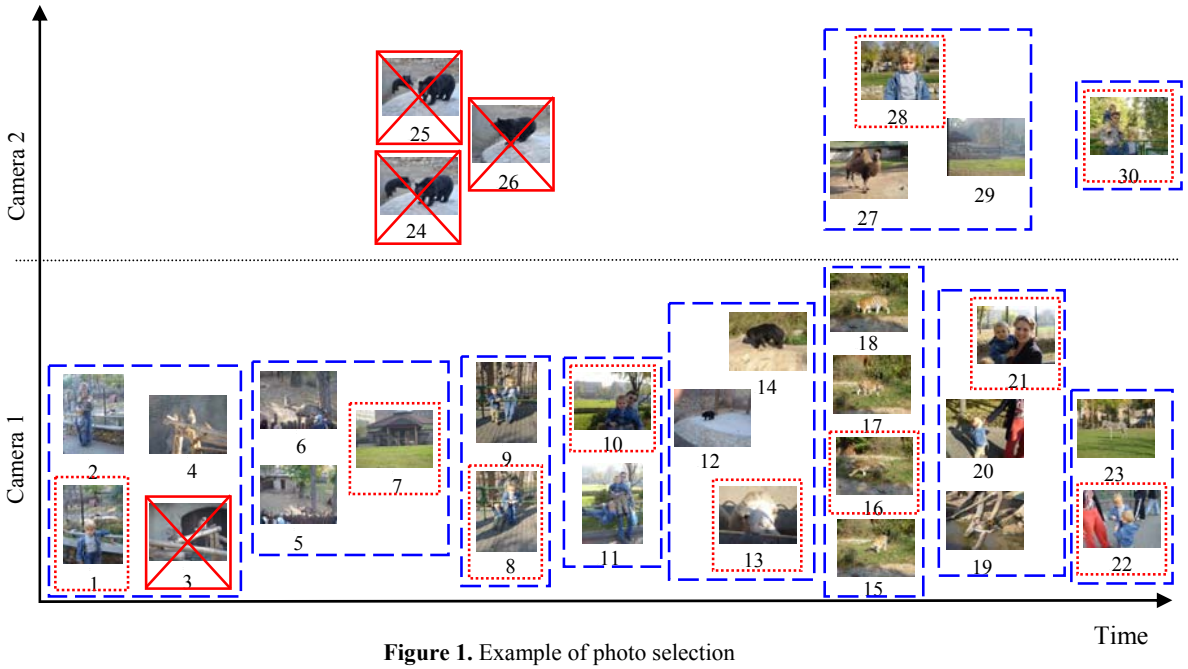


Figure 1. Example of photo selection

High photo processing performance is an important challenge for such kind of applications. The thousand photos can be processed and total selection procedure should take acceptable time, for example several minutes on modern PC. That means we should reach processing time for one photo less or at least about 1 second.

2. RELATED WORKS

Just appeared paper [1] discusses the same problem: automatic selection of images for photobook. The paper describes hierarchical time clustering, which is traversed at a specific hierarchy level in order to select images by alternating among all time clusters, and selecting the most relevant images in that cluster. The relevance ordering is based on a combination of features such as detected faces and smile, image appeal measures, where the measures such as sharpness, contrast, colorfulness, homogeneity are calculated for segmented regions. Exclusion of duplicate photos is based on the time analysis. In general the algorithm looks reasonable but it has a lot of non-obvious heuristic parameters and it looks too complex for fast implementation.

At present time for photo browsers and media applications clustering, which is based on the time of photo, where time is extracted from EXIF, is a common approach [2, 3, 4]. In addition these techniques try to exploit content-based information. In [5] blurred, underexposed and overexposed photos are excluded from analysis automatically. However other types of low-quality images are not considered. Time-based clustering is used to select photos for a slideshow.

Paper [6] is devoted to collage creation including automatic photos selection for the collage. The described algorithm is realized in impressive software application MS Research AutoCollage. The representative images are selected in three different ways: textually "interesting", mutually distinct and presence of faces in the image. "Interestingness" of the image is assessed applying entropy of histograms *ab* in color space *Lab*, mutual difference in distance between their histograms *ab*. Images with greatest entropy are interesting. However, in our opinion it is not right to consider informativeness from the viewpoint of the information theory. For example, it is known that image entropy increases with growing of image noises. Thus, images with high level of noises are selected.

3. AUTOMATIC PHOTO SELECTION

3.1 Ge neral workflow

In our algorithm we try to inherit positive trends from prior-art and overcome disadvantages of existing methods. The technique comprises three main steps: detection of low-quality photos and excluding of these photos from further processing, adaptive quantization of survived photos in time-camera plane, and selection the most appealing photos from each quantized rectangle (see fig. 2).

For detection of low-quality photos we use machine learning, namely complex classifier which comprises of consecution of one simple threshold classifier and two AdaBoost classifiers committees. Photos with exposure defects, blurred and images affected by strong JPEG artifacts are detected fast and confidently. For adaptive quantization we propose the method similar to median-cut color quantization algorithm [7]. For each quantized rectangle required number of photos is selected. Our hypothesis is the following: the most salient photo in sense of human vision model is more appealing for people. Until now, universal model of human vision did not exist, but pre-attentive vision model based on feature integration theory is well-known [8, 9]. We propose novel scheme for photos ranking based on visual salience. As a rule, presence of humans on a photo increases the appeal of the image and so we use face detection for strengthening of salience for photos with faces. For time optimization we work with downsampled versions of original photos during all steps of the algorithm.

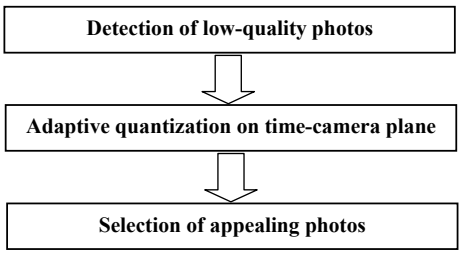


Figure 2. General workflow of proposed algorithm.

3.2 Detection of low-quality photos

Our investigations have revealed that up to 25 % of user's photos have serious defects such as blurriness, noise, compression artifacts, color misbalance as well as various types of exposure defects. Part of these images can be corrected but another part is irreparably defaced. Obviously such low-quality photos should be excluded from further processing. We consider fast algorithms only because of processing time should be extremely low. General scheme of our detection of low-quality photos algorithm is shown on fig. 3.

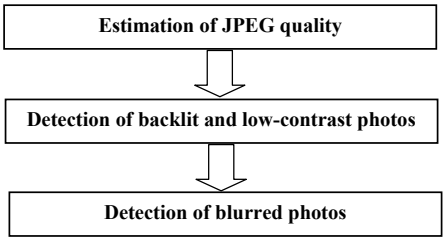


Figure 3. Scheme of detection of low-quality photos.

Sometimes photos are affected by color cast, for example, when indoor scene illuminated by an incandescent lamp is photographed. The simplest and fast color cast detection method is based on "gray world" assumption, i.e. averages in red, green and blue channels are equal. In [10] various information on a huge data set of 4.8 million photos is analyzed. Of those photos, only 74% have no dominant color that supports the general thesis about "gray world". In general it is hard to distinguish between intentional color cast and cast caused by color misbalance. Therefore exclusion of photos with any dominating color from collection is unwise.

Another important factor affecting image quality is noise. We could not find fast and reliable algorithm for noise level estimation that would provide adequate results for real-world photos. Approaches like described in [11] confuse textures with noise; error rate can reach 30% according to our experiments. It is inapplicable. More comprehensive approaches provide slightly better results but require significantly more time. Fortunately

modern DSCs have comprehensive noise reduction schemes and as a rule high noise presents on dark under-exposed photos, which can be detected easily.

Frequently compression artifacts look as noise. Absolute majority of user's photos are in JPEG format. In [12] a filter for deblocking and deringing was proposed. For adjustment of filter parameters the top left corner of the square of 3x3 quantization table of brightness channel is analyzed:

$$K = \frac{1}{9}\sum_{i,j=1}^{3} q_{i,j}.$$

Our research has shown that this metric correlates with visual assessment of JPEG images better than the compression ratio, which is strongly dependent upon the content. Figure 4 demonstrates plot *K* vs. compression ratio for high-quality photos and images with irritating artifacts level. *K* allows to separate good-quality JPEG images from JPEGs with strong artifacts. Our experiments have shown the best threshold for detection JPEG images with irritating artifacts is $K = 6.5$. The approach does not require any image content analysis and is extremely fast.

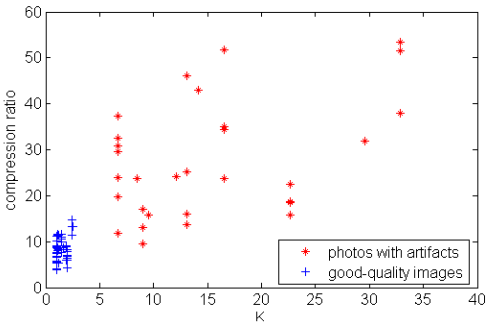


Figure 4. *K* vs. compression ratio.

There are a lot of image selection approaches where overexposed and underexposed photos are excluded from further processing. However modern DSCs have sophisticated algorithms; under-exposed and especially over-exposed photos happen rarely. More often photos damaged by backlighting can be found. Such photos have low dynamic range in dark areas. The comprehensive review of the problem was done in [13]. *Ibidem* the correction algorithm was described, but part of backlit images is irreparably defaced and correction produce unsightly outcomes.

Paper [13] describes features based on brightness histogram analysis and decision tree for adjusting correction parameters. We have repeated investigation of these features for our own test set and have found out that thresholding several of the features allows to detect backlit photos with probability 0.6-0.8. Combination of the simple classifiers and classifier for detection of low-contrast photos in AdaBoost committee as it is shown on fig. 5 allows to build classifier with high detection rate.

There are several AdaBoost algorithms which differ in approaches for optimization of weights w_i. Some realizations of these algorithms are capable to adjust thresholds of simple classifiers. We used GML AdaBoost Matlab Toolbox for feature selection and building of classifiers committee. GML AdaBoost Matlab Toolbox is a set of Matlab functions and classes, which implement Real AdaBoost, Gentle AdaBoost and Modest AdaBoost techniques. Real AdaBoost is the generalization of a basic AdaBoost algorithm first introduced by Fruend and Schapire [16]. Gentle AdaBoost [17] is a more robust and stable version of Real AdaBoost. In our case Gentle AdaBoost performs slightly better than Real and Modest AdaBoost [18].

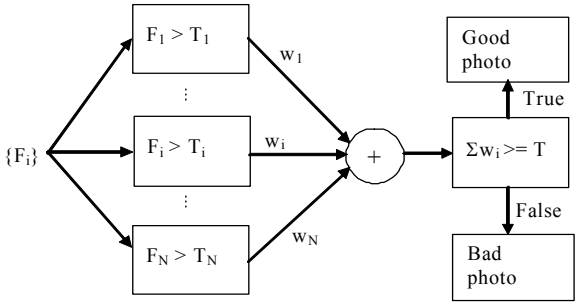


Figure 5. Scheme of AdaBoost classifiers committee.

The following features, which are calculated from brightness histogram *H* for image size *MxN* and color depth of brightness 8 bpp, were selected for classifiers committee:

S_1 / S_2 - ratio of tones in shadows to midtones, S_{11} / S_{12} - ratio of tones in first to second part of shadows, M_1 / M_2 - ratio of the histogram maximum in shadows to global histogram maximum, P_1 - location of the histogram maximum in shadows, *C* – global contrast, where

$$S_1 = \sum_{[0,85]} H(i)/(M \times N),\ S_2 = \sum_{(85,170]} H(i)/(M \times N),$$

$$S_{11} = \sum_{[0,42]} H(i)/(M \times N),\ S_{12} = \sum_{(42,85]} H(i)/(M \times N),$$

$$M_1 = \max_{[0,85]}(H(i))/\max_{[0,\ 255]}(H(i)),\ M_2 = \max_{(85,170]}(H(i))/\max_{[0,\ 255]}(H(i)),$$

$$P_1 = l \mid H(l) = \max_{[0,85]}(H(i)),$$

$$C = high - low,$$

$$low = \min(\min\{i \mid H[i] \ge H_0\}, \min\{i \mid \sum_{k=0}^{i} H[k] \ge C_0\}),$$

$$high = \max(\max\{i \mid H_R[i] \ge H_1\}, \max\{i \mid \sum_{k=i}^{1} H_R[k] \ge C_1\}),$$

where H_0, H_1 and C_0, C_1 are threshold values for histogram area and intensity correspondingly.

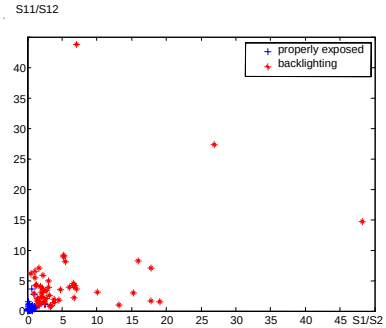


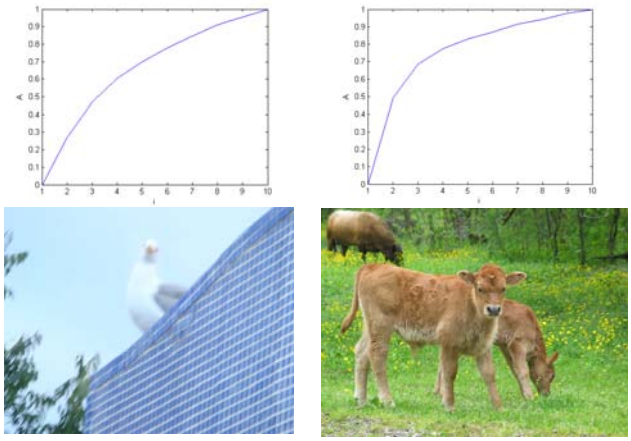
Figure 6. *S1/S2* vs. *S11/S12* for properly exposed and backlit.

The plot on fig. 6 demonstrates distribution of properly exposed and backlit photos on plane features *S1/S2* and *S11/S12*, backlit

photos can be detected with high probability based on these features. Our training set contains 188 photos with various exposure defects and 292 high-quality photos. Error rate on cross-validation test is about 0.055. Our testing set contains 1830 photos but only about 2% of the photos have low-contrast or affected by backlit. The number of False Positives (FP) is 10, number of False Negatives (FN) is 3. As a rule False Positives are night shots with wide dark background.

The proposed technique is very fast because only features calculated from histogram are used and classifiers committee is very simple in sense of computational complexity.

Blurriness is one of the most common image defects. It can be caused by mistake of focusing or camera shaking. Paper [14] proposes non-reference automatic sharpness level estimation, which is based on analysis of variations of edges histograms, where edge-images are produced by high-pass filters with various kernel sizes, array of integrals of logarithm of edges histograms characterizes photo sharpness. Let's consider that approach in more details. On the first step gray channel I of initial image is scaled to destination size according to viewing or printing conditions.



a) b)

Figure 7. Array $\{A_i\}$ for blurred (a) and sharp (b) photos.

AdaBoost classifiers committee is applied to detect out-of-focus photos. Further I is filtered by set from 10 high-pass filters with convolution kernels Z_i [-1 1], [-1 0 1], [-1 0 0 1] … [-1 0 0 0 0 0 0 0 0 1]:

$$E_i = I \otimes Z_i$$

For each abs(E_i) histogram He_i is calculated. The entropy characterizes the 'flatness' and 'peakedness' of histogram, but value of entropy strongly depends on total number and magnitude of edges that depends on photo content. It was proposed to normalize entropy by dividing by a number of edges for all edge magnitudes:

$$A_i = \sum_k \frac{-He_i(k)\log(He_i(k)+1)}{-He_i(k)} = \sum_k \log(He_i(k)+1).$$

Array $\{A_i\}$ varies for blurred and sharp photos. Figure 7 demonstrates $\{A_i\}$ for such photos as well as corresponding photos itself.

Thereupon several features $\{F_i\}$ to characterize $\{A_i\}$ were formulated and AdaBoost classifiers committee was constructed.

We propose to boost the committee by means of addition Crete's sharpness metric [15] as one more feature. Crete's non-reference sharpness estimation is based on the idea that a high variation between the original and the blurred image means that the original image was sharp whereas a slight variation between the original and the blurred image means that the original image had been already blurred.

In our committee the following features are applied:

$$F_1 = An_3 - An_2,\ F_2 = \sum_{i=2}^{11} An_i,\ F_3 = A_2,$$

where An_i elements are A_i normalized to [0, 1] range by means of dividing on max(A_i).

The fourth feature F_4 is similar to Crete's sharpness metric but is calculated for rows only:

$$F_4 = (SDI - SV_h)/SDI,$$

$$B_h = I \otimes LPF,\ DI = I \otimes HPF,\ DB_h = B_h \otimes HPF,$$

$$SDI = \sum_{\forall r,c} DI(r,c),$$

$$SV_h = \sum_{\forall r,c} (DI(r,c) - DB_h(r,c)) \times \frac{1}{1+e^{-100\times(DI(r,c)-DB_h(r,c))}},$$

where *LPF* is a low-pass filter with convolution kernel [1 1 1 1 1 1 1 1 1 1]/9, *HPF* is a high-pass filter with convolution kernel [1 -1], *(r,c)* are coordinates of image pixels.

Our training set contains 205 blurred photos and 311 high-quality photos. The error rate on cross-validation test is about 0.07. Our testing set of 1830 photos contains 171 blurred photos. The number of FP is 34, the number of FN is 10.

The processing time for classification of one photo on sharp or blurred classes is about 0.4 c. Whole classification cascade takes about 0.5 c for one photo, including downsampling to target printing or viewing size.

3.3 Time and camera-based quantization

In order to divide whole collection on M groups, where one photo is selected from each group, various quantization algorithms are intended. Time-based clustering or quantization provides satisfactory outcomes in event coverage sense in cases where all photos are photographed by just one camera only. Papers in prior-art do not discuss image selection from collection of photos captured by various cameras. Is it typical for users to collect photos from several cameras for one and the same event? We conduct user study to define whether amateurs collect images from several cameras for one storytelling application. Survey participants were asked three questions:

1) Do you have some photos for one event captured by several cameras at your paper or web photo album?

2) How many events from your album are captured by several cameras?

3) What are your arguments to collect photos from different cameras?

31 participants assisted in our survey. Albums of 83% of the users contain event photos captured by more than one camera and portion of such events is about 30%. Fig. 8 presents answers distribution for third question. So existence application of several cameras for one event is a widespread practice.

In case of usage of several cameras time quantization only is ineffective sometimes due to imperfection of cameras time synchronization and the same event can be presented on photos

with different time in EXIF. Usually mistiming is about several minutes, but can be equal to years. We propose quantization in 2D plane, where the first axis is time and the second axis is a camera name obtained from EXIF information. If EXIF is absent, then all images are merged into a separate virtual camera.

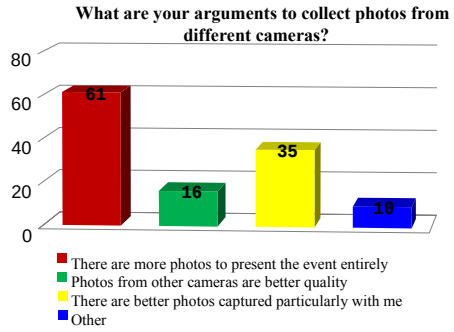


Figure 8. Motives to collect photos from different cameras.

Fig. 9 illustrates rule for time-camera plane creation. Cameras are sorted in ascending order according to the number of photos in collection. *L* is the time between the least and the biggest time for the camera with the largest number of photos. Yp_i coordinates on Ticks on *Camera* axis are calculated as follows:

$$Yp_i = \begin{cases} H \times (i+1)/2: & i \text{ is odd} \\ H \times (Nps + 1 - i/2): & i \text{ is even} \end{cases}$$

where *i* is index of camera, *Nps* is number of cameras,

$$H = 2 \times L / M .$$

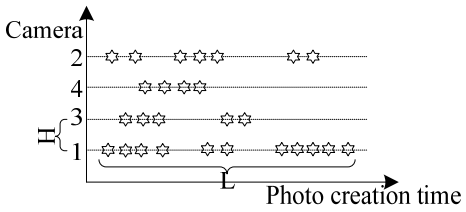


Figure 9. Time-Camera 2D plane

The general idea of the proposed quantization algorithm is the following: each group should contain approximately equal number of photos. The idea is similar to well-known Heckbert's color quantization technique [7]. In present time there is a trend to use modifications of the algorithm for plenty of tasks, for example for light probe sampling [23]. Taking inspiration from the median cut algorithm we can partition time-camera plane in the rectangular regions as follows:

1. Calculate bounding boxes for regions;
2. For region with the greatest number of photos divide on to sub-regions along the longest dimension such that the new subregions contain approximately equal number of photos;
3. If the number of current regions is less than number of the required groups, then return to step 1.

In addition we introduce limitation on minimal dimension of region: if the longest dimension is less than *Tr* then the region is not divided. The condition is intended for combining of duplicate photos in one group. The example of partitioning is shown on fig. 1. Further for each group one of the most appealing photo is selected.

3.4 Salient photo selection

We assume that the most appealing and relevant photo in a set is the most noticeable. This assumption leads us to conclusion that the most appealing photo is the most salient photo. General approach for construction of saliency map is described in Itti-Koch-Niebur [8, 9] papers. Usually saliency map is used for tasks related to pre-attention vision and scene analysis in order to determine the most valuable parts of the image or the scene, so-called regions of interest (ROI) [19]. Until now saliency map did not apply for comparison of images with each other; we propose the appropriate way for selection of the most salient photo.

The schema of saliency map building is shown on fig. 10. Every image in the set has red (r), green (g) and blue (b) channels. Intensity map is obtained as:

$$I = (r + g + b)/3 .$$

Four color channels *R, G, B, Y* are created from *r, g, b* in the following way:

$$R = r - \frac{g+b}{2}, \quad G = g - \frac{r+b}{2},$$

$$B = b - \frac{r+g}{2}, \quad Y = \frac{r+g}{2} - \frac{|r-g|}{2} - b.$$

For *I, R, G, Y* 8-level Gaussian pyramids are constructed using Gauss separable filter with convolution kernel [1 5 10 5 1]. From intensity map 8-level Gabor pyramids for different orientations $\theta \in \{0, 45, 90, 135\}$ are created to obtain local orientation information. We compute 42 feature maps using center-surround difference:

$$I(c,s) = |I(c) - I(s)|$$
$$RG(c,s) = |(R(c) - G(c)) - (G(s) - R(s))|$$
$$BY(c,s) = |(B(c) - Y(c)) - (Y(s) - B(s))|$$
$$O(c,s,\theta) = |O(c,\theta) - O(s,\theta)|$$

where $c \in \{2,3,4\}$ and $s = c + \delta, \delta \in \{2,3\}$.

All feature maps are normalized using local maximum technique and combined into conspicuity maps using across-scale addition:

$$\bar{I} = \sum_{c=2}^{4} \sum_{s=c+3}^{c+4} N(I(c,s))$$

$$\bar{C} = \sum_{c=2}^{4} \sum_{s=c+3}^{c+4} [N(RG(c,s)) + N(BY(c,s))],$$

$$\bar{O} = \sum_{\theta \in \{0,45,90,135\}} N\left(\sum_{c=2}^{4} \sum_{s=c+3}^{c+4} N(O(c,s,\theta))\right),$$

where N() is a normalization operator which increases strong peaks and decrease noise.

Normalization operator consists of two parts. In the first part Gaussian filter is applied to the image in order to decrease noise. In the second part average local maximum is computed and the whole image is multiplied by the difference of the maximum value on the image and local maximum value. This operation helps to prevent strong but individual peaks and also helps not to take into account such things as very bright background.

So, we obtain the following conspicuity maps: $\bar{I}$ for intensity, $\bar{C}$ for color, $\bar{O}$ for orientation. Conspicuity maps are summed with specific weights into final image which is called saliency map:

$$S = \frac{weightI \cdot N(\bar{I}) + weightC \cdot N(\bar{C}) + weightO \cdot N(\bar{O})}{weightI + weightC + weightO} .$$

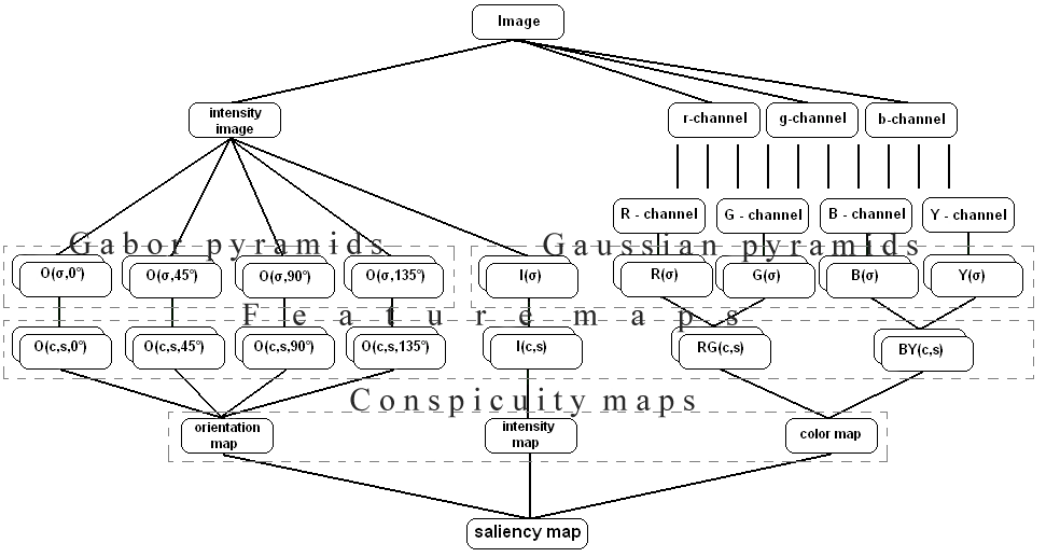


Figure 10. The schema saliency map building.

The main problem is to find right weights cause due to normalization different conspicuity maps have different contribution to final result. The majority of the previous works consider to sum conspicuity maps in equal proportions which in our opinion is not completely right. To solve this problem we considered to make an experiment. This experiment as input data has a number of images (normally from 30 to 50 images). For each picture in the set the most salient regions were marked by several experts. In order to determine the best weights we were finding maximum of the following function using simplex algorithm:

$$\sum_{p \in ROI} S(p) \to \max$$

where ROI are noted areas on the image, $p \in ROI$ and $S(p) \geq S_{\max} / 4$.

Mathematical expectations of weights were calculated after finding values for every image in the set.

Experiment has shown that weights locate in the following ranges:

$$weightI = 0.2..0.5,\ weightC = 0.4..0.6,\ weightO = 0.2..0.5.$$

Specific values depend on the person and his perception of the surrounding world, his preferences and features, everyone can choose what he or she likes more. Fig. 11 demonstrates the photo and its conspicuity maps as well as the final saliency map (weightI = 0.5, weightC = 0.25, weightO = 0.3).

The last step was to find a criterion which ranges photos in the set and gives clear answer what photo is the most salient among others. "Saliency Index" *SI* is counted as following:

$$SI = \sum S(x,y)/(w \cdot h)$$

where $S(x,y) \geq S_{\max} / 4$, w is image width, h is image height.

This criterion was applied to different photo sets and it was found that itproduced appropriate results. Thr example of how photos are ranged by *SI* can be seen on fig. 12. The photos "camel" and "boy" have close values of *SI* but camel's *SI* is a little bit greater.

Algorithm works less than 1 second for color images with size 500 x 500. Processing time can be decreased considerably due to parallel calculation on GPU as it is described in [20].

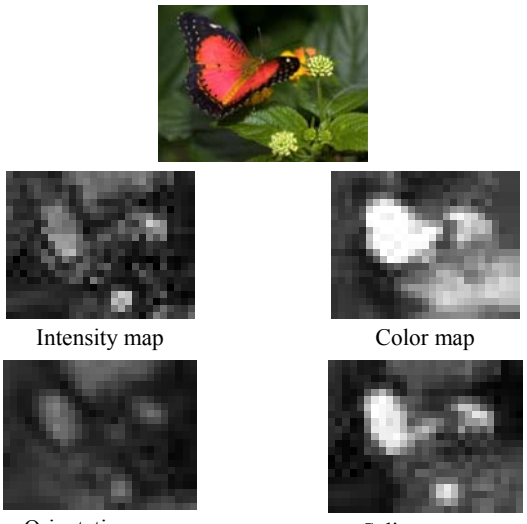


Figure 11. Photo and its conspicuity and saliency maps.

3.5 Fa ce detection

Images of people prevail in a lot of amateur photo collections. This kind of photos attracts attention more than images without humans. A face detection algorithm can be used for search of human presence. Face processing is a rapidly expanding area and a lot of researches have been conducted in recent years. One of the acknowledged algorithms for face detection is the one developed by Viola and Jones [21]. The Intel OpenCV library provides an efficient implementation of the Viola-Jones face detector.

We analyzed implementation of Viola-Jones algorithm in OpenCV library for typical user's photos with sizes from 4 to 6 Mpix and found out the following main disadvantages: average number of FP is more than 3; the processing time is more than 5 seconds for modern PC. The face detection outcomes of initial

OpenCV version is shown on the photo “boy” on fig. 12. The face is detected correctly but two places were detected erroneously.

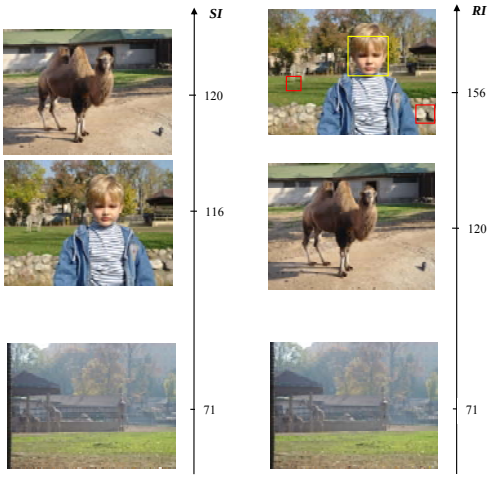


Figure 12. Ranking results by *SI* and summary index *RI*.

We have carried out some modifications to improve the algorithm efficiency. Specifically, conducted modifications add up to the following: the optimal image downsampling is applied at preprocessing step for speed increasing; optimization of search region using color information is performed. Detailed information about the experiment conditions, obtained data and comparative results were described in [22]. There are no False Positives on photo “boy” after our modifications.

Detection time is about 1 second; now it is greater than it is necessary of our task. In particular it is connected with extra conversion to internal OpenCV structures and other programming issues. We expect to reach 0.2-0.3 s time for processing of image with size 850x640 on PC.

3.6 Ph otos ranking

We propose to combine saliency index *SI* and face detection outcomes for calculation of summary appealing index *RI* as follows:

$$RI = SI + w \times \sqrt{NF} \ ,$$

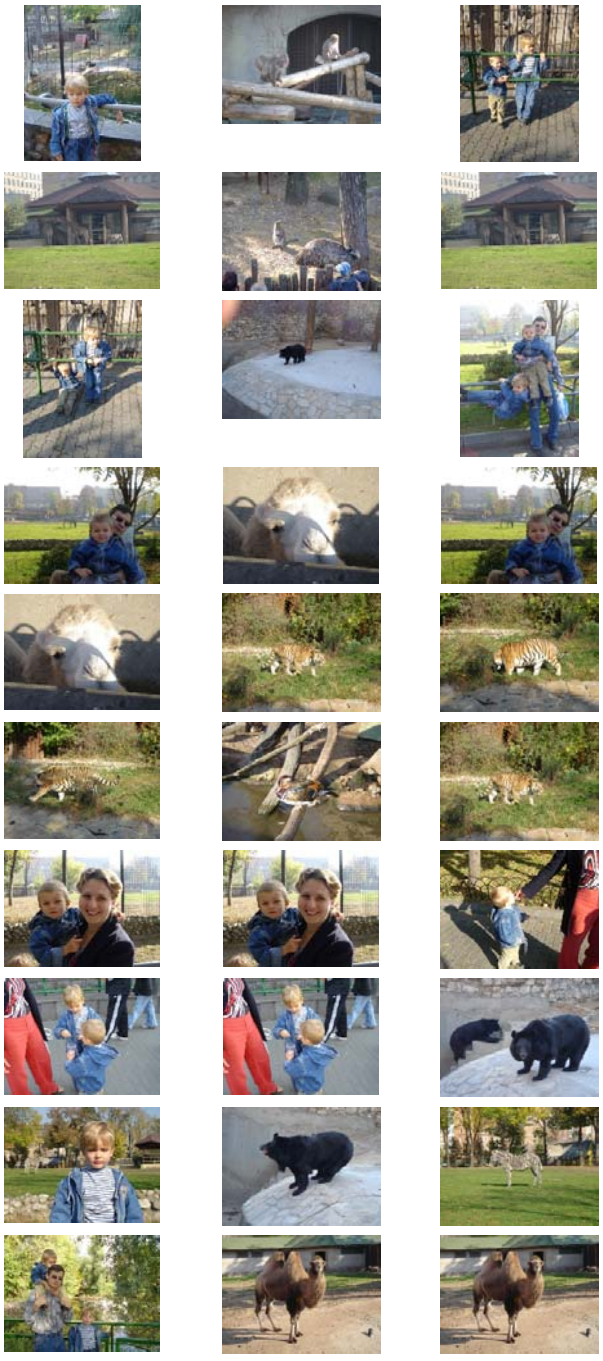
where *NF* is the number of detected faces, *w* is weight.

The heuristic formula and preferable *w*=25 value were obtained during plenty of experiments. The final ranking results by *RI* are shown on fig. 12. Accordingly photo “boy” is selected as the most appealing from the group of three photos.

4. RESULTS AND DISCUSSION

The set of 30 photos captured by two cameras is shown on fig. 1.

Let 10 photos should be selected. At the first stage low-quality photos are detected. For given set four poor images were detected. These photos are blurred actually and they are excluded from further processing. On fig. 1 excluded images are crossed out by solid red line. Next stage is median-cut-like quantization on time-camera plane. The 10 groups, which are the result of quantization, are outlined by blue dash line. The final stage is selection of the most appealing photo among images of each group. The selected 10 photos are outlined by red dot line. The owner of the collection evaluates achieved outcomes as accurate: 6 photos coincide with manual selection by the expert (# 1, 8, 10, 21, 28, 30 on fig. 1), 3 photos are considered as acceptable (# 13, 16, 22) and only one photo (# 7) the expert counts as uninteresting.



a) b) c)

Figure 13. Results of photo selection by proposed technique (a), MS AutoCollage (b) and random selection (c).

Unfortunately the majority of existing solutions for automatic photos selection is inaccessible for testing, but we had possibility to compare our selection results with outcomes of MS Research AutoCollage (ver. 1.1.2009.0130) and with simple random selection. The function of AutoCollage software application is automatic creation of photo collage and the first stage is selection

of photos from a collection. Two photos selected by AutoCollage (# 3, 26) are blurred and expert counts their as inacceptable. Only two photos (# 19, 21) coincide with manual selection; other 6 photos (#6, 12, 13, 17, 22, 27) are considered as acceptable.

So ground truth is manual selection by the owner of photo collection and his/her expertise. In our opinion the number of inacceptable photos in selected set is principal criterion. For estimation of efficiency of proposed technique we have processed 5 sets of photos; each set contains 30 photos. Table 1 reflects obtained results.

TABLE 1 RESULTS OF AUTOMATIC SELECTION FOR 5 SETS.

		Set 1	*Set 2*	*Set 3*	*Set 4*	*Set 5*	**Sum**
Proposed	Agree with expert	6	5	6	5	7	**29**
	Acceptable	3	4	4	4	2	17
	Inacceptable	1	1	0	1	1	**4**
AutoCollage	Agree with expert	2	2	2	6	5	**17**
	Acceptable	6	7	7	0	4	24
	Inacceptable	2	1	1	4	1	**9**
Random	Agree with expert	2	2	3	4	4	**15**
	Acceptable	5	5	4	2	5	21
	Inacceptable	3	3	3	4	1	**14**

Both tested solutions demonstrate high efficiency and good coverage of event. Random selection demonstrates serious errors. Automatic selection is capable to improve creation of photo album for media and entertainment applications such as photobook and slide-show.

Sometimes AutoCollage selects low-quality images whereas proposed algorithm has no such drawback. In AutoCollage number of FP on face detection stage is high enough. The number of FP in our face detection module is less in 2 times with preserving the same detection rate. As regards processing time, AutoCollage spends about 1.1 s per image. Our method is a little bit slower; it spends about 1.4 s per image. Bottleneck is face detection module. We expect to speed up face detection in the future.

5. REFEREN CES

1. P.Obrador, N.Moroney, “Automatic Image Selection by means of a Hierarchical Scalable Collection Representation”, Proc. of SPIE-IS&T Electronic Imaging, 2009.
2. J.C. Platt, M. Czerwinski, B.A. Field, “PhotoTOC: Automatic Clustering for Browsing Personal Photographs”, Proc. IEEE Pacific Rim Conf. Multimedia, pp. 6-10, 2003.
3. A.Graham, H.Garcia-Molina, A.Paepcke, T.Winograd, “Time as essence for photo browsing through personal digital libraries”, Proc. of the 2nd ACM/IEEE-CS conference on Digital libraries, 2002.
4. D.F.Huynh, S.M.Drucker, P.Baudisch, C.Wong, “Time Quilt: Scaling up Zoomable Photo Browsers for Large, Unstructured Photo Collections”, CHI'05 extended abstracts on Human factors in computing systems, 2005.
5. J.Chen, W.Chu J.Kuo, C.Weng, J.Wu, “Tiling Slideshow: An Audiovisual Presentation Method for Consumer Photos”, In Proc. of ACM Multimedia Conference, pp. 36-45, 2007.
6. C.Rother, L.Bordeaux, Y.Hamadi, A.Blake, “Autocollage”, ACM Transaction on Graphics, SIGGRAPH-2006, vol 25, pp. 847-852, 2006.
7. P.Heckbert, “Color Image Quantization for Frame Buffer Display”, SIGGRAPH-1982, ACM Press. pp. 297-207, 1982.
8. L.Itti, C.Koch, E.Niebur, “A model of saliency-based visual attention for rapid scene analysis”, IEEE Transactions on Pattern analysis and machine intelligence, Vol. 20, No. 11, pp. 1254-1259, 1998.
9. L.Itti, C.Koch, “A saliency-based search mechanism for overt and covert shifts of visual attention”, Vision Research 40, pp 1489-1506, 2000.
10. D.Wueller, R.Fageth,”Statistic Analysis of Millions of Digital Photos”, Proc. of SPIE/IS&T Electronic Imaging, 2008
11. P.Meer, J.M.Jolion, A.Rosenfeld, “A Fast parallel algorithm for blind estimation of noise variance”, IEEE Tran. on Pattern Analysis and Machine Intelligence, Vol. 12, No. 2, 1990.
12. A.Foi, V.Katkovnik, K.Egiazarian, “Pointwise Shape-Adaptive DCT for High-Quality Denoising and Deblocking of Grayscale and Color Images”. IEEE Transaction on Image Processing, v.16, No 5, 2007.
13. I.Safonov, “Automatic correction of amateur photos damaged by backlighting”, Graphicon-2006.
14. I.V.Safonov, M.N.Rychagov, K.M.Kang, S.H.Kim, “Adaptive sharpening of photos”, Proc. of SPIE-IS&T Electronic Imaging, 2008.
15. F.Crete, T.Dolmiere, P.Ladret, M.Nicolas, “The Blur Effect: Perception and Estimation with a New No-Reference Perceptual Blur Metric”, Proc. of SPIE-IS&T Electronic Imaging, 2007.
16. Y.Freund, R. Schapire. Experiments with a new boosting algorithm, Int. conf. on Machine Learning, pp. 148-156, 1996.
17. J.Friedman, T.Hastie, R.Tibshirani, "Additive logistic regression: A statistical view of boosting", The Annals of Statistics, 38(2), pp. 337-374, 2000.
18. A.Vezhnevets, V.Vezhnevets, Modest AdaBoost – teaching AdaBoost to generalize better, Graphicon, pp.322-325, 2005.
19. C.M.Privitera, L.W.Stark, “Algorithms for defining visual regions-of-interest: comparison with eye fixations”, IEEE Transactions on Pattern Analysis and machine intelligence, vol. 22, no. 9, pp. 970-982, 2000.
20. P.Longhurst, K.Debattista, A.Chalmers, “A GPU-based saliency map high-fidelity selective rendering”, Proc. of ACM AFRIGRAPH, 2006.
21. P.Viola, M.Jones, “Robust Real-time Object Detection”, Compaq Technical report, 2001
22. M.A.Egorova, A.B.Murynin, I.V.Safonov, “An improvement of face detection algorithm for colour photos”, Proc. of Pattern Recognition and Image Analysis, 2008.
23. P.Debevec, “A Median Cut Algorithm for Light Probe Sampling”, ACM Transaction on Graphics: SIGGRAPH-2005 posters, 2005.

About the authors

Ekaterina Potapova is a 5 year student at National Nuclear Research University (Moscow Engineering Physics Institute, MEPhI).

Marta Egorova received her MS degree in cybernetics from MEPhI in 2008. At present time she is a post-graduate student of MEPhI.

Ilia Safonov received his MS degree in automatic and electronic engineering from MEPhI in 1994 and his PhD degree in computer science from MEPhI in 1997.

Восстановление формы страницы текста для коррекции геометрических искажений

Ramiz Zeynalov, Alexander Velizhev, Anton Konushin
Department of Computational Mathematics and Cybernetics
Moscow State University, Moscow, Russia
ramiz.zeynalov@gmail.com, avelizhev@graphics.cs.msu.ru, ktosh@graphics.cs.msu.ru

Аннотация

В последнее время цифровые камеры получили широкое распространение. Поэтому их часто используют для быстрого фотографирования текста. При этом тексты на таких фотографиях плохо подходят для распознавания и не очень удобны для чтения. Основная проблема – это непрямые строки текста. Существующие на сегодняшний день системы оптического распознавания текста не могут дать удовлетворительного результата на подобных входных данных без соответствующей предобработки. Наибольшую сложность в этой предобработке представляет геометрическая коррекция. В данной статье предлагается развитие предложенного ранее метода исправления описанных выше искажений, основанного на восстановлении поверхности страницы по форме найденных на изображении строк текста.

Ключевые слова: *строки текста, распознавание текста, улучшение качества текста.*

1. ВВЕДЕНИЕ

В последнее время цифровые камеры получили широкое распространение – они встраиваются в сотовые телефоны и КПК. Поэтому их часто используют для быстрого фотографирования текста. При этом тексты на таких фотографиях плохо подходят для распознавания и не очень удобны для чтения.

Предлагается развитие метода из работ [18] и [19], который позволял решить частную задачу со следующими ограничениями:

1. Искажения формы происходят только в направлении, перпендикулярном строкам текста
2. Перспективных искажений нет
3. Главная оптическая ось камеры проходит через центр изображения текста по вертикали

В этой статье описано развитие метода, в котором удалось отказаться от всех вышеперечисленных ограничений, что существенно расширило область применения метода.

1.1 Постановка задачи

Дано изображение текста, которое может быть подвергнуто следующим типам искажений:

- Неравномерное освещение и блики
- Перспективные искажения
- Непланарность исходного текста

Требуется исправить перечисленные типы искажений, чтобы результат работы OCR был лучше и визуально строки текста стали прямыми, одного размера и ориентированы горизонтально.

2. ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ

2.1 Регрессия искривленных линий

Этот метод используется для коррекции геометрических искажений на сканированных изображениях, возникающих вследствие деформации поверхности книги в окрестности границы между страницами.

Здесь отдельно обрабатывается затенённая область и незатенённая область. Концы строк с затенённой области аппроксимируются полиномами второй степени ([5]) и выпрямляются с учётом этого полинома. После выпрямления концов строк они объединяются с прямыми фрагментами строк с незатенённой области. Более подробно о методе можно прочитать в [2].

Достоинства:

- Простота реализации
- Высокая скорость работы

Недостатки:

- Исправляются искажения только в окрестности границы страницы

2.2 Использование патча Безье

В этом методе используется понятие скелета многоугольной фигуры. Для решения задачи строится внешний скелет для межстрочных интервалов, после чего производится фильтрация скелета, затем строится патч Безье. С помощью полученного патча итерационным методом производится выпрямление строк. Подробнее о методе можно узнать из [8].

Достоинства:

- Высокая скорость работы
- Простота реализации

Недостатки:

- Реальное искажение не всегда можно приблизить плоской деформацией

2.3 Робастное приближение формы кривых строк текста

В данном подходе осуществляется поиск связанных компонент, затем для каждого символа находится локальная линия, на которой лежит строка, содержащая данный. Для каждого найденного символа вычисляется расстояние между строками, затем оно усредняется и считается постоянным

для всего текста. Используя найденное среднее расстояние между строками и расстояния между строками на изображении, строится поверхность книги, с использованием которой выполняется коррекция изображения текста. Более подробно узнать о методе можно в [3].

Достоинства:

- Корректируются и геометрические, и перспективные искажения

Недостатки:

- Высокая вычислительная сложность
- Необходимы данные о начертании символов (для использования алгоритма RAST)

2.4 Глобальная оптимизация формы поверхности

Для описания поверхности текста используется полигональная модель. Сначала строятся векторные поля – направление текста, направление расположения строк, нормали к поверхности, и учитываются параметры камеры. Затем строится функционал, который учитывает ограничения на ортогональность введённых полей в каждой точке поверхности и на гладкость результирующей поверхности. Этот функционал итерационно оптимизируется. Затем, каждая грань полигональной модели поверхности независимо отображается на скорректированное изображение, после чего результаты склеиваются с учётом условия непрерывности полученной плоской поверхности текста. Подробнее о методе можно узнать из [4].

Достоинства:

- Решает широкий спектр проблем (поиск областей с текстом, поворот, перспективная и геометрическая коррекция текста)

Недостатки:

- Сложность в использовании (много ручных параметров, задача определения оптимальных значений которых нетривиальна)
- Низкая производительность по сравнению с другими методами
- Хорошо исправляются лишь перспективные искажения

2.5 Выводы

Все рассмотренные методы обладают своими достоинствами и недостатками. Как правило, каждый метод, решающий поставленную задачу, находит компромисс между эффективностью и универсальностью. Из методов, которые обладают высокой производительностью, качественнее всех работает метод, использующий патч Безье для аппроксимации плоской деформации изображения.

3. КРИТЕРИИ ОЦЕНКИ КАЧЕСТВА РЕШЕНИЯ

С точки зрения системы оптического распознавания символов качество результата можно оценить через увеличение процента правильно распознанных символов после выполнения коррекции. Однако различные системы распознавания работают по-разному и используют разную информацию для своей работы. Таким образом, использовать систему распознавания текста в качестве критерия качества некорректно. Поэтому в качестве критерия будем использовать визуальное качество результата, для этого формализуем это понятие, учитывая, что человеку удобнее читать прямой текст, ориентированный горизонтально и под прямым углом [17].

4. ОПИСАНИЕ МЕТОДА

Алгоритм коррекции деформаций состоит из нескольких последовательных шагов:

1. Подавление неравномерного освещения и бинаризация
2. Объединение символов в строки
3. Анализ геометрии строк и построение карты глубины
4. Построение поверхности по карте глубины
5. Проецирование с учётом полученной поверхности

4.1 Подавление неравномерного освещения и бинаризация

Целью данного шага является выравнивание освещенности на изображении в целом. Для этого используются два метода – деление изображения на его низкочастотную составляющую и ретинекс [10].

4.2 Объединение символов в строки

Для упрощения задачи поиска строк текста, отдельные символы и слова объединяются в связные компоненты, соответствующие строкам текста. Это осуществляется путем последовательного применения операций математической морфологии ([18, 19]).

4.3 Анализ геометрии строк и построение карты глубины

Сначала производится поиск связных компонент на бинарном изображении, полученном на предыдущем шаге. Для этого используется алгоритм [13]. Области классифицируются как шум, иллюстрации и строки. По строкам текста строятся кривые, описывающие их форму. Полученная линия хорошо описывает форму строки во внутренних точках, а окрестности границ строк обрабатываются отдельно. Далее, для каждой страницы вычисляются линии левой и правой границ. Для нахождения этих прямых используется RANSAC ([15]). По полученным границам рассчитывается матрица гомографии ([16]) из четырёхугольника, образованного пересечением этих границ с верхней и нижней границами изображения. После чего определяется область действия этого преобразования как четырёхугольник, в котором лежат все точки всех строк. В результате получается изображение без перспективных искажений. Кроме того, искажения из-за неплоской поверхности текста имеют место только в вертикальном направлении относительно плоскости страницы. После того, как область с текстом найдена, осуществляется поиск горизонтальной линии, в окрестности которой эти искажения минимальны. Полученные значения аппроксимируются прямой линией и находится значение ординаты, при которой эта прямая пересекает 0. Далее, с учётом формы строк, их желаемого расположения и оптического центра по вертикали, строится форма поверхности – глубина поверхности для каждого пиксела

входного изображения, соответствующего точек на строке. Затем находится зависимость между искажением формы строки и глубиной соответствующей точки строки. Подробное описание этих действий можно найти в статьях ([18, 19]).

4.4 Построение поверхности по карте глубины

Необходимо построить поверхность во всех точках таким образом, чтобы полученная поверхность была гладкой. Кроме того, в точках, в которых она уже была определена, возможно, с ошибками, она была близка к точкам, полученным ранее. Определить поверхность в остальных точках надо для того, чтобы итоговое преобразование было непрерывным, чтобы нивелировать ошибки для отдельных строк и преобразовать все области, а не только области, соответствующие центрам строк.

4.5 Проецирование с учётом полученной поверхности

Для того чтобы полученное изображение было гладким, применяется обратное преобразование к каждой точке изображения-результата, в результате чего вычисляется цвет рассматриваемого пиксела.

5. РЕЗУЛЬТАТЫ

Тестирование производилось на нескольких наборах данных, которые отличались сложностью искажений и разрешением.

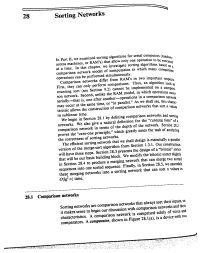
До коррекции	После коррекции
	
а)	б)

Рис. 1: Пример работы метода: а) исходное изображение, б) скорректированное изображение.

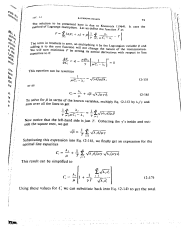
До коррекции	После коррекции
	
а)	б)

Рис. 2: Пример работы метода: а) исходное изображение, б) скорректированное изображение.

6. СРАВНЕНИЕ

Для сравнения с предложенным методом был выбран метод, основанный на построении патча Безье (из работы [8]). Выбор обусловлен тем, что метод, использующий патч Безье, согласно проведённому обзору, лучший на сегодняшний день. Сравнение производилось на стандартной для решаемой задачи тестовой базе IUPR [9], критерии сравнения были описаны в разделе “Критерии оценки качества решения”. Ниже на рисунках 3-5 изображены диаграммы с результатами сравнения ошибок неискривлённости, ориентации и перспективы для предложенного метода и метода, основанного на моделирования плоского искажения патчем Безье и скелете изображения.

Из рисунка 3 видно, что по критерию неискривлённости предложенный метод превосходит патч Безье.

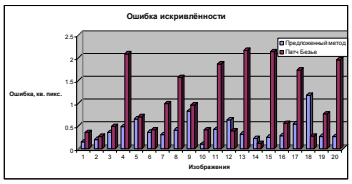


Рис. 3. Ошибка искривлённости (отклонение от прямой)

Из рисунка 4 видно, что ошибка ориентации чаще всего получается близкой к нулю.

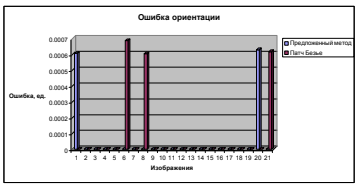


Рис. 4. Ошибка ориентации (отклонение от горизонтали)

Из рисунка 5 видно, что ошибка перспективы у обоих методов незначительная.

Рис. 5. Ошибка перспективы

7. ПРОИЗВОДИТЕЛЬНОСТЬ

Скорость работы приложения зависит от разрешения входного изображения и от некоторых параметров. Ниже приведена таблица (таблица 1) зависимости средней скорости работы от разрешения входного изображения. Измерения производились на компьютере средней конфигурации.

Разрешение, *Мпикс*	**Время работы,** *секунды*
1.3	1-1.5
6	5-6
8	7-10

Таблица 1. Производительность предложенного метода

8. ЗАКЛЮЧЕНИЕ

В данной работе предложен новый метод, позволяющий исправлять геометрические и перспективные искажения строк текста. Произведено сравнение с одним из существующих методов на стандартной тестовой базе. Метод для сравнения был выбран с тем же порядком производительности, что и предложенный.

9. ЛИТЕРАТУРА

[1]Владимир Вежневец. Задача восстановления 3D формы по закраске [HTML] (http://cgm.graphicon.ru/content/view/59/62/)

[2]Zheng Zhang, Chew Lim Tan. Correcting Document Image Warping Based on Regression of Curved Text Lines // Proc. ICDAR-2003, 2003, pp.589-563

[3]Adrian Ulges, Christoph H. Lampert, Thomas M. Breuel. Document Image Dewarping using Robust Estimation of Curled Text Lines // Proc. ICDAR-2005, 2005, pp.1001- 1005

[4]Adrian Ulges, Christoph H. Lampert, Thomas M. Breuel. Unwarping Images of Curved Documents Using Global Shape Optimization // Proc. International workshop on Camera-based Document Analysis and Recognition, 2005, pp.25-29

[5]Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков. Численные Методы. Бином, 2003, Москва

[6]T. M. Breuel. Robust least square baseline finding using a branch and bound algorithm // Proc. of SPIE/IS&T 2002 Document Recognition & Retrieval IX Conf. (DR&RIX), Jan. 2002, pp. 20–27

[7]Li Zhang, Chew-Lim Tan. Warped Document Image Restoration Using Shape-From-Shading and Physically-based Modeling // Proc. WACV, 2007, pp.29-32

[8]Антон Масалович, Леонид Местецкий. Использование патча Безье для аппроксимации искажения изображений текстовых документов // Proc. Graphicon-2007, 2007, pp.171-174

[9]Faisal Shafait, Thomas M. Breuel. Document Image Dewarping Contest // 2nd Int. Workshop on Camera-Based Document Analysis and Recognition, CBDAR'07, Brazil. Sep. 2007. pp 181-188.

[10]Zhixi Bian, Yan Zhang. Retinex Image Enhancement Techniques // Final Project Report for EE264

[11]John W. Tukey. Exploratory Data Analysis // Addison-Wesley, Reading, MA. 1977.

[12]The Open Computer Vision Library [HTML] (http://www.opencv.org/)

[13]Кормен, Лейзерсон, Ривест, Штайн. Алгоритмы. Построение и анализ // Издательский дом "Вильямс", Москва, 2005, стр. 635

[14]Шапиро, Стокман. Компьютерное зрение // Бином, Москва, 2006, стр. 172

[15]Антон Конушин. Устойчивые алгоритмы оценки параметров модели на основе случайных выборок [HTML] (http://cgm.computergraphics.ru/content/view/47)

[16]Антон Конушин. Геометрические свойства нескольких изображений [HTML] (http://www.cgm.computergraphics.ru/content/view/141)

[17]Legge GE, Pelli DG, Rubin GS, Schleske MM. Psychophysics of reading. I. Normal vision. // Vision Res., 1985;25: pp.239–252

[18]Зейналов Р. Ш., Геометрическая коррекция фотографий документов // Proc. Ломоносов-2008, 2008, pp. 60

[19]Зейналов Р. Ш., Геометрическая коррекция фотографий документов // Proc. Graphicon-2008, 2008, pp. 258-261

10. ОБ АВТОРАХ

Зейналов Рамиз Шакирович – выпускник факультета вычислительной математики и кибернетики Московского Государственного Университета им. М. В. Ломоносова.

E-Mail: ramiz.zeynalov@gmail.com

Велижев Александр Брониславович – к. т. н., научный сотрудник Лаборатории Компьютерной Графики и Мультимедиа факультета вычислительной математики и кибернетики Московского Государственного Университета им. М. В. Ломоносова.

E-Mail: avelizhev@graphics.cs.msu.ru

Конушин Антон Сергеевич – к. ф.-м. н., научный сотрудник, руководитель группы компьютерного зрения Лаборатории Компьютерной Графики и Мультимедиа факультета вычислительной математики и кибернетики Московского Государственного Университета им. М. В. Ломоносова.

E-Mail: ktosh@graphics.cs.msu.ru

Document images geometrical distortions correction using text lines shape extraction

Abstract

By reason of PDA and Smartphones with integrated digital cameras became widespread, people have started to use it to capture images from broadside of a book. Naturally the text on this-way captured images is suffered by both geometrical and perspective distortions because of non-planar geometry shape of the book broadside. The OCR methods couldn't get text from such images with satisfactory accuracy. In this article an enhancement of previous method of geometric correction of such images is described. There are some suggestions in new method retracted. Using this method the recognition accuracy and text lines straightness are improved. This method is based on the surface shape from text lines shapes extraction.

About the authors

Ramiz Zeynalov is graduate student at Moscow State University, Faculty of Computational Mathematics and Cybernetics. Contact E-Mail: ramiz.zeynalov@gmail.com

Alexander Velizhev is researcher at Moscow State University, Faculty of Computational Mathematics and Cybernetics. Contact E-Mail: avelizhev@graphics.cs.msu.ru

Anton Konushin is researcher at Moscow State University, Faculty of Computational Mathematics and Cybernetics. Contact E-Mail: ktosh@graphics.cs.msu.ru

A Comparison Of Suitable Object Recognition Methods For Mobile Voiced Visual Assistant

Vassili Kovalev, Igor Safonov
Biomedical Image Analysis Group
United Institute of Informatics Problems, Minsk, Belarus
vassili.kovalev@gmail.com, safonov@tut.by

Abstract

The objective of this work is to find an optimal object recognition method for Mobile Voiced Visual Assistant (MVVA). MVVA is under development in the Biomedical Image Analysis Group of the Institute. It's aimed to assist visually-impaired people in recognition and audio interpretation of surrounding scenes and objects in real time. In this paper we consider some applicable recognition methods based on color co–occurrence matrices.

Keywords: *object recognition, co–occurrence matrix, SVM, PCA, Mobile Voiced Visual Assistant.*

1. INTRODUCTION

The subject of our study is to find an optimal recognition method for future use in developing of Mobile Voiced Visual Assistant (MVVA). MVVA is aimed to assist visually-impaired people in recognition and audio interpretation of surrounding scenes and objects in real time (indoors or outdoors).

Let us remark that the use of special devices is not planned here. It is anticipated that the software will work as an application on any portable device with a camera. Such devices as ultra-mobile PC, netbooks and mobile phones are implied. These are resulted in some specific requirements to the software.

First of all, the application should work with images of low or medium quality.

The second requirement is the recognition algorithm's stability. The algorithm must provide good results under different environment. Along with this, in some algorithms the correct choice of parameters might be crucial for obtaining good results. Therefore an important step of the analysis is to estimate these parameters. In order to achieve this, we have developed and implemented a plan of experimental research the results of which are reported with this study.

The structure of the article is as follows. Section 2 reviews input data, hardware and software tools. In the Section 3 research stages themselves are described. The final section summarizes the results of this experimental investigation.

2. INITIAL DATA, THE HARDWARE AND SOFTWARE TOOLS

As a development environment we have used R, a system for statistical computing and graphics [1]. It consists of a language plus a run-time environment with graphics, a debugger, access to certain system functions, and the ability to run programs stored in form of R script files. The add-on package e1071 was employed here too.

Input data were 24-bit images of 320×240 pixels in size (Fig.1). All the images came from Logitech QuickCam Pro 9000 web camera. We consider 73 images. A variety of the data can be grouped into certain classes. Eight classes are considered in framework of this study that conditionally categorized into: «bag», «chair», «cup», «doorway», «flower», «phone», «wardrobe» and «window».

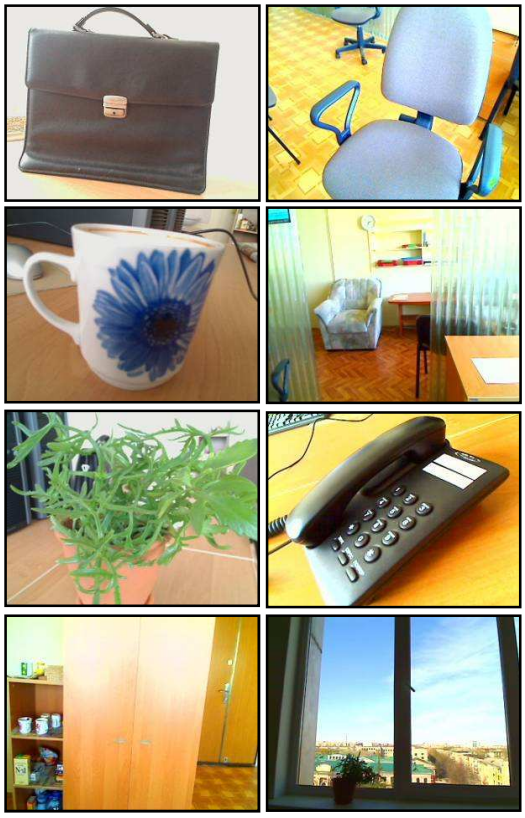


Figure 1: Example of input images.

Pictures of each class are split into a training and test sets. If *N* is a number of images from some class, then training set contains *N/2* elements, and test set contains the rest of *N-N/2* elements. We utilize the re-sampling techniques to generate test and training sets which are the random subsets of the original image collection of each type.

Image descriptors were created by way of a vectorization of color co-occurrence matrices [2,3]. It should be noted that for 24-bit color RGB images the corresponding color co-occurrence matrix can be very large. We reduce color space from 24-bit down to 8-bit using the common quantization scheme known as “3-3-2”.

Thus, all further manipulations were carried out in the system for statistical computing and graphics R with the whole set of 73 image files. The files contain pre-calculated co-occurrence matrices of 8-bit images. At every stage of experimentation the

resultant estimates reported in this paper are the mean values computed over 300 iterations for reliability. On each iteration step, the original data set is randomly re-sampled with replacement to generate independent replications.

3. RESEARCH STAGES

We will subsequently pass through the four stages of the assessment changing the conditions of experiments on:

- Selection of optimal parameters for a support vector machine with RBF-kernel;
- Changing the type of co-occurrence matrices and their control parameters;
- Selecting the way of pre-processing of co-occurrence matrices;
- Recognition with weighted distances.
- Most of these steps are very comprehensive computational tasks.

To reduce the feature space and to decrease computing cost, a simple descriptors' preprocessing was used. Positions, where the elements in each descriptor are equal to zero, were excluded from further analysis. Such a technique allows to significantly reduce the dimensionality of the data. The descriptors can reach 65,536 elements at most. After removal of zero elements, the descriptors contain no more than 5000 elements.

Detailed information on each experiment is given below.

3.1 Selection of optimal parameters for a support vector machine with RBF-kernel

Support vector machines (SVMs) [4] are a set of related supervised learning methods used for classification and regression. Viewing input data as two sets of vectors in an *n*-dimensional space, an SVM will construct a separating hyperplane in that space, one which maximizes the margin between the two data sets.

For classification task, we use C-classification with the RBF kernel

$$K(x, y) = e^{-\gamma\|x-y\|^2}, \gamma > 0.$$

In [5] authors suggested that in general RBF is a reasonable choice. The RBF kernel nonlinearly maps samples into a higher dimensional space, so it, unlike the linear kernel, can handle the case when the relation between class labels and attributes is nonlinear. In addition to that, there are only two parameters while using RBF kernels: C and γ.

The calculations were performed with grid-search on C and γ using cross-validation. Since doing a complete grid-search may still be time-consuming, we used a coarse grid first. After identifying a better region on the grid, a finer grid search on that region was conducted.

For each data set we first use a coarse grid on the initial interval $C=2^{-5}, 2^{0},\ldots,2^{15}$ and $\gamma=2^{-50}, 2^{-48},\ldots,2^{10}$ (Fig.2) . After finding the best (C, γ) we conduct a finer grid search on the neighborhood of the point (Fig.3).

As illustrated in Fig.3, a better rate 84,89% at (2^{3},2^{-33}) is gained.

3.2 Changing the type of co-occurrence matrices and their parameters

The image recognition experiments on the first stage are carried out for the co–occurrence of colors of neighboring pixels only. It is reasonable to suppose that if we add to the existing descriptors information about the neighboring pixels that are located at a greater distance from each other, the recognition quality could be improved.

Another way of looking to further improvements of descriptors is to consider so–called concatenated descriptor, which can be obtained by concatenation of co-occurrence descriptors of original images and their pyramided versions (i.e., with every second pixel row and column eliminated).

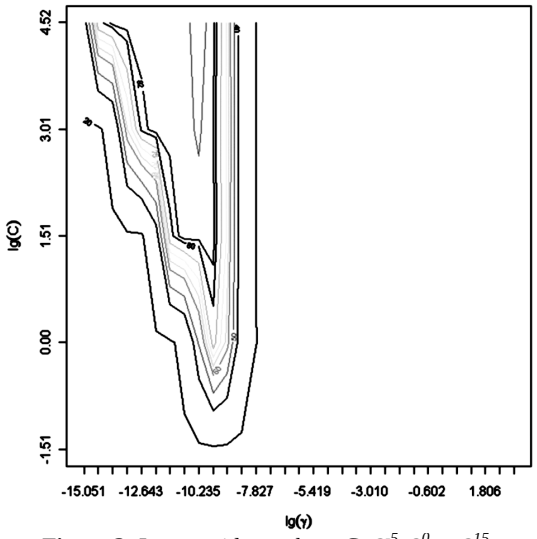


Figure 2: Loose grid-search on $C=2^{-5}, 2^{0},\ldots,2^{15}$ and $\gamma=2^{-50}, 2^{-48},\ldots,2^{10}$.

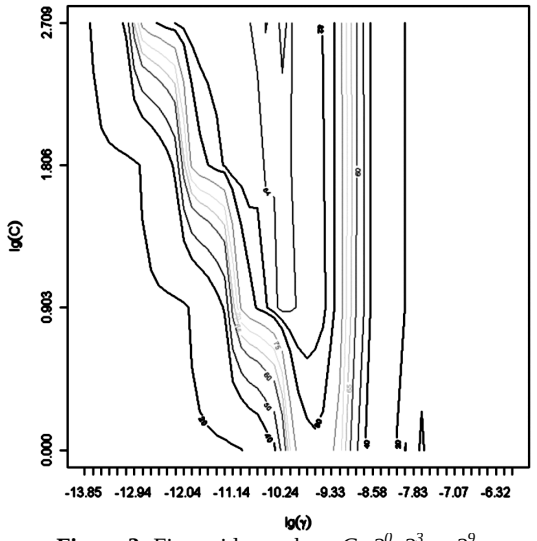


Figure 3: Fine grid-search on $C=2^{0}, 2^{3},\ldots,2^{9}$ and $\gamma=2^{-46}, 2^{-45.5},\ldots,2^{-20}$.

The decimation consists in removal of every second row and every second column from original image.

If $\|U_{ij}\|, i=\overline{1,n},\ j=\overline{1,m}$ is co–occurrence matrix of an original image, then a vector $u_{image}=(u_{11},u_{12},\ldots,u_{1m},u_{21},\ldots,u_{2m},\ldots,u_{n1},\ldots,u_{nm})$ is its descriptor. In a similar manner, if $\|V_{ij}\|, i=\overline{1,k},\ j=\overline{1,l}$ is co–occurrence matrix of decimated image, then as its descriptor we

have a vector $v_{image} = (v_{11}, v_{12}, \ldots, v_{1l}, v_{21}, \ldots, v_{2l}, \ldots, v_{k1}, \ldots, v_{kl})$, for $k = \left\lceil \frac{n}{2} \right\rceil$, $l = \left\lceil \frac{m}{2} \right\rceil$.

Further calculations are performed on normalized vectors of the form $w = u_{image} \cup v_{image}$. The normalization is achieved by division of each vectors' part into total pairs of first (305522 pairs) and second (75962 pairs) images respectively

The descriptors, thus defined, second data set presents. Such an experiment have been performed in perfect analogy to the previous stage. In that case the greatest possible recognition efficiency amount to 84.67%.

3.3 Selecting the way of pre-processing of co-occurrence matrices

Partially pre–processing of co–occurrence matrices was occurred in each of the previous experiments. The pre–processing consists in reduction of common zero elements. This allows to increase processing speed.

But in addition, any image from digital camera has a noise. It does not carry any useful information and it must be eliminated from further analysis.

For this purpose we applied thresholding. The threshold was defined as a percentage of the total number of occurrence pairs in the picture. The next threshold values were defined: 0,1% 0,2% 0,5%. For the source images with a total of 305,522 pairs after thresholding we can get 306, 611 and 1528 pairs respectively. For decimated images with a total number of 75,962 pairs we can get 76, 152 and 380 pairs.

Of course, the number of descriptors' elements significantly reduced after the operation.

In Table 1 elements count are listed after thresholding of descriptors.

Table 1: Elements count in descriptors.

	Initial number	After thresholding		
		0,1%	0,2%	0,5%
Basic descriptor	256^2	563	423	245
Descriptor of pyramided image	256^2	323	229	155
Concatenated descriptor	2×256^2	886	652	400

Further experiments were conducted to determine the quality of recognition for the filtered descriptors (Figure 4).

It can be seen that recognition quality is best achieved on descriptors of original image with threshold 0,5% (88,69%). That corresponds to the analysis of 245 features (look at Table 1).

With combining the two types of descriptors appreciable recognition quality was not gained. When it is considered that concatenated descriptor construction and its further use requires additional resources (processing time and additional memory to store), there is little point in using of concatenated descriptors.

Another possible approach to the pre–processing co–occurrence matrices is to use principal component analysis (PCA) [6]. PCA is a useful statistical technique that has found application in fields such as face recognition and image compression, and is a common technique for finding patterns in data of high dimension.

Experiments have been performed to determine quality of recognition after rearrangement of co-occurrence matrices by PCA. For three sets of data, was taken number of principal components with cumulative proportion of 0,5, 0,6, 0,7, 0,8 and 0,9. Obtained principal components were used as features to perform recognition with the help of SVM.

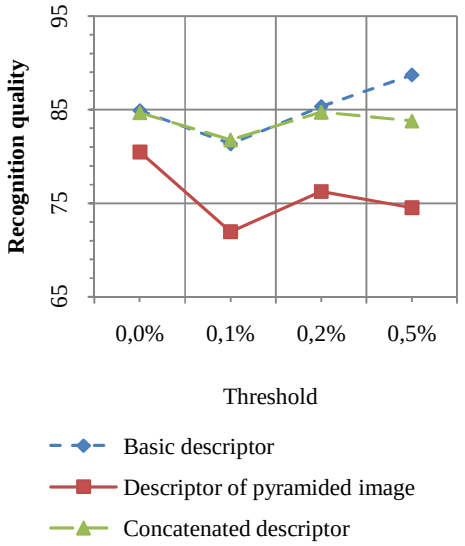


Fig.4: Recognition quality before and after thresholding.

Considering Table 2, it can be concluded that the best recognition quality (86,51%) is achieved for the descriptors of decimated image and the cumulative proportion equal to 0,5. It according to analysis of 12 principal components only.

Table 2: Relationships between cumulative proportion of PCA variance and recognition quality, %.

	0,5	0,6	0,7	0,8	0,9
Basic descriptor	79,52	83,31	81,73	74,05	63,62
Descriptor of pyramided image	86,51	77,17	73,58	66,44	54,78
Concatenated descriptor	83,24	83,97	81,37	74,01	59,55

3.4 Recognition with weighted distances

The next stage of the study is recognition with minimal weighted distances to a class object.

For each recognition object weighted distance to all images of each class was calculated. Minimum distance was a criterion of belonging of some class. To calculate the distance between the descriptors, four different distances were taken: Euclidean distance, Chebyshev distance, Manhattan distance and Canberra distance.

For two vectors, $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$ the distances are defined as follows:

- Euclidean distance: $d_{euclidean}(X,Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$,

- Chebyshev distance: $d_{chebyshev}(X,Y) = \max_i |x_i - y_i|$,
- Manhattan distance: $d_{manhattn}(X,Y) = \sum_{i=1}^{n} |x_i - y_i|$,
- Canberra distance: $d_{canberra}(X,Y) = \sum_{i=1}^{n} \frac{|x_i - y_i|}{x_i + y_i}$.

The results of the experiment are summarized in Table 3.

You can see that the most effective recognition was 67,64%. The value is achieved on descriptors of original image with Canberra distance. With the use of Canberra distance we achieve the best recognition results on other types of descriptors.

Table 3: Recognition quality for different types of descriptors and different distances.

	Euclidean			**Chebyshev**		
Basic descriptor	59,15			55,94		
Descriptor of pyramided image	57,04			45,85		
Concatenated descriptor	59,64			49,53		
Basic descriptor with thresholding (0,1%; 0,2%; 0,5%)	59,89	59,48	59,44	56,35	56,44	56,19
Descriptor of pyramided image with thresholding	56,55	56,16	57,03	45,63	45,11	45,93
Concatenated descriptor with thresholding	59,36	60,04	60,24	48,72	49,81	49,58
	Manhattan			**Canberra**		
Basic descriptor	63,14			67,64		
Descriptor of pyramided image	55,43			62,43		
Concatenated descriptor	58,94			65,20		
Basic descriptor with thresholding (0,1%; 0,2%; 0,5%)	63,01	62,73	62,23	66,71	66,57	66,25
Descriptor of pyramided image with thresholding	55,05	54,77	55,41	61,22	60,78	61,26
Concatenated descriptor with thresholding	58,23	58,72	58,31	63,86	64,07	63,79

4. CONCLUSION

The paper examined several approaches for recognition of 24–bit color images sized 320×240 pixels, acquired with the help of the web camera. An investigation on the influence of different factors on the object recognition quality has been accomplished. Specifically, the following factors were studied:

- Selection of optimal parameters for a support vector machine with RBF-kernel;
- Changing the type of co-occurrence matrices and their control parameters;
- thresholding of descriptors
- principal component analysis
- Selecting the way of pre-processing of co-occurrence matrices;
- Recognition with weighted distances.

Based on the experiments, it may be deduced that the best recognition quality (88,69%) is achieved using SVM classifier with basic color co-occurrence descriptors of original images and with the matrix element selection threshold equal to 0,5% of the total number of pixel pairs.

These results will be considered on the implementation of Mobile Voiced Visual Assistant aimed to a practical use.

Acknowledgments. This work has been partially supported by the ISTC grant B-1682.

5. REFERENCES

[1] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008, http://www.r-project.org/

[2] Kovalev V. and Volmer S. *Color Co-Occurrence Descriptors for Querying-by-Example*, Int. Conf. on Multimedia Modelling, Oct. 12-15, Lausanne, Switzerland, IEEE Comp. Society Press, pp. 32-38, 1998.

[3] Kovalev V.A. *Towards image retrieval for eight percent of color-blind men*. Int. Conf. on Pattern Recognition, Cambridge, UK, 23-26 Aug 2004, IEEE Comp Society Press, Vol. 2, pp. 943-946, 2004.

[4] Vapnik V. *The nature of statistical learning theory*. New York, NY: Springer–Verlag, 1995.

[5] C.–W. Hsu, C.–C. Chang, C.–J. Lin. *A practical guide to support vector classification* . Technical report, Department of Computer Science, National Taiwan University. July, 2003.

[6] Jolliffe I. T. *Principal Component Analysis, Series: Springer Series in Statistics*, 2nd ed., Springer, NY, 2002.

About the authors

Vassili Kovalev is a Head of Biomedical Image Analysis Group, United Institute of Informatics Problems. His contact email is vassili.kovalev@gmail.com.

Igor Safonov is a junior researcher of Biomedical Image Analysis Group. His contact email is safonov@tut.by.

Super-resolution and Optical Flow reliability fields

Oleg Maslov and Konstantin Rodyushkin*
Intel Corporation

Abstract

Accurate optical flow estimation of motion fields is crucial for video super-resolution algorithms. Existing algorithms for optical flow calculation may produce erroneous motion fields on complex dynamic scenes with multiple moving, occluding non-rigid objects. In this paper we propose to use so called optical flow reliability weights in order to reduce impact of erroneous motion vector estimates on quality of super-resolution. We propose method for estimation of reliability weights which is based on structural analysis of motion vector fields. We present results on real video sequences and demonstrate the advantages of the proposed methods compared to conventional optical flow based super-resolution methods.

Keywords: Image and video processing, Super-resolution, Optical Flow

1 Introduction

We consider problem of applying optical flow based super-resolution methods to real world video sequences. Such problems evolves in everyday demand for enhancing quality of low resolution video from hand-held mobile devices, low resolution obsolete video cameras or from video web services with high degree of compression. Calculating optical flow (OF) between frames often produces inadequate and erroneous estimates that in place results in bad quality of super-resolved images since registration information accuracy is crucial for super-resolution quality.

In this paper we propose a novel method for evaluating quality of optical flow data and then we apply this method in bayesian super-resolution framework and show the quality improvement on a set of real videos. The structure of this paper is organized as follows. In Section 2 we discuss papers where optical flow data is used as registration parameter for super-resolution. In section 3 we introduce our method. Then in Section 4 results for several video sequences are presented.

2 Related Work

There have been published a number of papers related to super-resolution based on OF registration data [Baker and Kanade 1999; Zhao and Sawhney 2002; Fransens et al. 2007]. Real video sequences may have multiple moving non-rigid and occluding objects. One method of describing non-rigid transformations between frames is a usage of dense motion vector fields proposed by Baker and Kanade in [Baker and Kanade 1999]. Authors proves feasibility of usage of OF as registration data for super-resolution. There exists a few number of methods for evaluating OF motion fields between images [Farnebäck 2003; Bruhn et al. 2005; Lucas and Kanade 1981]. However even most accurate of them may produce erroneous and noisy results. In [Lee and Kang 2003] authors considered registration error in global translation parameters. Using regularization term in Maximum a Posteriori (MAP) estimate they incorporate registration error caused by inaccurate motion information into minimization deconvolution functional. In [Zhao and Sawhney 2002] authors present OF error as additive stochastic noise in motion vectors measurements and show that consistency and accurateness of OF estimation is crucial to robust OF based super-resolution. In [Zomet et al. 2001] there was proposed robust median estimator in a super-resolution for dealing with outliers. It is used in gradient calculation during minimization of cost error function. It was shown that using median allows effectively suppress outliers caused by motion errors, noise, motion blur. In [Ben-Ezra et al. 2005] authors used adaptive super-resolution method, which detects blocks of 16×16 pixels with multiple motions present. In [Weiss 1998] it was proposed to Expectation-Maximisation (EM) algorithm for estimating simultaneously motion layers and motion parameters. In [Fransens et al. 2007] a set of so called visibility maps is introduced. Visibility maps signal whether or not a scene point on reconstructed high resolution (HR) image is visible in the low resolution (LR) input images. These maps are hidden binary variables, corresponding to visibility or occlusion, respectively. Expectation-Maximisation algorithm is then used, which iterates between estimating values for the hidden quantities, simultaneously optimizing the OF motion vectors and super-resolution image.

In this paper we propose to assess OF quality in form of set of so called OF reliability weights $W \in (0, 1)$. For each point of OF field between each pair of input LR images such OF reliability weights determine quality (accuracy) of OF in that point. These weights are estimated based on structural analysis of motion vector fields and are incorporated into super-resolution minimization functional. It allows us during deconvolution process to take into account reliable OF data and corresponding LR image pixels while rejecting OF errors like geometry and intensity outliers.

3 Method

This section discusses our method. We introduce imaging model and statistical bayesian super-resolution with reliability weights. Two subsections 3.2.1 and 3.2.2 describe methods we propose for estimating reliability weights.

3.1 Robust Super-Resolution with reliability weights

Suppose we have a set of $2N_f+1$ low-resolution (LR) input images $T_i, i \in [-N_f, \ldots, N_f]$ with resolution $N_w \times N_h$ pixels. Imaging model similar to [Fransens et al. 2007] is used to reconstruct unknown high resolution (HR) image $J(\mathbf{x})$ of size $mN_w \times mN_h$ pixels, m - magnification factor so that

$$T_i(\mathbf{x}) = S * P * J(F_i(\mathbf{x})) + \varepsilon, \quad (1)$$

where HR image $J(\mathbf{x})$ is warped by OF based operator $F(\mathbf{x})$; $*$ - two-dimensional convolution operator; then point spread function P and downsample operator S are applied and measurement noise ε is added to produce set of LR input images T_i. ε is assumed to be normally distributed with zero mean and covariance Σ, $\mathbf{x}$ is vector of two-dimensional image coordinates, subsampling operator S is dependant on magnification factor m. Since imaging model (1) is linear we may combine warping, PSF and subsampling operator as follows

$$T_i(\mathbf{x}) = \mathbf{H}_i^T(\mathbf{x})\mathbf{J} + \varepsilon, \quad (2)$$

where $\mathbf{H}_i(\mathbf{x})$ is a column vector dependant on operators $F(\mathbf{x})$, P and S, $\mathbf{J}$ is a column vector of HR image intensities $J(\mathbf{x})$ rearranged in lexicographical order.

*e-mail:{oleg.maslov,konstantin.rodyushkin}@intel.com

Suppose we have function $W_i(\mathbf{x})$ determining OF vector quality for each pixel $\mathbf{x}$ of observed LR image T_i. If $W_i(\mathbf{x}) = 1$ then OF vector for $\mathbf{x}$ is good and can be used for super-resolution, if $W_i(\mathbf{x}) = 0$ then OF vector is bad and the pixel $\mathbf{x}$ should not be used in super-resolution. Let us call this function $W_i(\mathbf{x})$ function of *OF reliability weights*.

Using approach from [Nestares et al. 2006] and introduced function of OF reliability weights let us formulate deconvolution functional in form of bayesian MAP estimate. This functional consists of likelihood part and a priori part. For likelihood and a priori nonlinear M-estimators based robust function are used which models outliers for pixel values. In this paper we use Cauchy function $\rho(x)$ as a robust M-estimator.

$$\rho(x) = \frac{c^2}{2}\log\left(1 + (x/c)^2\right) \tag{3}$$

As an a priori probability for image we use Markov Random Fields (MRF) since they are a common choice to model prior probabilities on images. Resulting deconvolution functional is:

$$E(\mathbf{J}) = \sum_i \sum_{\mathbf{x}} W_i(\mathbf{x})\rho\left(\frac{\mathbf{H}_i^T(\mathbf{x})\mathbf{J} - \hat{T}_i(\mathbf{x})}{\sigma_l}\right) + \lambda \sum_{(k,l)\in\Omega} \rho\left(\frac{\mathbf{V}_{(k,l)}^T\mathbf{J}}{\sigma_p}\right) \to \min \tag{4}$$

where $\hat{T}_i(\mathbf{x})$ are input LR images; σ_l, σ_p are scale factors defining error magnitude that is considered to be an outlier in likelihood and prior terms correspondingly; λ - relative weight given to the prior with respect to the likelihood; Ω - index pairs of neighbor pixels of MRF; $\mathbf{V}_{(k,l)}$ - is a vector defined by

$$\mathbf{V}_{(k,l)} = \{\delta_{i,k} - \delta_{i,l}\}, \quad i = 1, \ldots, mN_w \times mN_h \tag{5}$$

here $\delta_{i,l}$ - Kronecker symbol. For finding minimum of (4) we use nonlinear conjugate gradient optimization method [Barrett et al. 1994]. The gradient of (4) is

$$\nabla E(\mathbf{J}) = \sum_i \sum_{\mathbf{x}} \frac{W_i(\mathbf{x})}{\sigma_l}\rho'\left(\frac{\mathbf{H}_i^T(\mathbf{x})\mathbf{J} - \hat{T}_i(\mathbf{x})}{\sigma_l}\right)\mathbf{H}_i(\mathbf{x}) + \lambda \sum_{(k,l)\in\Omega} \frac{1}{\sigma_p}\rho'\left(\frac{\mathbf{V}_{(k,l)}^T\mathbf{J}}{\sigma_p}\right)\mathbf{V}_{(k,l)}. \tag{6}$$

This method uses iterations. Each iteration consists of the following steps:

1. Calculate gradient $\nabla E(\mathbf{J})$ using (6)
2. Calculate conjugate gradient search direction $\mathbf{D}$
3. Estimate search step size α along direction $\mathbf{D}$ by minimizing $E(\mathbf{J} + \alpha\mathbf{D})$ using Newton-Raphson method
4. Update HR image as $\mathbf{J} = \mathbf{J} + \alpha\mathbf{D}$

Iterations repeat until convergence.

3.2 Ways of calculating reliability weights

Super-resolution procedure described above implies that we have accurate OF motion fields $\mathbf{v}(\mathbf{x})$ which are used by warping operator $F(\mathbf{x}) = m(\mathbf{x} + \mathbf{v}(\mathbf{x}))$. Basically it means that motion vectors map pixels of arbitrary object of the scene exactly to the same pixels of this object across frames. But it might not be true because of the following reasons:

1. Some objects on the scene may disappear across frames. So there might be no correct OF vectors for pixels that belong to this object.
2. OF algorithm itself can produce erroneous results especially on boundaries of moving objects.
3. Objects of the scene can change appearance e.g. because of lighting conditions change or object deformation. In this case OF doesn't work well and even if we have good real OF vectors then these pixels are not good for super-resolution algorithm and should be rejected.

We propose two ways to estimate reliability weights for pixels based on OF map. Let us define reliability weight images for the two ways as W^1 and W^2 correspondingly. Here and below we will omit subscript image index because weights for each LR image are calculated independently and in the same way. The resulting reliability weights will be obtained as follows $W(\mathbf{x}) = W^1(\mathbf{x})W^2(\mathbf{x})$.

3.2.1 Normalization

The idea behind this way of reliability weight calculation is as follows. Each estimated pixel of HR image should use data from not more than one pixel from every input LR frame. e.g. let us consider OF map and corresponding warping operator $F(\mathbf{x})$ that warps two different pixels $\mathbf{x}_1$ and $\mathbf{x}_2$ into one pixel $\mathbf{x}_{hr}$. In other words

$$\mathbf{x}_{hr} = F(\mathbf{x}_1) = F(\mathbf{x}_2). \tag{7}$$

In this case super-resolution procedure considers intensities of both pixels $\mathbf{x}_1$ and $\mathbf{x}_2$ of the same input LR image as independent measurements which are used for estimation of one pixel $\mathbf{x}_{hr}$ of HR image $J(\mathbf{x})$. However it could not be true because from one LR frame we can get only one measurement for one HR pixel $\mathbf{x}_{hr}$. Then we should set such reliability weights w_1 and w_2 for LR pixels $\mathbf{x}_1$ and $\mathbf{x}_2$ so that $w_1 + w_2 = 1$ e.g. *normalize* them. We assume that x_1 and x_2 have equal impact on the HR pixel therefore we can set reliability weights for these pixels $w_1 = w_2 = 1/2$. In case of three pixels warped to one pixel of HR image we set reliability weights for such pixels as $1/3$ and so on. To obtain weights for the whole image we apply the following algorithm:

1. Calculate vector $\mathbf{C}_{hr}$ as follows:

$$\mathbf{C}_{hr} = \sum_{\mathbf{x}'} \mathbf{H}(\mathbf{x}') \tag{8}$$

   where $\mathbf{C}_{hr}$ is vector of image C_{hr} pixels' values rearranged in lexicographical order and summation is done over all pixels of LR image. In result C_{hr} image is a "back projection" of LR image, which pixels are all set to one. Thus $C_{hr}(\mathbf{x})$ is the number of pixels from LR image used for estimation of the pixel $\mathbf{x}$ from HR image.

2. Map image C_{hr} to input LR image to obtain C_{lr} image ("direct projection")

$$C_{lr}(\mathbf{x}) = \mathbf{H}^T(\mathbf{x})\mathbf{C}_{hr} = \mathbf{H}^T(\mathbf{x})\sum_{\mathbf{x}'} \mathbf{H}(\mathbf{x}') \tag{9}$$

   In result $C_{lr}(\mathbf{x}) - 1$ is number of pixels from input LR image which are used to estimate HR pixel along with LR pixel $\mathbf{x}$.

3. Calculate weight image $W^1(\mathbf{x}) = \frac{1}{C_{lr}(\mathbf{x})}$

The illustration of such normalized transformation is shown on figure 1.

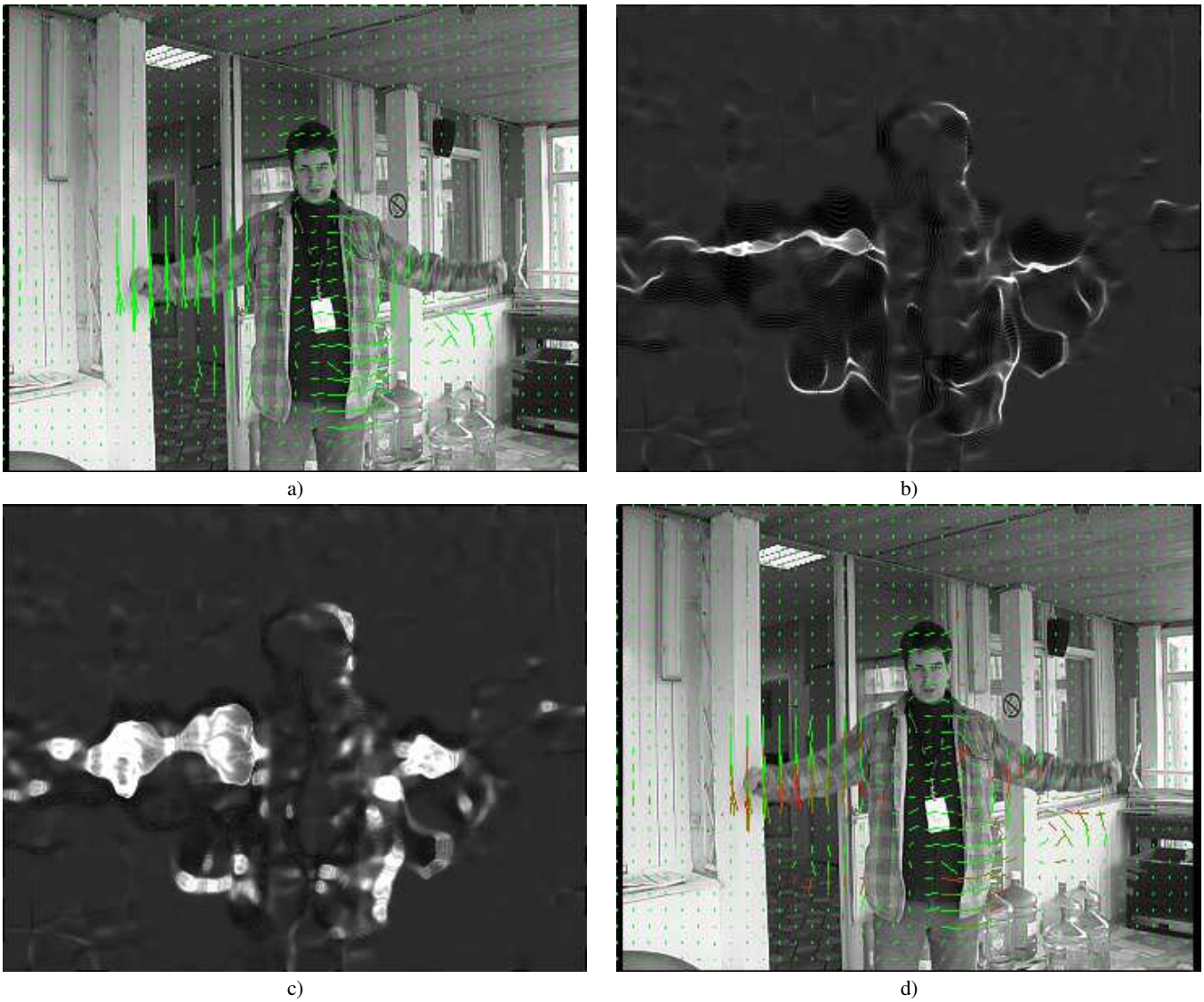


Figure 1: *Calculation of OF reliability weights W^1. a) Input LR image with motion vectors, b) HR image C_{hr} - "inverse projection" of LR image containing one pixel values, c) C_{lr} image - "direct projection" of C_{hr}, d) Input LR image with reliability weighted motion vectors. Red vectors have zero weight and will not be used for super-resolution. Green vectors are related to good for super-resolution LR pixels with weight* 1.

3.2.2 Motion clusters

The second way of reliability weights calculation is based on the following idea. In order to obtain good super-resolution quality we should have image of the same object on several frames. It means that image of the object should have no significant deformations across frames. And therefore all motion vectors which belong to the object should be described by rigid motion e.g. translation, translation and rotation, or arbitrary affine transform.

This section describes algorithm of detecting groups of pixels with OF vectors within the same rigid motion i.e. *motion clusters*. When such detection is done we can mark all pixels which belong to the same rigid motion as good ones for super-resolution and mark the rest as bad and remove them from further consideration.

Suppose two images consist of pixels that either belong to one of N motion clusters described by rigid transformation $T(p_i, \mathbf{x})$, $i = 1 \dots N$ or belong to motion cluster that could not be described by rigid transformation. Each rigid transformation is defined by parameter vector p_i. If pixel from first image belong to rigid transformed area i then its position on the second image can be calculated as follows $\mathbf{x} + T(p_i, \mathbf{x})$.

We introduce a set of hidden variables $\nu(\mathbf{x})$. If $\nu(\mathbf{x}) = 0$ then corresponding pixel x does not belong to any of rigid motion clusters. $\nu(\mathbf{x}) = 1, 2, \dots, N$ means that pixel belongs to one of the N rigid motion clusters described by rigid transformation $T(p_i, \mathbf{x})$ and transformation parameters p_i $(i = 1, \dots, N)$.

We suppose that motion vectors are independent random values. So we can write joint probability distribution function (PDF) of all motion vectors $\mathbf{v}$ as product of PDF of each motion vector $\mathbf{v}(\mathbf{x})$ over all pixels $\mathbf{x}$.

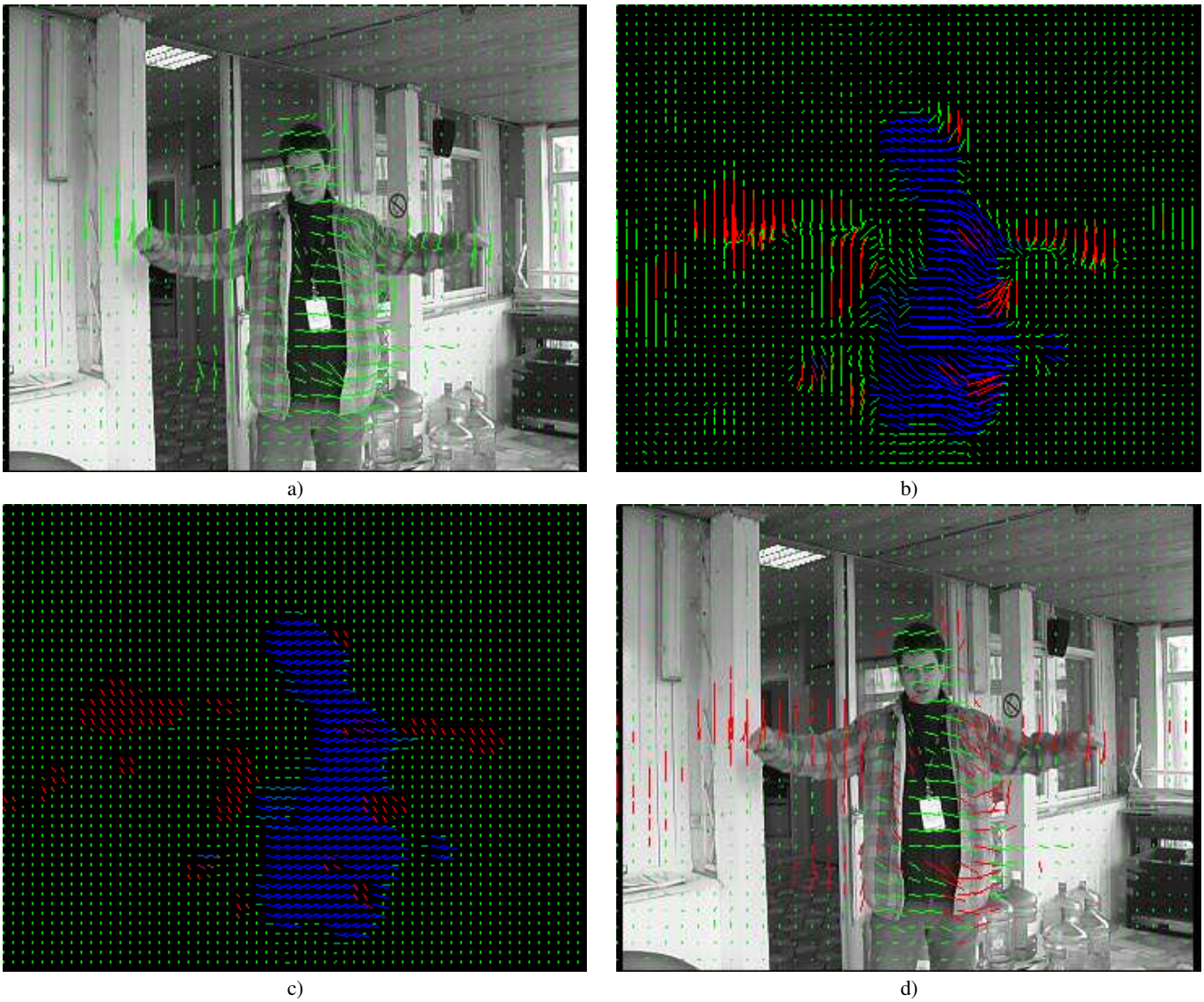


Figure 2: *Calculation of OF reliability weights W^2. a) Input LR image with motion vectors, b) Initial OF vectors. Red color denotes vector which does not belong to any rigid motion, green and blue colors indicate that vectors belong to first or second rigid motion cluster correspondingly. c) Rigid motion OF vectors. They show average motion directions for each cluster. d) Input LR image with reliability weighted motion vectors. Red vectors have zero weight and will not be used for super-resolution. Green vectors are related to good for super-resolution LR pixels with weight* 1.

$$P(\mathbf{v}|\mathbf{p},\nu) = \prod_{\mathbf{x}} P(\mathbf{v}(\mathbf{x})|\mathbf{p},\nu(\mathbf{x})), \tag{10}$$

here $\mathbf{p} = [p_0, p1, \ldots, p_N]$ is vector with transformation parameters , p_0 is a unknown parameters for non-rigid motion and $p_i, i = 1, \ldots, N$ are unknown parameters for motion vectors of i^{th} rigid transformation.

We suppose that the distribution function $P(\mathbf{v}(\mathbf{x})|\mathbf{p},\nu(\mathbf{x}))$ for each motion vector is described by 2D gaussian distribution:

1. If pixel $\mathbf{x}$ does not belong to rigid motion cluster i.e. $\nu(\mathbf{x}) = 0$ then

$$P\left(\mathbf{v}(\mathbf{x})|\mathbf{p},\nu(\mathbf{x}) = 0\right) = \frac{1}{2\pi\sigma_{out}^2} e^{-\frac{|\mathbf{V}(\mathbf{X}) - T(p_0,\mathbf{X})|^2}{0.5\sigma_{out}^2}} \tag{11}$$

2. If pixel $\mathbf{x}$ belongs to rigid motion cluster $k = 1 \ldots N$ i.e. $\nu(\mathbf{x}) = k$ then

$$P\left(\mathbf{v}(\mathbf{x})|\mathbf{p},\nu(\mathbf{x}) = k\right) = \frac{1}{2\pi\sigma^2} e^{-\frac{|\mathbf{V}(\mathbf{X}) - T(p_k,\mathbf{X})|^2}{0.5\sigma^2}} \tag{12}$$

where σ_{out} is deviation parameters for non rigid motion, and σ is deviation parameters for rigid motion. We suppose that σ_{out} and σ values are known and $\sigma_{out} \gg \sigma$.

The goal is to find transformation parameters of rigid and non rigid motions $\mathbf{p}$ using maximum likelihood criteria

$$\hat{\mathbf{p}} = \arg\max_{\mathbf{p}} \left\{ \log \sum_{\nu} P(\mathbf{v}|\mathbf{p},\nu) \right\} \tag{13}$$

k-3 k k+3

Figure 3: *Original 3 frames from set of 15 input frames.*

We will use expectation-maximization (EM) algorithm [Dempster et al. 1977] to solve problem (13).

E-step: One the E-step of the k^{th} iteration conditional probability distribution function $P(\nu|\mathbf{v}, \hat{\mathbf{p}}_{k-1})$ of hidden variables ν is estimated. $\hat{\mathbf{p}}_{k-1}$ is transformation parameters estimated on previous iteration during M-step. We suppose that $\nu(\mathbf{x})$ are independent from each other. Therefore we can rewrite the estimated distribution as product of PDFs of $\nu(\mathbf{x})$

$$\hat{P}(\nu|\mathbf{v}, \hat{\mathbf{p}}_{k-1}) = \prod_{\mathbf{x}} \hat{P}(\nu(\mathbf{x})|\mathbf{v}(\mathbf{x}), \hat{\mathbf{p}}_{k-1}), \tag{14}$$

$\hat{P}(\nu(\mathbf{x})|\mathbf{v}(\mathbf{x}), \hat{\mathbf{p}}_{k-1})$ is estimated as follows

$$\hat{P}\left(\nu(\mathbf{x}) = i|\mathbf{v}(\mathbf{x}), \hat{\mathbf{p}}_{k-1}\right) = w_i(\mathbf{x}), \tag{15}$$

$$w_i(\mathbf{x}) = \frac{P\left(\mathbf{v}(\mathbf{x})|\hat{\mathbf{p}}_{k-1}, \nu(\mathbf{x}) = i\right)}{\sum_{k=0}^{N} P\left(\mathbf{v}(\mathbf{x})|\hat{\mathbf{p}}_{k-1}, \nu(\mathbf{x}) = k\right)} \tag{16}$$

where $P(\mathbf{v}(\mathbf{x})|\hat{\mathbf{p}}_{k-1}, \nu(\mathbf{x}))$ are gaussian distributions defined in (11) and (12).

M-step: On the M-step transformation parameters $\mathbf{p}$ are estimated by means of optimization of following cost function

$$Q(\mathbf{p}) = \sum_{\nu} \hat{P}(\nu|\mathbf{v}, \hat{\mathbf{p}}_{k-1}) \log P(\mathbf{v}|\mathbf{p}, \nu) \tag{17}$$

Using (10), (11), (12), (14), and (16) it can be shown that optimization of (17) is equal to minimization of the following expressions

$$\hat{p}_{i,k} = \arg\min_{p_i} \left\{ \sum_{\mathbf{x}} w_i(\mathbf{x}) \left|\mathbf{v}(\mathbf{x}) - T(p_i, \mathbf{x})\right|^2 \right\} \tag{18}$$

here i is cluster index $i = 0 \dots N$, $w_i(\mathbf{x})$ is estimated conditional PDF from (16).

For linear transformations T i.e. translation, scaling, rotation or arbitrary affine transform the task (18) is equal to solution of system of linear equations and can be solved analytically.

When new parameters $\hat{p}_{i,k}$ are calculated the next iteration is made. After several iterations we obtain the final estimation

$$\widetilde{p}_i = \hat{p}_{i,L} \tag{19}$$

where L is number of iterations.

Once we estimated parameters for N rigid motions clusters we can check if pixel belongs to one of the rigid motion clusters. In order to do this for each pixel $\mathbf{x}$ we calculate minimal distance between motion vector $\mathbf{v}(\mathbf{x})$ and motion vectors $T(\widetilde{p}_i, \mathbf{x})$ of rigid motion clusters found.

$$d(\mathbf{x}) = \min_{i=1\dots N} |\mathbf{v}(\mathbf{x}) - T(\widetilde{p}_i, \mathbf{x})| \tag{20}$$

Based on this distance map we calculate weights for each pixel as follows

$$W^2(\mathbf{x}) = \begin{cases} 1, \text{ if } d(\mathbf{x}) < \sigma \\ 0, \text{ if } d(\mathbf{x}) \geq \sigma \end{cases} \tag{21}$$

The figure 2 shows example of rigid motion cluster estimation. In this example two rigid motion clusters and non-rigid motion cluster are detected.

4 Experiments

The purpose of experiment was to compare quality of conventional super-resolution algorithm and proposed super-resolution algorithm with optical flow reliability weights. The algorithm was

a) b) c) d)

Figure 4: *Results of super-resolution algortihms: a) The original input LR image (central frame), b) result by bicubic interpolation, c) result by conventional super-resolution algorithm, d) result by super-resolution algorithm with reliability weights*

tested on two real world examples. There exists a variety of optical flow method. For estimating OF fields we used multiresolution two-frame OF algorithm proposed by Farnebäck in [Farnebäck 2003] since it gives a robust and adaptive to lighting conditions motion vectors estimation. OF averaging window size was 25×25, 3 levels of gaussian pyramids were used and 6 iterations were made on each pyramid level. Super-resolution was performed on the whole input image. For processing color images frames were transformed to luminance and color opponent space YCbCr. In this space image channels were super-resolved independently. Since luminance component contains high-frequency content typically it took at least $15 - 20$ iterations of conjugate gradient algorithm to converge. For color components we iterated 4 times. Magnification factor $m = 2$ was applied. For PSF function box filter of size $[m \times m]$ was used. We used 15 adjacent LR video frames to obtain each output HR video frame. Central frame magnified by factor m was used as an initial approximation for conjugate gradient algorithm. We assume here that typical scene contains 3 types of motion: foreground, background and non-rigid motion. So the number of motion clusters was set to 3. The variation parameters for OF vector PDFs (11), (12) were $\sigma = 2$ and $\sigma_{out} = 100\sigma$ correspondingly. Proposed algorithm was implemented in Microsoft Visual C/C++. Experiments were conducted on Pentium IV 3.2 Ghz machine.

For the first experiment we used real video captured by DV (digital video) camcorder in outdoor environment. Input image resolution was 360×288 pixels, 50 frames per second (FPS). Output video resolution Video contains static background and foreground with man rapidly waving arms. Some of the input 15 frames are presented in the top row of figure 3. The camera is shaking slightly in order to obtain sub-pixel shifts for super-resolution. The comparative results of bicubic interpolation and super-resolution algorithms are shown in the top row of figure 4. Here whole super-resolved image and zoomed fragment containing arms is shown. Rapidly moving arms produce large error in the optical flow estimation especially for the pairs of the most distant frames. It results in clearly visible ghost artifacts on the super-resolved image obtained by conventional algorithm. Our algorithm detects erroneous motion vectors and adjusts reliability weights correspondingly. In result output HR image contains significantly less artifacts caused by rapid motion.

For the second experiment video taken by Canon point and shoot digital camera. Input resolution was 320×240 pixels, FPS 15. The complex scene contains rapidly moving and occluding cars. The camera tracks car position. Some of input frames are presented in bottom row of figure 3. Results are shown in bottom row of figure 4. Conventional algorithm produces significant artifacts on moving cars while super-resolution with reliability weights suppresses them.

5 Summary

In this paper we proposed a novel method of using reliability weights as a robust way of producing high-quality super-resolution images based on erroneous and noisy estimates of motion vectors on complex dynamic videos. We proposed two novel methods for estimation of reliability weights based on structural analysis of motion vector fields. This approach allows to suppress visual artifacts' appearance during super-resolution caused by multiple moving deformable non-rigid objects, and may be applied also to scene detection where adjacent frames contain completely different scenes. Results on real video sequences demonstrate the advantages of the proposed methods compared to conventional optical flow based super-resolution method.

References

BAKER, S., AND KANADE, T. 1999. Super-resolution optical flow. *Technical Report CMU-RI-TR-99-36, The Robotics Institute, Carnegie Mellon University*.

BAKER, S., AND KANADE, T. 2002. Limits on super-resolution and how to break them. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 9, 1167–1183.

BARRETT, R., BERRY, M., CHAN, T. F., DEMMEL, J., DONATO, J., DONGARRA, J., EIJKHOUT, V., POZO, R., ROMINE, C.,

AND DER VORST, H. V. 1994. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*. SIAM, Philadelphia, PA.

BEN-EZRA, M., ZOMET, A., AND NAYAR, S. K. 2005. Video super-resolution using controlled subpixel detector shifts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*, 6, 977–987.

BRUHN, A., WEICKERT, J., AND SCHNORR, C. 2005. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *IJCV 61*, 3 (February), 211–231.

DEMPSTER, A. P., LAIRD, N. M., AND RUBIN, D. B. 1977. Maximum likelihood from incomplete data via the em algorithm. *J. Royal Stat. Soc. 39*, 1–38.

FARNEBÄCK, G. 2003. Two-frame motion estimation based on polynomial expansion. *Proceedings of the 13th Scandinavian Conference on Image Analysis* (June-July), 363–370.

FARSIU, S., ROBINSON, D., ELAD, M., AND MILANFAR, P., 2004. Advances and challenges in super-resolution.

FRANSENS, R., STRECHA, C., AND VAN GOOL, L. 2007. Optical flow based super-resolution: A probabilistic approach. *CVIU 106*, 1 (April), 106–115.

LEE, E. S., AND KANG, M. G. 2003. Regularized adaptive high-resolution image reconstruction considering inaccurate subpixel registration. *IEEE Transactions on Image Processing 12*, 7, 826–837.

LUCAS, B., AND KANADE, T. 1981. An iterative image registration technique with an application to stereo vision. *IJCAI81*, 674–679.

NESTARES, O., HAUSSECKER, H., AND ETTINGER, S. 2006. Computing a higher resolution image from multiple lower resolution images using model-based, robust bayesian estimation. *U.S. Patent Application No. 20060002635*.

WEISS, Y. 1998. *Bayesian motion estimation and segmentation.* PhD thesis, MIT.

ZHAO, W., AND SAWHNEY, H. 2002. Is super-resolution with optical flow feasible? *ECCV02*, I: 599 ff.

ZOMET, A., RAV-ACHA, A., AND PELEG, S. 2001. Robust super-resolution. *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on 1*, I–645–I–650 vol.1.

Regression Analysis of Correlation between Video Coding parameters and Sequence Modification Analyzers

Ilya Brailovskiy, Natalya Solomeshch
Intel Corporation,
{ilya.v.brailovskiy, natalya.solomeshch}@intel.com

Abstract

In this paper we analyze correlation between video sequence modification scoring and video coding algorithms such as Motion Estimation and Rate Control. Using two open source codecs X.264 and IPP H.264 we conducted number of tests. By means of linear regression analysis we processed obtained test data. Based on this analysis we conclude that Decimation, Motion and Noisy Frame techniques have high correlation with different codec algorithms. Moreover, we refined our experiments further and found out that higher rank in Noisy Frames insertion analysis correlates with better Scene Change Detection algorithm and with higher visual quality. However, the abovementioned methods have tangible correlation with such encoding characteristics as number of B pictures between reference frames and thus can't be used in competitive ranking without certain assertions. We also point out that Overall analyzer has high correlation with all key encoding decision algorithms and potentially very good for "black box" codec competitive analysis.

Keywords: *Video codec analysis, visual quality metrics, linear regression, scene change detection*

1. INTRODUCTION

Automated analysis of visual quality for video codecs such as MPEG codecs or AVC [1,10,11] is one of the most crucial and important aspects which arises in design, development and implementation of video codecs. Full reference image quality metrics like PSNR or SSIM [2-3] provide relatively good correlation with subjective human perception. Such Visual Quality metrics however do not provide a way to characterize codecs in a holistic way. Codecs quality depends on a great number of input parameters such as bitrate, input video source and different algorithmic options. Visual Quality metrics could characterize only one point in the codec parameter space since a full reference metric just compares one input video source with one output video sequence. Such comparison could provide certain indication on how good a codec performs on specific input and with fixed parameter settings but gives no information on how the codec operates for different bitrate or another source clip.

To overcome this limitation number of methodologies had been proposed [5-7]. In this paper we will focus on the methodology proposed in papers [6,7] which allows systematically analyze and compare video codecs by iterating though different encoding parameters following certain rules. Compared to some other methodologies (for example [5]) this approach presumably allows some insight into codec internals as well as connecting these internals with the output visual quality.

Goal of this paper is by means of computational experiments for existing codec implementations analyze and prove or disprove correlation between codecs scoring provided by the methodology [6,7] and specific encoding options. Another goal we would like to achieve is to find analyzer's and codec's parameters limitations (if any) for "black box" testing based on empirical study for the selected set of analyzers proposed in [6,7]. Please note that initially we do not assume that a more advanced algorithm provides higher scoring, or higher scoring provides better visual quality. Our goal is just to establish a correlation between codec scoring and encoding parameters. However during the course of our research we found certain empirical connection between higher scoring and better visual quality when more advanced encoding configurations are applied.

2. CODEC ANALYSIS METHODOLOGY

We briefly recall codec assessment and analysis methodology proposed in [6,7]. We will strictly follow these papers in our overview.

2.1 Video Codec Scoring with Specially Prepared Video Sequence

The overall high-level scheme of the examined method is provided at Fig. 1.

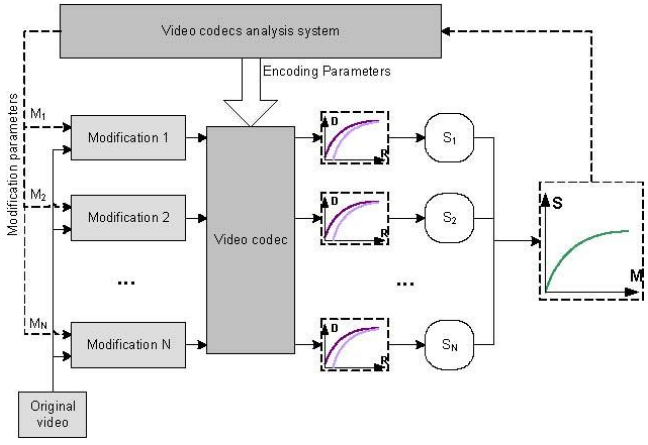


Fig. 1. General scheme of video codec analysis.

The first stage involves generation of a series of video feeds according to known modification parameters M_i. These video feeds could be created by modifying natural video or could be generated fully artificially. Regardless the way these feeds are created we will call them "modified sequences". Modified sequences are submitted to the codec under the test, which encodes and decodes each of the modified streams. Resulted reconstructed data is compared with initial modified sequences and Rate-Distortion (RD) curves are calculated. The same procedure is applied to the reference video codec and RD curves for the reference codec are constructed. Based on calculated RD curves, baseline scoring S^r is computed for each of the modifications by the following formulae:

$$S_{1,2}^{[a,b]} = \exp\left\{\frac{1}{b-a}\int_a^b \ln\left(\frac{R_1(D)}{R_2(D)}\right)dD\right\},$$

where $R_1(D)$ and $R_2(D)$ are the RD curves for the tested and

reference codecs correspondingly and *[a,b]* is the range of the quality metric that we use to conduct our comparison. Value of $S_{1,2}^{[a,b]}$ characterizes the average ratio of bitrate, for a same quality for a set amount of introduced distortion that can be achieved by a video codec with corresponding RD characteristics.

At the next phase the video codec analysis system takes the arrays of scoring results S^r, S^t, $S_{ref,tested}^{[a,b]}$ for the original sequence (if any) and the array of modification parameters *M* as the input and calculates the scoring value estimate for a sequence modification by following formula:

$$Q(M,S^r,S^t) = (1-\gamma)\left(\alpha(M,S^t) - \alpha(M,S^r)\right) + \gamma\, S_{ref,tested}^{[a,b]},$$

$$\alpha(M,S) = \frac{n\sum_i M_i\, S_i - \sum_i M_i \sum_i S_i}{n\sum_i {M_i}^2 - (\sum_i M_i)^2},$$

where γ is a constant in the range *[0,1]*, i is modified sequence number, n - total number of modifications for given analyzer and α*(M,S)* is the slope of the approximating line (the average rate of change of the coding efficiency).

The final step combines estimates for several different codec analyzers and produces the final codec score. For more details on scoring value calculation please refer to [6,7].

Let us briefly recall what the Noisy Frames, Decimation, Motion and Overall analyzers are.

2.2 Modification analyzers overview

Noisy frames analyzer. The analysis lies in modification of input video sequence in order to complicate temporal prediction algorithms. In modified sequence some frames are replaced by "noisy frames", the whole consisting of random noise. One can vary the number of such frames from test to test, usually from one to ten. This method is assumed to reveal a quality of work of the scene detector and of the rate control method in the first place.

Decimation analyzer compares RD-curves obtained during encoding of original and modified video sequences. The modified sequence is created by systematic removal of frames from the original one. One can vary the number of the removed frames – one out of each two frames, two out of three frames, etc. This method is expected to have the highest correlation with scene analyzer and rate control method.

Motion analyzer makes the comparison of RD-curves, obtained during encoding of synthetic video sequences. Synthetic sequences are composed of blocks of randomly generated sizes and colors. Some of the blocks are moving with random velocity. Complexity of this analyzer could be varied by varying number of such blocks, their speed and color. Motion analyzer is as it can be by judged by the name supposed to allow evaluating methods of Motion Estimation implemented in the encoders.

The main idea behind the **Overall analyzer** is to use a set of objective metrics over number of source streams simultaneously in order to increase the approximation with subjective measurements. It is expected that the Overall analyzer would correlate with all the encoding parameters with some emphasis on commonly accepted encoding algorithms responsible for the quality of codecs: Rate Control, Scene Change Detection, Rate-Distortion optimizations, MacroBlock Decisions as well as some others algorithms.

2.3 Regression Analysis and Test Setup

For studying dependency of modification analyzers scores on codec encoding options we are using quite straightforward and well-proven linear regression analysis [12]. Against each possible combination of encoding option calculate analyzer score and linear regression coefficients for encoding options. We consider correlation between chosen analyzer and encoding option as a high if corresponding regression coefficient is "high". We would divide regression coefficients on "highs" and "lows" empirically by examining the difference between the maximal and minimal coefficient values.

Unfortunately we can't execute such experiments for all possible codecs and for all possible encoding options. We need to limit our analysis to certain codecs, specific streams and specific encoding options bound by many practical limitations. As we will show a little later even limited range of options leads to quite significant computational time.

We used 4 streams of High Definition (HD), Standard Definition and Common Intermediate Format (CIF) resolutions in our experiments. For each resolution we selected 5 bitrates for RD curves calculation, more details on selected test streams can be found in the Table 1. We run our tests iterating though all encoding parameters simultaneously but for each resolution (HD, SD and CIF) separately.

We conducted our experiments with several codecs: X.264 Open Source H.264 encoder [8], IPP Media Sample H.264 encoder and IPP Media Sample MPEG2 encoder [9]. All these codecs are Open Source thought distributed under difference licenses. Compared to some other Open Source codecs such as JM reference model these codecs are significantly faster which makes execution of bigger number of test runs possible. In the Table 2 you could see which systems we used for testing and in the Table 3 you could see how long it took us to run those tests even with selected relatively fast codecs.

Although we didn't run some other codecs such as MPEG4 part 2 or VC1 [1] because of computational resource limitations we think that the results obtained for H.264 codecs wouldn't be significantly different for other codecs. Some indications that that might be true are provided by the fact that for several runs of MPEG2 encoder we observed quite similar to H.264 codecs results. Extending results of our research for the whole family of AVC encoders looks quite reasonable for us since we took different AVC encoders implementations and observe certain consistency in the results. However, strictly speaking, our study applies for those codecs we tested only.

Type stream	Name	Resolution	Bitrate (Kbit/s)
CIF	Foreman	352x288	200, 400, 800, 1600, 2200
SD	Iceage	720x576	750, 1200, 2000, 4000, 7000
	Lotr	720x416	
HD	Troy	1920x1072	2000,5000,8000,11000,15000

Table 1. Parameters of the used sequences.

Computer numbers	OS	CPU	Cores Num	CPU (GHz)	MEM (Gb)
1	Windows Server 2003 R2 Ent	Xeon	16	2.4	16
2	Windows Vista Enterprise	Xeon 3230	8	2.66	2
3	Windows Server 2003	Core 2 Quad	4	2.4	4
4	Windows Server 2003 R2 x64	P4	2	2.8	1

Table 2. System Configuration used for testing.

	Noise frames	Overall	Decimation	Motion
X264 Cif	0.83 day (1)*	2 days(4)	2 days(4)	2 days (4)
X264 SD	25 days (1)	17 days(4)	17.5 days(4)	
X264 HD	25 days(1)	14.5 days (3)	30 days(3)	
IPP264 Cif	2.5 days (1)	2 days (3)	2 days(3)	3.33 days (4)
IPP264 SD	22 days(2)	16.33 days(3)	13 days(3)	
IPP264 HD	26 days(1)	27 days (3)	19 days(3)	

Table 3. Average time of executing various measurements. In the brackets one can see the number of used computer.

* The numbers in the brackets correspond to used computer numbers.

For the iterations we choose following encoding parameters mostly available with command line options of the selected encoders:

1. The number of frames between the Intra-frames (keyint: 100, 200, 300);

2. The minimal number of Bi-predictive frames between the Intra and Predicted frames (Bframes: 0, 2);

3. Bitrate control method (rc method): for X.264 the options are (variable bitrate and constant bitrate bufsize 1100 kbit), and for IPP.264 (variable bitrate and constant bitrate);

4. Motion Estimation method: for X.264 (me method: diamond search, hexagonal search, uneven multi-hexagon search, exhaustive search, hadamard exhaustive search); for IPP.264 (MV_SEARCH_TYPE: LOG, EPZS, FULL_ORTHOGONAL, LOG_ORTHOGONAL, UMH, SQUARE, FTS, SMALL_DIAMOND);

5. Macroblocks size: (subblock split:

1. I16x16 P16x16 B16x16;
2. I16x16 P16x16 B16x16 I8x8 P8x8 B8x8;
3. I16x16 P16x16 B16x16 I8x8 P8x8 B8x8 I4x4 P4x4 B4x4);

6. Combination of various optimizations: for X.264 (subme + trellis: 1&0, 3&0, 4&1, 6&1, 7&2, 9&2) and for IPP.264 (combination of various optimizations – specific value in par-file which turns on/off similar options: 0,1,2,3,4,5);

7. Flag which indicates presence of Scene Change Detector Analyzer: for X.264 (no-scenecut), for IPP.264 (internal building preprocessing directive option) IPP MPEG2.

We wouldn't go into the details of the meaning of these options, please refer to [8,9,11].

3. RESULTS

We summarized results of our regression analysis coefficients calculation in the Tables 4 and 5. We present only SD resolution results since HD and CIF results have essentially the same pattern for both AVC codecs. Let us highlight some of the observations one can make by looking into the tables.

At first we would like to go though the results which are common for both codecs. Scene Change detection algorithm, Rate Control method and Subblock split option all have very high impact for both codecs for almost all analyzers. KeyFrame interval has small impact on both codecs. ME algorithm choice have small impact on the quality for both codecs mostly because it has high impact on performance which is out of the scope of this study. Noisy Frame and Decimation analyzers both are mostly influenced by RC method and Scene Change Detection algorithms. Overall analyzer demonstrates similar correlation pattern for both codecs though Scene Change Detection algorithm has much smaller impact on this Analyzer for IPP H.264 compared to X.264. Regression coefficients for Motion analyzer are mostly influenced by Subblock Split and Scene Change Detection options because these algorithms contribute most to Mode Decision process.

Second, let us emphasize some differences between codecs. B-frames number has significant impact on IPP H.264 while it has almost negligible impact on X.264 encoder. IPP H.264 is more influenced by Rate Control for all analyzers while X.264 is not.

Varying parameters	Noise frames	Decimation	Motion	Overall
Keyint	0.1506	1.3935	0.0156	0.3789
Bframes	0.0110	0.06545	0.3854	0.0301
Rc method	24.2025	9.0777	0.6607	7.4427
Me method	0.1876	0.0732	0.8339	0.4331
Subblock split	7.5508	2.3768	9.5350	7.1469
Regimes of optimization	4.0886	2.0157	1.3081	6.6421
Scene Change Detection	19.7372	22.2783	14.9058	26.2165

Table 4. Linear Regression Analysis Coefficients for X.264 Open Source H.264 encoder on SD sequences.

Varying parameters	Noise frames	Decimation	Motion	Overall
Keyint	1.1311	2.020	0.0000	1.9778
Bframes	6.4867	8.018	9.7880	0.4410
Rc method	25.3586	31.0780	14.0873	23.555
Me method	0.0529	0.0103	1.1588	0.0828
Subblock split	4.6134	1.9246	10.6397	2.2763
Regimes of optimization	4.9054	0.3718	0.0357	8.8911
Scene Change Detection	10.5375	4.6043	8.8423	2.0616

Table 5. Linear Regression Analysis Coefficients for IPP Media Sample H.264 encoder on SD sequences.

Most of the differences between two codecs are observed for Motion analyzer, and they fall into the general codecs' differences most likely caused by the difference in the implementation of these codecs. However Scene Change Detection and Subblock Split regression coefficients for Motion Analyzer are on the higher side for both encoders.

In order to examine influence of Scene Change Detection and Subblock Split options on Motion Analyzer more precisely we performed additional test on X.264 Open Source H.264 encoder using extended set of parameters. We have conducted additional linear regression analysis for Motion Analyzer (Table 6) and included chromaticity motion estimation option (no-chroma-me on and off) into consideration. The names of the synthetic sequences 1, 2, …, 15 represent the complexity of the Motion

Analyzer. Smaller numbers represent simpler sequences with slower motion; bigger numbers represent higher complexity and higher motion. From obtained data we can observe that influence of Scene Change Detection goes up with higher speed and more complex sequences. This effect could be explained that when complexity goes up Scene Change Detection algorithm can't find good correlation between two consecutive frames and often assumes a new scene.

Varying parameters	Names of the comparing sequences			
	1,2,3,4,5,6, 7,8,9,10,11 , 12,13,14,1 5	1,15	1,2	14,15
Me method	1.2517	0.8376	1.9692	0.8367
Subblock split	7.8453	9.9343	12.9871	8.9431
Regimes of optimization	1.3446	1.0707	5.9788	0.7874
Scene Change Detection	18.9256	18.4604	4.4785	19.4741
Chroma-ME	1.8508	1.1622	0.1311	1.4784

Table 6. Results of the regression analysis for the Motion analyzer for X.264 Open Source H.264.

From the results of this experiment we can suggest that if Motion Analyzer is planned to be used for evaluating quality of Motion Estimation and Macroblock Decisions algorithms it should set smaller speed for moving blocks. Complexity classes from 1 to 3 are probably enough to evaluate the complexity of core Motion encoding algorithms.

4. INFLUENCE OF THE SCENE CHANGE DETECTION ON THE QUALITY OF ENCODED VIDEO SEQUENCE.

As we mentioned earlier the influence of Scene Change Detection (SCD) algorithm is quite significant for both codecs especially in case of Noisy Frames. (Please note that we will be using acronym SCD for Scene Change Detection in this section). It's naturally to look further and understand if a better SCD algorithm provides better quality in case of inserted Noisy Frames. For that we will investigate cases when SCD algorithm is on or off. We assume that when SCD is on the quality should be better, and our goal is to check this assumption.

For verifying our assertions in regard with SCD quality impact we need to make sure that 1) encoder with SCD produces different encoded streams compared to an encoder without SCD for the streams with noisy frames. 2) quality of the sequences encoded with SCD has better than quality 3) it's good to see if the encoder uses more Intra frames and places them where the Scene Changes occurs (it's not necessary – sometimes encoders can mitigate a scene change by different means).

To see that SCD indeed improves the visual quality we've built RD curve to compare quality of the encoder when SCD on and off. As you can see from Fig.2 curve with SCD on is higher which means better visual quality.

Table 7 summarize results of analyzing Frame Type statistics for the streams with and without SCD and answers the question 1). Table 7 partially answers question 3) since we see increase of I frames number in case of SCD. We also did more thorough analysis where SCD algorithm places I frames, in most cases excessive I frames are placed where Noisy Frames inserted – which means they are placed where Scene Change occurs.

SCD on	Percentage of frame types	Ordinal numbers of the frames coming after an Intra-type frame				
		№ 1	№ 2	№ 3	№ 4	№ 5
	i, %	15.4	9.2	12.3	1	<1
	p, %	47.4	64.9	53.6	55.6	64.9
	b, %	37.1	25.7	34	43.2	35

SCD off	Percentage of frame types	Ordinal numbers of the frames begin with the frame on which the encoder with analyzer placed an Intra-type frame					
		№ 0	№ 1	№ 2	№ 3	№ 4	№ 5
	i, %	5.1	<1	<1	<1	<1	<1
	p, %	40.2	40.2	19.5	40.2	40.2	19.5
	b, %	54.6	59.7	80.4	59.7	59.7	80.4

Table 7. Frame type summary for SD-type Iceage video sequence with eight noisy frames inserted.

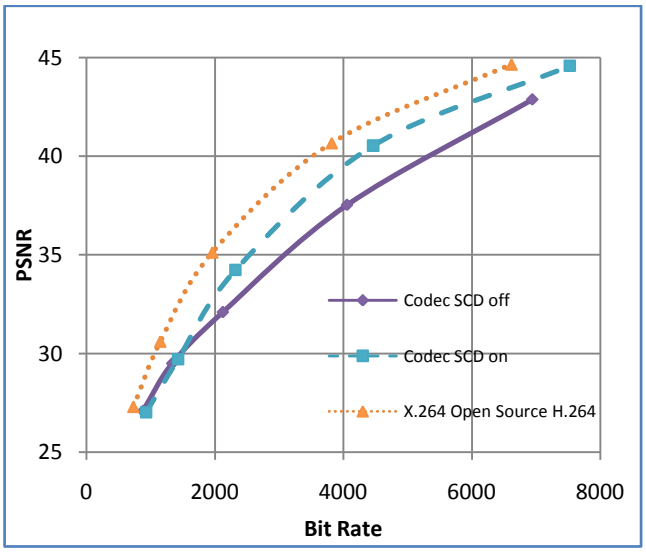


Figure 2. R-D curves for IPP H.264 on a Noisy Stream (8 noise frames inserted) on Iceage stream.

To understand Noisy Frame correlation with SCD better for other class of codecs we extended our experiments further and did similar analysis for IPP Media Sample MPEG2. We obtained quite consistent results such as on Figure 2 for MPEG2 as well which we don't copy to this paper to avoid redundancy with AVC results.

Additionally for both codecs we did visual inspection with enabled SCD and disabled SCD. For both AVC and MPEG2 codecs we noted better or much better visual quality depending on bitrate or on the number of inserted frames. Video clips with higher bitrate or with lesser number of inserted frames looked less different for codecs with SCD on and off. However when there are more inserted frames we observe much better quality for the

codecs with SCD on. We put example of observed visual differences on Figure 3. There are significant blocking artifacts on Figure 3 (a) compared to Figure 3 (b). As the only difference during the coding was the presence of the SCD, it can be concluded, that the analysis of a scene change makes a tangible contribution to the visual quality which can be practically demonstrated with Noisy Frame Insertion technique.

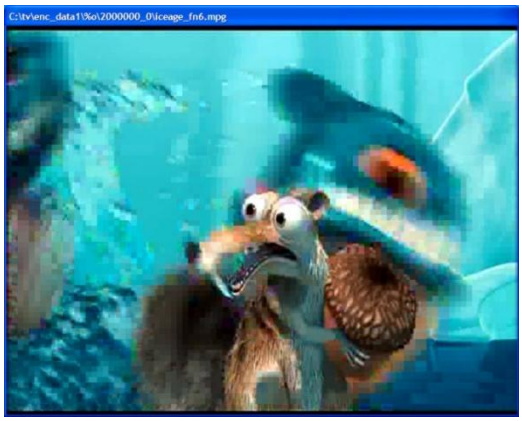


Figure 3.A Scene Change Detection is absent.

Figure 3.B Scene Change Detection is present.

Figure 3. The visual difference of the quality on 2000 Kbits, 6 noisy frames, for IPP MPEG2 codec with Scene Change Detection (SCD) on and off. (A) SCD is absent. (B) SCD is present.

5. CONCLUSION

Based on performed analysis we conclude that for evaluated X.264 and IPP H.264 codecs the analyzer methodology proposed in [6,7] provides high correlation with key internal encoding algorithms and can be effectively used for testing and comparing video encoders.

Some of the methods described above are not suitable however for "black box" codec testing. For example, Decimation Analysis has high correlation with the number of B Frames for IPP H.264 encoder and not neglible correlation with keyframe interval for X.264 encoder. In order to effectively use Decimation Analysis we recommend aligning GOP structures of the codecs under consideration.

If Motion analyzer is intended to evaluate codecs difference related to Motion Estimation and Macroblock Decisions then less complicated modifications should be used, preferable in the complexity range from 1 to 3.

Nosiy Frame analysis can be effectively used as an indicator of quality of Scene Change Detection algorithm.

Overall analyzer has expectedly high correlation with all major codec quality algorithms and potentially could serve as a measure of overall codec quality. Based on our study applicability of overall analyzer to "black box" codec competitive analysis is highly probable. At the same time, overall analyzer should be examined further since in this paper we haven't covered correlation of overall analysis ranking with visual quality.

6. ACKNOWLEDGMENTS

We would like to thank Alexey Leonenko for very useful discussions and feedback during whole course of this research.

We also would like to thank Alexander Parshin and Dmitry Vatolin for their kind permission of publishing Figure 1 and some other materials obtained during joint research works.

7. LITERATURE

[1] E. G. Richardson H.264 and MPEG-4 video compression: Wiley and Sons, 2003

[2] Digital video quality: vision models and metrics. Stefan Winkler. Published by John Wiley and Sons, 2005

[3] Digital video image quality and perceptual coding Hong Ren Wu, Kamisetty Ramamohan Rao. Published by CRC Press, 2006

[4] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, Apr. 2004

[5] S. Wolf and M. H. Pinson, "Low bandwidth reduced reference video quality monitoring system", First Int'l Workshop on Video Proc. And Quality Metrics, Jan 2005

[6] A. Parshin, D. Vatolin, I. Brailovskiy and P. Corriveau, Novel Approach for Video Codecs Scoring using Specially Prepared Video Sequences. In Proceedings of the Fourth International Workshop on Video Processing and Quality Metrics for Consumer Electronics conference, Jan 2009.

[7] A. Parshin and D. Vatolin Methods of analysis of videocodecs based on modifications of natural of natural videosequencies. Graphicon 2008 conference, June 2008

[8] X.264 encoder: http://x264.nl/

[9] IPP H.264 and MPEG2 encoders: http://software.intel.com/en-us/intel-ipp/

[10] ITU-T and ISO/IEC JTC 1, "Generic coding of moving pictures and associated audio information – Part 2: Video," ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG-2), Nov. 1994 (with several subsequent amendments and corrigenda).

[11] Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC). Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVTG050, 2003.

[12] Introduction to Linear Regression Analysis (3rd ed.). Douglas C. Montgomery, Elizabeth A. Peck, and G. Geoffrey Vining . New York: Wiley, 2001.

About the authors

Natalya Solomeshch is an Engineering Intern at Intel and student of MIPT. Her contact email is natalya.solomeshch@intel.com.

Ilya Brailovskiy, PhD, is a Video and Platform Architect with Intel Corporation. His contact email is ilya.v.brailovskiy@intel.com.

Improved Visible Differences Predictor Using a Complex Cortex Transform

Alexey Lukin
Laboratory of Mathematical Methods of Image Processing,
Department of Computational Mathematics and Cybernetics,
Moscow Lomonosov State University, Russia
lukin@graphics.cs.msu.ru

Abstract

Prediction of visible differences involves modeling of the human visual response to distortions in the image data. Following the approach of Daly [1], this paper introduces several algorithm improvements allowing for more accurate calculation of threshold elevation and modeling of the facilitation effect during phase-coherent masking. This is achieved by introducing a complex-valued cortex transform that separates the response magnitude from the instantaneous phase within each band of the cortex transform. The magnitude component is used for calculation of the mask contrast. The phase component takes part in modeling of the facilitation effect.

Keywords: *Visible Differences Predictor, VDP, Cortex Transform, Image Quality Assessment, Complex Cortex Transform, Human Visual System, HVS, Masking.*

1. INTRODUCTION

Prediction of visible differences means estimation of subjective visibility of distortions in the image data. Algorithms for prediction of such differences are important in automated quality assessment of imaging systems, including lossy compression of video signals, assessment of transmission channel distortions, optimization of realistic image synthesis algorithms, etc.

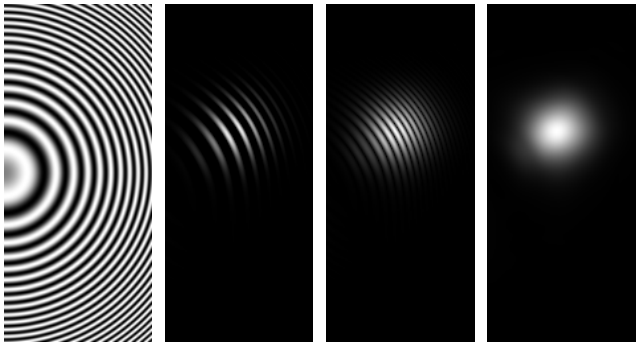
Many image quality metrics have been proposed in the literature. The most successful objective metrics include models of the *Human Visual System* (HVS) for prediction of such effects as non-uniform sensitivity to spatial frequencies and visual masking, like the *Visual Differences Predictor* (VDP) proposed by Daly [1].

Daly's VDP uses a (modified) *cortex transform* [2] to decompose the image into subbands of different spatial frequency and orientation. It allows modeling of frequency-dependent and orientation-dependent masking in the human visual system. For each cortex transform band, the contrast of the difference signal and the contrast of the masking signal are evaluated. Threshold elevations are calculated from the contrast of the mask signal. They are used to calculate the probability of detection of the difference signal, subject to visual masking. The detection probabilities are summed over all cortex transform subbands.

The work of Mantiuk et. al. [3] proposes several improvements to the model of Daly, including evaluation of contrast in JND (just noticeable difference) units, and varying CSF (contrast sensitivity function) depending on the local luminance adaptation level.

A shortcoming of the "traditional" cortex transform is the inability to accurately model phase-invariant masking (explained in the next section). For example, a chirp image signal in Fig. 1a would produce an oscillating signal in each cortex band, as in Fig. 1b. This, in turn, would produce an oscillating mask contrast signal and oscillating threshold elevation, as in Fig. 1c.

In this paper, a modification of the cortex transform is introduced to obtain phase-independent estimates of the masking contrast, as in Fig. 1d. Section 2 describes the cortex transform and its phase-variance. Section 3 introduces the Complex Cortex Transform (CCT) and its computation algorithm. Section 4 illustrates the use of CCT for evaluation of masking thresholds in VDP. Section 5 presents the computational results of threshold elevations.



(a) (b) (c) (d)

Figure 1: (a) A chirp image; (b) Response of a "traditional" cortex filter (only positive signal part is shown); (c) Threshold elevation image (or mask contrast) produced using a "traditional" cortex filter; (d) Threshold elevation image produced using a "complex" cortex filter proposed in this paper.

2. CORTEX TRANSFORM IN VDP

The cortex transform is first described by Watson in [2] as an efficient means of modeling the neural response of retinal cells to visual stimuli. The cortex filter in the frequency domain is produced as a product of 2 filters: the 'dom' filter providing frequency selectivity and the 'fan' filter providing orientation selectivity:

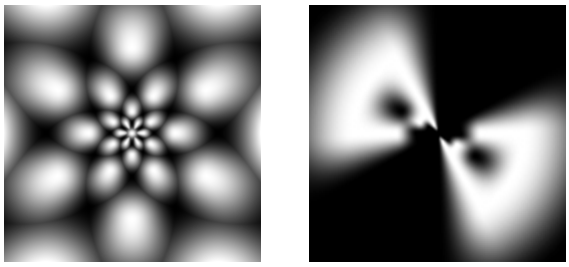
$$\text{cortex}_{k,l}\left(\rho,\theta\right) \equiv \text{dom}_k\left(\rho\right) \cdot \text{fan}_l\left(\theta\right), \qquad (1)$$

where k is the index of the frequency band, l is the index of orientation, and (ρ, θ) are polar coordinates in the frequency space (corresponding Cartesian coordinates will later be denoted as (ω_1, ω_2)). Fig. 2 illustrates frequency responses of several cortex filters, and Fig. 3a shows the example *impulse response* (point spread function) of the cortex filter.

The cortex transform decomposes the input image $I(i,j)$ into a set of subband images $B_{k,l}(i,j)$ (*cortex bands*) as follows

$$B_{k,l}(i,j) \equiv \mathcal{F}^{-1}\left\{\text{cortex}_{k,l}\left(\rho,\theta\right) \cdot \mathcal{F}\left\{I(i,j)\right\}\right\}, \qquad (2)$$

where $\mathcal{F}$ is the 2D discrete Fourier transform.



(a) (b)

Figure 2: Frequency responses of several cortex filters (brightness represents gain for the given spatial frequency).
(a) Responses plotted separately for cortex$_{0,0}$; $_{1,0}$; $_{1,2}$; $_{1,4}$; $_{2,1}$; $_{2,3}$; $_{2,5}$; $_{3,0}$; $_{3,2}$; $_{3,4}$; $_{4,1}$; $_{4,3}$; $_{4,5}$; $_{5,0}$; $_{5,2}$; $_{5,4}$ — each filter produces 2 symmetrical blobs on a complex 2D frequency plane;
(b) Responses of 8 (out of 31) cortex filters are added together towards a constant gain.

It can be noted (Fig. 2) that cortex filters are linear-phase bandpass filters. Their frequency responses are designed to sum up to 1, which means that the sum of cortex bands is equal to the input image: $\sum_{k,l} B_{k,l}(i,j) = I(i,j)$.

One of the problems with the cortex transform is that real-valued cortex bands lack separation between magnitude and phase components of the neural response. According to formula (14.32) in [1], the strength of visual masking $T_e^{k,l}(i,j)$ (also known as *Threshold Elevation*) depends on the absolute value of the normalized mask contrast $\left|m_n^{k,l}(i,j)\right|$

$$T_e^{k,l}(i,j) = \left(1 + k_1 \left(k_2 \left|m_n^{k,l}(i,j)\right|^s\right)^b\right)^{1/b}, \qquad (3)$$

where k_1, k_2, s, and b are psychophysically derived constants.

The normalized mask contrast $m_n^{k,l}(i,j)$ is calculated as the cortex transform of the CSF-filtered input image in the perceptually linearized luminance scale:

$$m_n^{k,l}(i,j) = \mathcal{F}^{-1}\left\{\text{cortex}_{k,l}(\rho,\theta) \cdot \text{csf}(\rho,\theta) \cdot \mathcal{F}\left\{I(i,j)\right\}\right\} \quad (4)$$

It is easy to see that for sinusoidal mask signals, the mask contrast $m_n^{k,l}(i,j)$ exhibits oscillations between 0 and the magnitude of the masker (Fig. 6a,b). These oscillations are also present in the threshold elevation map. Their cause lies in insufficient separation of magnitude and phase of neural response by the modulus operation $\left|m_n^{k,l}(i,j)\right|$ in Eq. (3).

According to this simplified model, sinusoidal gratings produce maximal threshold elevation at positive and negative peaks of the masker waveform and absolutely no masking at zero crossings of the waveform. This contradicts with psychophysical data [4] assuming that sinusoidal gratings produce spatially uniform (or nearly-uniform) threshold elevation.

3. THE COMPLEX CORTEX TRANSFORM

3.1 Explanation of Goals

To eliminate this mismatch between calculated masking maps and psychophysical experiments, we suggest a more sophisticated model for separation of magnitude and phase information in the neural response. The importance of such separation has been acknowledged in [1] and [3], but the separation algorithm has not been elaborated.

Our proposed algorithm stems from the publication of Pollen and Ronner [5] which investigates phase relationships between adjacent cells in the visual cortex. It has been found that adjacent simple neural cells are often tuned to the same orientation and spatial frequency, but their responses differ by the phase angle that is often approximately $\pi/2$. In other words, receptive fields of corresponding neural cells comprise *quadrature* (phase-complementary) filters.

A recently published *Berkeley Wavelet Transform* [6] is the orthogonal wavelet transform using phase-complementary wavelets of 4 different orientations. Its filters are localized in space, frequency and orientation, so the transform is suitable for modeling of visual receptive fields.

However our method is based on the cortex transform [2],[1] because it allows for more flexible tiling of the 2D frequency plane and better frequency/orientation selectivity. Our goal is to modify the cortex transform in order to enable efficient magnitude/phase separation and provide the shift invariance of magnitude estimates for sinusoidal gratings.

3.2 A 2D Hilbert Transform

A well studied method for extracting magnitude and phase information of narrow-band 1D signals is the *Hilbert transform* [7]. The Hilbert transform can be considered as a filter that rotates every frequency component of the signal by $\pi/2$, for example $\mathcal{H}(\cos t) = \sin t$. The Hilbert transform can be used to convert a real-valued signal $f(t)$ into an analytic complex-valued signal $f(t) + i\mathcal{H}f(t)$, whose instantaneous magnitude and phase are defined as

$$A(t) = \sqrt{f^2(t) + (\mathcal{H}f(t))^2}, \qquad (5)$$

$$\phi(t) = \arctan \frac{\mathcal{H}f(t)}{f(t)}. \qquad (6)$$

An efficient computational algorithm for the Hilbert transform employs a direct $\pi/2$ phase rotation of the complex-valued Fourier spectrum $(\mathcal{F}f)(\omega)$ of signal $f(t)$:

$$(\mathcal{F}\mathcal{H}f)(\omega) = i \cdot \text{sgn}(\omega) \cdot (\mathcal{F}f)(\omega)$$

One problem with the Hilbert transform is that it does not have a trivial extension to the 2D case. One possible 2D extension called "skewed Hilbert transform" [8] applies a 1D Hilbert transform along only vertical or only horizontal direction in a 2D image. This is equivalent to multiplying the 2D Fourier spectrum of the image $(\mathcal{F}I)(\omega_1, \omega_2)$ by either $i\,\text{sgn}\omega_1$ or $i\,\text{sgn}\omega_2$ (for vertical or horizontal direction).

In order to build phase-complementary filters of arbitrary orientation, we are proposing a modification of the skewed 2D Hilbert transform that multiplies a 2D Fourier spectrum of the image by the similar filter

$$H_l(\omega_1, \omega_2) = i\,\text{sgn}(p\,\omega_1 + q\,\omega_2), \qquad (7)$$

where the line equation $p\,\omega_1 + q\,\omega_2 = 0$ specifies the desired "direction" of the modified 2D Hilbert transform, and l is the orientation index from Eq. (1).

3.3 Design of Quadrature Cortex Filters

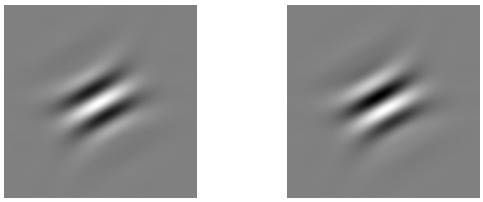
For each cortex filter $\text{cortex}_{k,l}(\omega_1, \omega_2)$, we are designing a phase-complementary filter $\widehat{\text{cortex}}_{k,l}(\omega_1, \omega_2)$ with the same passband frequency range and orientation using the modified 2D Hilbert transform of the impulse response of the filter. Their frequency responses are related as

$$\widehat{\text{cortex}}_{k,l}(\omega_1, \omega_2) \equiv H_l(\omega_1, \omega_2) \cdot \text{cortex}_{k,l}(\omega_1, \omega_2),$$

where p and q in Eq. (7) are linked with the cortex filter orientation angle θ_c (depending on l) as

$$\tan\theta_c = \frac{p}{q}.$$

Fig. 3 shows impulse responses of 2 phase-complementary cortex filters for $k = 2$, $l = 1$, and $\theta_c = \pi/6$. It can be noted that $\text{cortex}_{k,1}$ is a linear-phase filter and its impulse response is centrally symmetrical (Fig. 3a).



(a) (b)

Figure 3: Impulse responses of a pair of phase-complementary cortex filters: (a) $\text{cortex}_{k,1}$; (b) $\widehat{\text{cortex}}_{k,1}$.

This pair of phase-complementary cortex filters can be combined into a single complex-valued quadrature filter:

$$\text{cc}_{k,l}(\omega_1, \omega_2) = \text{cortex}_{k,l}(\omega_1, \omega_2) + i\,\widehat{\text{cortex}}_{k,l}(\omega_1, \omega_2) \qquad (8)$$

The decomposition of the input image into complex-valued subbands using filters $\text{cc}_{k,l}$ will be called a Complex Cortex Transform (CCT).

$$\text{CCT}_{k,l}(i,j) \equiv \mathcal{F}^{-1}\left\{\text{cc}_{k,l}(\omega_1, \omega_2) \cdot \mathcal{F}\left\{I(i,j)\right\}\right\} \qquad (9)$$

3.4 Properties of the Complex Cortex Transform

1. Since both $\text{cortex}_{k,l}$ and $\widehat{\text{cortex}}_{k,l}$ are real filters, from Eqs. (2), (8), and (9) we obtain that cortex bands $B_{k,l}(i,j)$ are equal to the real part of corresponding CCT bands.

   $$B_{k,l}(i,j) = \text{Re}\left\{\text{CCT}_{k,l}(i,j)\right\}$$

   Substituting this into Eq. (4), we obtain

   $$m_n^{k,l}(i,j) = \text{Re}\left\{\mathcal{F}^{-1}\left\{\text{cc}_{k,l} \cdot \text{csf} \cdot \mathcal{F}\left\{I(i,j)\right\}\right\}\right\}$$

2. The real and imaginary CCT bands contain phase-complementary responses, similarly to those described in [5]:

   $$\text{Im}\left\{\text{CCT}_{k,l}(i,j)\right\} = \mathcal{F}^{-1}\left\{H_l \cdot \mathcal{F}\left\{\text{CCT}_{k,l}(i,j)\right\}\right\}$$

3. Similarly to the 1D case (Eqs. (5) and (6)), instantaneous magnitude and phase within each CCT band can be approximated as

   $$A_{k,l}(i,j) = \left|\text{CCT}_{k,l}(i,j)\right|,$$

   $$\phi(t) = \arg\left\{\text{CCT}_{k,l}(i,j)\right\}.$$

   Here the separation of magnitude and phase information happens along the orientation θ_c of the corresponding cortex filter $\text{cc}_{k,l}$.

3.5 Summary of the CCT Algorithm

The Complex Cortex Transform defined by Eq. (9) is calculated similarly to the regular cortex transform (Eq. (2)), except the fact that the cortex filter is now complex-valued, and the resulting subband images $\text{CCT}_{k,l}(i,j)$ are complex-valued too. The algorithm looks like follows:

1. Complex-valued spectra $\text{cc}_{k,l}(\omega_1, \omega_2)$ of CCT filters are pre-calculated using Eqs. (1) and (8);
2. A 2D Fourier transform $\left(\mathcal{F}\left\{I\right\}\right)(\omega_1, \omega_2)$ of the real-valued input image $I(i,j)$ is calculated;
3. The complex-valued spectrum from step 2 is multiplied by the complex-valued filter $\text{cc}_{k,l}$ from step 1, for each frequency range k and orientation l;
4. An inverse complex-valued 2D Fourier transform is calculated for each spectrum obtained at step 3 — this is the resulting band $\text{CCT}_{k,l}(i,j)$ of the Complex Cortex Transform.

In the discrete case, all Fourier transforms can be replaced by Discrete Fourier Transforms (DFT) and computed via FFT. Boundary effects need to be taken into consideration when using DFT filtering, because multiplication of discrete spectra leads to a *circular* convolution in the spatial domain. This is less of an issue with quickly decaying impulse responses of cortex filters, but may still require explicit extension of the image data beyond its support area.

4. USING CCT FOR MODELING OF MASKING

4.1 Modeling Threshold Elevation Using CCT Magnitude

Now since the Complex Cortex Transform is available for efficient separation of magnitude and phase information in the neural response, it can be incorporated into Eq. (4) to yield the new model of phase-invariant masking:

$$\begin{aligned} \widehat{m}_n^{k,l}(i,j) &= \left|\mathcal{F}^{-1}\left\{\text{cc}_{k,l} \cdot \text{csf} \cdot \mathcal{F}\left\{I(i,j)\right\}\right\}\right| \\ &= \left|\text{CCT}_{k,l}\left\{I_{\text{csf}}\right\}\right| \end{aligned} \qquad (10)$$

where I_{csf} is the input image $I(i,j)$ pre-filtered by the CSF filter.

As illustrated in Fig. 6 and Section 5, this brings the calculated Threshold Elevation map in agreement with psychophysical data suggesting that sinusoidal masks produce approximately uniform masking.

4.2 Modeling the Facilitation Effect

The publication of Daly on VDP [1] discusses the *facilitation* or the *pedestal effect* — lowering of masking thresholds when the signal and the mask have the same frequency and phase. A more extensive study of the effect is available in [4]. It shows that facilitation is quickly diminishing when frequency or phase of the signal are departing from the frequency and phase of the mask. However the effect is strong enough to be included into the HVS model proposed here. It lowers masking thresholds by up to 2.5 times in the approximate range of 0.4–1.0 JND units.

To model the facilitation effect, we are proposing an additive term to the threshold elevation formula in Eq. (3):

$$\widehat{T}_e^{k,l}(i,j) = \left(1 + k_1 \left(k_2 \left|\widehat{m}_n^{k,l}(i,j)\right|^s\right)^b\right)^{1/b} - \; s_{k,l}(i,j) \cdot f\left(\left|\widehat{m}_n^{k,l}(i,j)\right|\right),$$

where $s_{k,l}(i,j) \in [0,1]$ is the strength of facilitation effect at each point and $f(t)$ is the negative Gaussian-shaped term added to the threshold elevation around the mask contrast of 0.7 (Fig. 4). To comply with psychophysical data from [4] (Fig. 2, 4), it can be modeled as

$$f(t) = 0.6 \exp \frac{(t-0.7)^2}{-2(0.25)^2}.$$

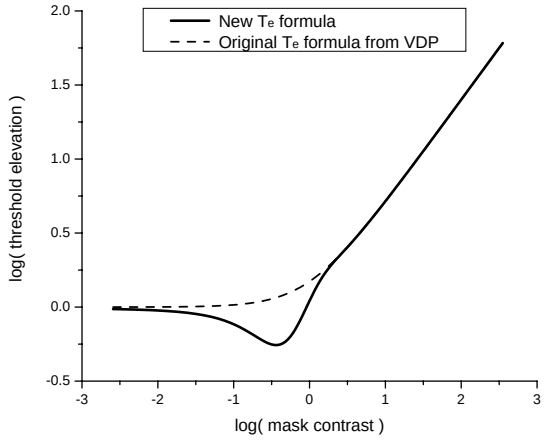


Figure 4: Threshold elevation curve $\widehat{T}_e^{k,l}(i,j)$ modeling the facilitation effect of strength $s_{k,l} = 0.5$.

According to [4], the strength of the facilitation effect depends on the proximity of frequency and phase estimates of the signal and the masker. In order to model this, we suggest using the phase information provided by CCTs of the mask and the signal:

$$s_{k,l}(i,j) = \exp \frac{\left(\phi_m^{k,l}(i,j) - \phi_s^{k,l}(i,j)\right)^2}{-2\sigma^2},$$

where $\phi_m^{k,l}(i,j) = \arg \mathrm{CCT}_{k,l}\left\{I_{\mathrm{csf}}\right\}(i,j)$ is the instantaneous phase angle of the mask CCT band, $\phi_m^{k,l}(i,j)$ is the similarly defined instantaneous phase angle of the signal CCT band, and $\sigma = 0.25$ provides about 2 times reduction in facilitation when the phase difference is $\pi/10$.

This model assumes that since both signal and mask are in the same CCT band, their frequencies are close. This may not always be true because bands of the cortex transform (and CCT) span one-octave frequency intervals. A more accurate approach to detection of same-frequency and same-phase signals may involve spatial averaging of absolute phase angle differences by averaging of $s_{k,l}(i,j)$:

$$\widehat{s}_{k,l}(i,j) = \mathcal{G}\left(s_{k,l}(i,j)\right),$$

where $\mathcal{G}$ is the Gaussian filtering operator. The radius of averaging is subject to additional psychophysical research, but our initial recommendation is to set it to 2 periods of the central frequency ρ_c of the cortex filter $\mathrm{cc}_{k,l}$.

4.3 Additional Modifications of the VDP

4.3.1 Luminance Adaptation

In [1] and [3] two luminance adaptation models are presented: the *global model* that averages the baseband luminance and the *local model* that uses pixel-wise values of luminance. In [3] the local model is used for adaptation of CSF filtering: different CSF filters are used depending on the adaptation luminance to model the psychophysical CSF data more accurately.

It can be argued that pixel-wise adaptation cannot produce stable results as the image resolution increases: the resulting CSF-filtered image may contain artifacts due to frequent switching of CSF filters. We suggest using a *spatially-smoothed* luminance for calculation of luminance adaptation. The radius of such smoothing is subject to additional research, but the initial estimate of 1° of the visual field is suggested.

4.3.2 Luminance Nonlinearity

Several referenced works describe ways for nonlinear transformation from the color model of the input image to the perceptually uniform luminance scale. An interesting observation made during our experiments may be helpful in the related research. Most images stored on personal computers do not have embedded color profiles and assume the sRGB color model. A typical way to display such images is to directly put their RGB (or luminance) values into the frame buffer of a video display adapter.

On the other hand, the most popular way of calibrating displays of personal computers (offered by many photographic web-sites) includes a black and white gradation chart with equal steps in RGB luminance, as the one in Fig. 5. The user is suggested to adjust brightness, contrast and gamma of the display until all the luminance steps in the chart become discernible and produce approximately equal subjective brightness increments.

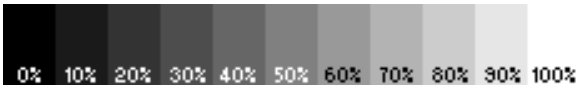


Figure 5: A typical monitor calibration chart.

It means that the user is suggested to adjust the display until *sRGB space becomes perceptually uniform* with respect to luminance. On the other hand, such a variant of "brightness" uniformity is different from uniformity in terms of JND steps, which are arguably more relevant for modeling of masking than brightness uniformity. This requires us to reconsider the nonlinear luminance transformation happening in the front end of VDP algorithms.

5. RESULTS OF MODELING

To evaluate the new model of masking, maps of threshold elevation have been computed for several test images. The source images have been generated in a perceptually uniform color space to eliminate the need for a nonlinear luminance mapping. Here we present threshold elevation maps for 2 test images that are important in psychophysical experiments.

The first mask image is a sinusoidal grating across the gradient background (Fig. 6a). Since the gradient is linear in a perceptually uniform luminance space, it is expected that the threshold elevation resulting from the grating will be spatially uniform.

Fig. 6 plots the resulting threshold elevation maps in the cortex band which contains the strongest response to the frequency of the grating. The luminance scale of the maps is stretched for easier evaluation.

As can be seen in Fig. 6b, the resulting T_e map generated by the original VDP by Daly exhibits oscillations due to inability of the cortex transform to separate phase and magnitude components of the masking stimulus.

Fig. 6c shows the modified T_e map produced by the "phase uncertainty" algorithm described by Mantiuk et. al. [3]. The algorithm smoothes variations in the threshold elevation map, but lowers the overall masking level.

Fig. 6d shows the $\widehat{T}_e$ map generated by the proposed method. It shows almost uniform elevation of the thresholds.

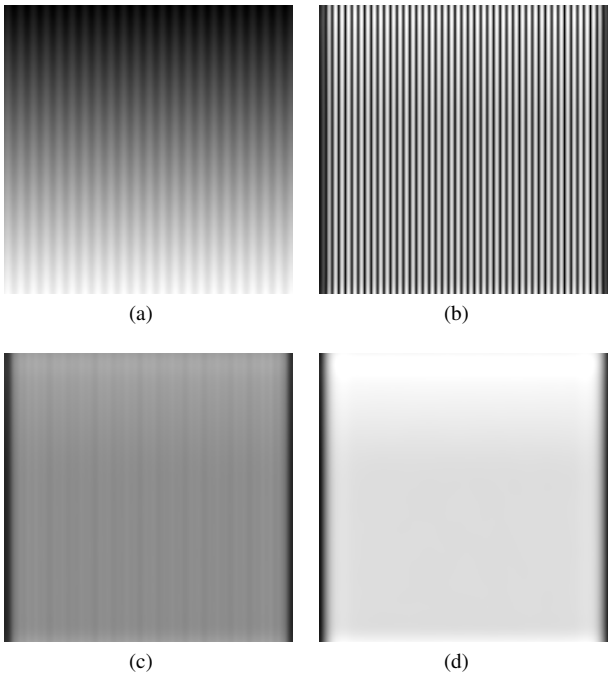


Figure 6: (a) Grating mask; (b) T_e using original VDP [1]; (c) T_e using "phase uncertainty" method from [3]; (d) $\widehat{T}_e$ using the proposed method.

The second mask image is the idealized edge image (Fig. 7a). The expected pattern of threshold elevation peaks at the edge and drops off away from the edge [9]. In Fig. 7b,c,d, the resulting T_e maps are plotted for one of the lower-frequency cortex bands with orientation corresponding to the edge orientation. Boundary effects resulting from the use of the Discrete Fourier Transform are visible around left and right boundaries of the map due to circular extension of the image.

It can be seen that the original VDP has produced the elevation map that actually has a notch at the location of the edge, flanked by a few peaks. The proposed approach for calculation of $\widehat{T}_e$ has shown the most uniform result again.

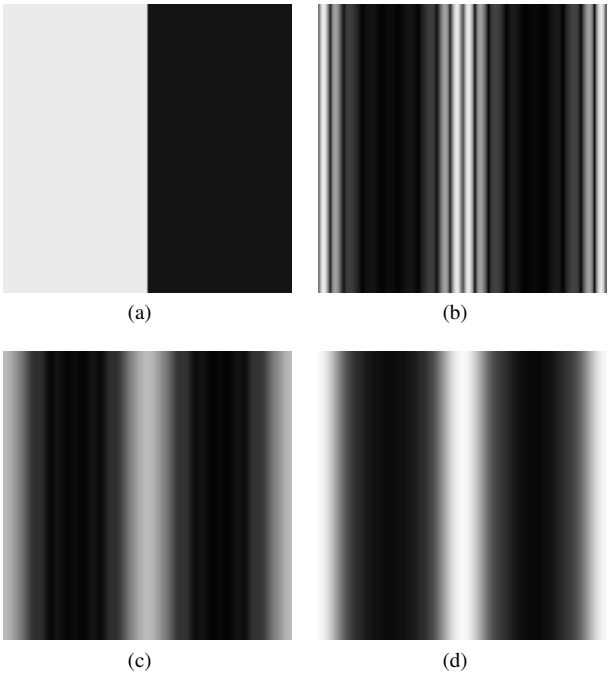


Figure 7: (a) Edge masker; (b) T_e using original VDP [1]; (c) T_e using "phase uncertainty" method from [3]; (d) T_e using the proposed method.

6. CONCLUSION

We have introduced a modification of the cortex transform using pairs of phase-complementary cortex filters. This new transform, called a Complex Cortex Transform, allows separation of magnitude and phase components of the neural response in visual cortex.

It is proposed to use the CCT for modeling of visual masking in VDP algorithms. In this work, CCT has been shown to improve consistency of threshold elevation estimates in the Visual Differences Predictor by Mantiuk/Daly. The improved VDP is also able to model the facilitation effect happening when mask and target signals are of the same frequency and phase.

In this paper, the advantages of the proposed masking model have been shown on simple, artificially generated test images. More thorough experiments need to be performed on natural images in order to assess correlation of the improved distortion measure with subjective quality data.

Acknowledgement

Author thanks Drs. Rafal Mantiuk and Scott Daly for useful comments on the paper and for sharing the source code of their VDP program for HDR images, which has been used as a base for experiments presented in this paper.

This work has been supported by RFBR grant 09-07-92000-HHC.

7. REFERENCES

[1] S. Daly, "The visible differences predictor: An algorithm for the assessment of image fidelity," *Digital Image and Human Vision*, pp. 179–206, 1993.

[2] A. Watson, "The cortex transform: Rapid computation of simulated neural images," *Computer Vision, Graphics, and Image Processing*, vol. 39, no. 3, pp. 311–327, 1987.

[3] R. Mantiuk, S. Daly, K. Myszkowski, and H.-P. Seidel, "Predicting visible differences in high dynamic range images - model and its calibration," in *Human Vision and Electronic Imaging X*, S. Daly B. Rogowitz, T. Pappas, Ed., 2005, vol. 5666, pp. 204–214.

[4] G. Legge and J. Foley, "Contrast masking in human vision," *Journal of the Optical Society of America*, vol. 7, no. 12, pp. 1458–1471, 1980.

[5] D. Pollen and S. Ronner, "Phase relationships between adjacent simple cells in the visual cortex," *Science*, vol. 212, no. 4501, pp. 1409–1411, 1981.

[6] B. Willmore, R. Prenger, M. Wu, and J. Gallant, "The berkeley wavelet transform: A biologically inspired orthogonal wavelet transform," *Neural Computation*, vol. 20, no. 6, pp. 1537–1564, 2008.

[7] M. Johansson, "The hilbert transform," M.S. thesis, Vaxjo University, 1999.

[8] J.P. Havlicek, J.W. Havlicek, N. Mamuya, and A. Bovik, "Skewed 2d hilbert transforms and computed am-fm models," in *IEEE Int. Conf. Image Processing*, 1998, vol. 1, pp. 602–606.

[9] S. Macknik, "Visual masking approaches to visual awareness," *Progress in Brain Research*, , no. 155, pp. 177–215, 2006.

ABOUT THE AUTHOR

Alexey Lukin has graduated from a Moscow State University in 2003 and received a Ph.D. degree in 2006 for his work on perceptually motivated audio signal and image processing. He is now a member of scientific staff at the Lab of Mathematical Methods of Image Processing. Alexey's interests include image processing, audio signal processing, multiresolution filter banks and adaptive methods of spectral analysis. His publications can be found at http://imaging.cs.msu.ru.

COMMENTS OF REVIEWERS

1. *Reviewer*: End of section 2: it's not clear what do "oscillations of sinusoidal mask signal" mean. Sinusoidal signal originally oscillates between zero and magnitude. But this is correct in space domain. As I remember, elevation formula should be applied to phase-frequency domain — not directly. So, for fixed $\sin(ax + n)$ there will not be any oscillations in T_e.

   Author replies: Eqs. (4) and (3) for T_e are actually adapted from [1], and later they are also used in [3]. They are based on "instantaneous" space-domain values — a shortcoming that is addressed in this paper.

2. *Reviewer*: A model of threshold elevation presented originally by Daly and referred in this paper as [1] is not the only model. More sophisticated model is presented in *Peter G.J. Barten "Contrast sensitivity of the human eye and its effects on image quality" (PIE International Society for Optical Eng., 1999)*. It reflects masking decreasing in small masking contrast areas. And mismatch of theoretical and practical data could be caused by usage of non-optimal model, not by specific issues with separation of magnitude/phase.

   Author replies: This publication of Barten describes a masking model for simple stimuli: sinusoidal gratings and noise. However it does not discuss how a complex image should be decomposed into sinusoidal components for calculation of masking at every spatial frequency. A Fourier analysis is mentioned in that publication, but it is not an acceptable means of such separation because of loss of space locality. The subject of this paper is improvement of the cortex transform — a means of separating the image into frequency subbands. Barten's masking model can be applied to the subbands produced by the proposed method.

3. *Reviewer*: A modified formula for T_e is another way for representing the facilitation effect at low masking contrasts. It seems that this is rather important to compare it with the model presented by Barten.

   Author replies: The model of Barten calculates the facilitation effect for simple stimuli. The attempt of this paper is to extend the model to complex stimuli. I agree that the developed model lacks experimental validation.

4. *Reviewer*: Section 4.2: Facilitation is accounted for in newer masking models, e.g. *A.B. Watson and J.A. Solomon "Model of visual contrast gain control and pattern masking" (Journal of the Optical Society of America, 1997)*. Facilitation is present only for stimuli of very similar spatial frequency and orientation. This rarely happens in complex images and it was the reason why Daly did not want to have it in the VDP.

5. *Reviewer*: Subsection 4.3.1: I would not agree with blurring by 1° Gaussian. The spatial adaptation mechanism probably cannot be explained by a linear filter because it originates from several mechanisms with different spatial extent and time constants. A huge part of adaptation happens in cones, so it is restricted to very small spots, definitely smaller than 1°. But I agree that the linear interpolation of several CSF-filtered images, as done in [3], is not very good either. These filters do not interpolate very well linearly.

   Author replies: I agree that mechanisms of luminance adaptation for masking are more complex than Gaussian smoothing. However Gaussian blur with a certain radius is probably better than adaptation to a single pixel of unknown size.

6. *Reviewer*: Subsection 4.3.2: Usually Gamma correction means that we estimate non-linearity of a monitor and try to compensate for it. As a result, the monitor will have approximately linear transfer function. However human perception has a non-linear feeling of physical luminance — it is proportional to the cubic root of the input value. Soww, this is actually a good question: what we should take into account and how we should do it.

Adaptive Image Deringing

Andrey V. Nasonov, Andrey S. Krylov
Laboratory of Mathematical Methods of Image Processing,
Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University,
{nasonov, kryl}@cs.msu.ru

Abstract

This paper proposes new and original adaptive image ringing detection and ringing suppression methods. Ringing detection is performed using a scale-space based computational analysis of ideal low-pass filtering of step edges. Edge width conception is introduced and is used to approximate ringing parameter. Method to choose strong isolated edges which are the best suitable edges for ringing analysis is proposed. A method of ringing level estimation for 2D images with unknown ringing parameters is introduced and used to perform real image deringing. New image quality metrics sensible to ringing artifact are introduced and used to analyze image resampling methods.

Keywords: *Ringing estimation, total variation, scale space, adaptive deringing.*

1. INTRODUCTION

Development of image enhancement methods is one of the most important image processing tasks. Ringing effect (Gibbs phenomenon) appears in images as oscillations near sharp edges. It is a result of a cut-off of high-frequency information. Ringing can appear as a result of image compression, image upsampling and other applications. An example of this effect can be seen in old video stored in analog format (Fig.1).

One of the main problems of image deringing is to detect the presence of ringing effect and to estimate the necessary ringing suppression level. But there is no algorithm which estimates ringing level in the general case.

There are image ringing estimation algorithms to measure ringing effect for a specific problem. In [1] wavelet decomposition is used and ringing effect is measured for JPEG-2000 compression as a difference between correlations of neighbor coefficients of different wavelet subbands. The problem of image deringing after JPEG-2000 compression is also considered in [2], [3].

Some metrics were developed to control image deringing as a post-processing. Regularization parameter estimation for image deringing using MAP approach is proposed in [4]. For the problem of image deringing after resampling, regularization parameter is estimated using information on the initial low resolution image [5]. In [6], the ringing metrics is defined as maximum of the differences between pixel values of the reference image and the processed image in the edge neighborhood. The size of this neighborhood is fixed a priori. In [7], the presence of ringing effect is detected by comparing the directions of image gradients at different scales.

The work [8] does not introduce a ringing estimation method, but it presents an algorithm to find regions where the ringing effect is the most visible. It is based on luminance masking and texture masking as typical for the human visual system.

In [9], a no-reference ringing detection method using Gabor filtering was suggested. It shows good results but it fails if image contains periodic structures like fence, geometrical textures, etc.

Figure 1: A video frame with a strong ringing effect.

In this article, we suggest new general ringing estimation algorithm based on total variation (TV) control. The TV was first used in image enhancement by Rudin, Osher and Fatemi [10] for image denoising. General relations between TV and ringing effect can be found in [11].

The rest of the paper is organized as follows. In section 2, we perform scale-space analysis of ringing effect and present a method of ringing level estimation for 1D edges with known ringing parameters. In section 3, we introduce a method of ringing level estimation for 2D images with unknown ringing parameters and an algorithm of strong isolated edges selection to choose the best suitable edges for ringing analysis. In section 4, we propose image quality metrics to compare different image restoration methods. In section 5, we suggest methods of ringing suppression for the problem of ringing reduction after interpolation and for the problem of blind image deringing. Section 6 concludes the paper.

2. 1D EDGE RINGING EFFECT ANALYSIS

We start from experimental analysis of one-dimensional edges with manually generated ringing artifact with known ringing parameters. The analysis consists of Total Variation (TV) calculation of edges convolved with Gauss filter with different radiuses.

2.1 Total Variation

In one-dimensional case, Total Variation functional is defined as

$$TV(f) = \int_{-\infty}^{\infty} |f'(x)|dx.$$

In a discrete case $-\infty < \ldots < x_{k-1} < x_k < \ldots < \infty$, it looks as:

$$TV(f) = \sum_{k=-\infty}^{\infty} |f(x_k) - f(x_{k-1})|.$$

We also consider weighted TV (WTV) with weight function $w(x)$:

$$TV(f, w) = \int_{-\infty}^{\infty} |f'(x)|w(x)dx.$$

In the discrete case it looks as

$$TV(f,w) = \sum_{k=-\infty}^{\infty} |f(x_k) - f(x_{k-1})| w\left(\frac{x_{k-1}+x_k}{2}\right).$$

2.2 Ringing Model

We consider the problem of ringing level estimation for ideal step edge (ISE)

$$\overline{f}(x) = \begin{cases} 1, & x \geq 0, \\ 0, & x < 0. \end{cases}$$

In the discrete case, $TV(\overline{f}) = 1$ for any grid.

Ringing effect can be generated by ideal low-pass filter which truncates high frequency data. We implement it using sinc interpolation of ideal step edge function $\overline{f}(x)$ given at the discrete set $\{x_k\}, x_k = dk + \frac{d}{2}$:

$$f_d(x) = \sum_k \overline{f}(x_k) \mathrm{sinc} \frac{x - x_k}{d}, \qquad (1)$$

where $\mathrm{sinc}(x) = \frac{\sin \pi x}{\pi x}$. We call here the value d as ringing half-period. We call here the value d as quasiperiod of ringing oscillations. This value is in inverse proportion to the signal bandwidth.

Real edges are corrupted by noise, so this fact is to be considered too. In this work, we analyze the case of additive uniform noise $f_{d,n}(x) = f_d(x) + \xi_n(x)$, $f_n(x) = \overline{f}(x) + \xi_n(x)$, where $\xi_n(x)$ is uniformly distributed random function which values are in $(-n/2, n/2)$ range. We consider $n \leq n_0$, where n_0 is the maximum noise value.

2.3 Scale-space TV Analysis

We performed an experimental analysis of TV of edges with different quasiperiods d at different scales σ. Real edges are not infinite. To take into account only several first ringing oscillations, we use weighted TV with Gaussian weight $w_{\alpha d}(x) = e^{-\frac{x^2}{2(\alpha d)^2}}$, where α controls the number of considered ringing oscillations. To find the differences between edges with ringing effect and edges without it, we analyze the functional

$$TV(f, \sigma, w_{\alpha d}) = TV(f * G_\sigma, w_{\alpha d}),$$

where $f * G_\sigma$ is a convolution of f with Gauss filter with radius σ.

To design an algorithm to find the value σ_0 that best discriminates between edges with ringing artifact and edges without ringing artifact for fixed d, fixed maximum noise level n_0 and given parameter α, the following analysis has been performed:

We generated a large number (about 1000) of edges with and without ringing effect and with random noise levels $0 \leq n \leq n_0$ for the analysis. For every σ, we found the maximal value of TV functional for step edges without ringing effect

$$g_*(\sigma, d) = \max_{f_n} TV(f_n, \sigma, w_{\alpha d})$$

and the minimal value of TV functional for step edges with ringing effect

$$g^*(\sigma, d) = \min_{f_{d,n}} TV(f_{d,n}, \sigma, w_{\alpha d}).$$

The higher noise level is, the higher is the value of edge TV. Thus, to calculate $g_*(\sigma, d)$, we used only step edges f_n with the maximum noise level $n = n_0$

$$g_*(\sigma, d) = \max_{f_{n_0}} TV(f_{n_0}, \sigma, w_{\alpha d})$$

and for $g^*(\sigma, d)$ calculation, we uses only step edge f_d without noise

$$g^*(\sigma, d) = TV(f_d, \sigma, w_{\alpha d}).$$

A typical result for $d = 10$, $n_0 = 0.1$ and $\alpha = 3$ is shown in Fig.2. It can be seen that for a given ringing quasiperiod d there is a set of σ which splits the edges by TV value. If we choose σ in the marked range, we can distinguish edges with ringing artifacts from edges without it.

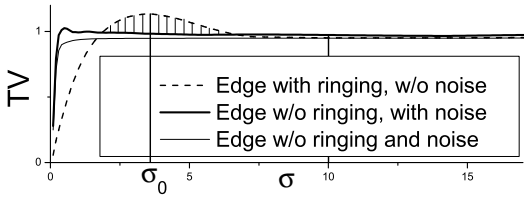


Figure 2: Scale-space TV analysis for the step edge with quasiperiod $d = 10$, maximum noise level $n_0 = 0.1$ and $\alpha = 3$ (number of oscillations ~ 3). Value $g_*(\sigma, d)$ is the maximal value of the TV of the step edge without ringing effect (thick solid line), $g^*(\sigma, d)$ is the minimal value of the TV of the step edge with ringing effect (dash line).

So we found the scale $\sigma_0(d)$ that corresponds to the maximal gap between $g^*(\sigma, d)$ and $g_*(\sigma, d)$:

$$\sigma_0(d) = \arg\max_{\sigma} (g_d^*(\sigma, d) - g_*(\sigma, d)).. \qquad (2)$$

The calculated values of σ_0 for $n_0 = 0.1$ and $\alpha = 3$ for different d are shown in Fig.3. For low quasiperiods d, there is no σ such that $g^*(\sigma, d) > g_*(\sigma, d)$, another words, we cannot distinguish the edges with ringing effect from the edges without ringing effect for given n_0 and α by analyzing the TV. Ringing effect can be reasonable detected by the proposed method only for $d > d_{min}$. The value d_{min} depends on noise level n_0: for higher noise values d_{min} is greater. The analysis of d_{min} is a subject of future work.

The function $\sigma_0(d)$ is close to linear function and we use an approximation

$$\sigma_0(d) = m_{\alpha, n_0} d.$$

For $\alpha = 3$ and $n_0 = 0.1$, good approximation is $m_\alpha = 0.35$.

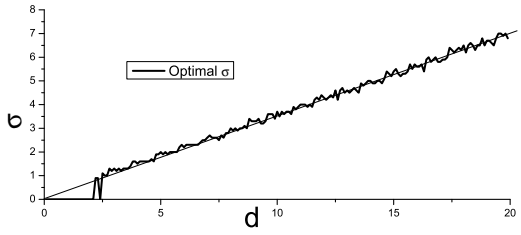


Figure 3: The results of σ_0 calculation (2) for $n_0 = 0.1$ and $\alpha = 3$ for different d. Thin line is a linear approximation of function $\sigma_0(d)$.

Experiments also show that m_α does not depend too much on α for reasonable range of α ($2 < \alpha < 5$). For the case $\alpha \leq 2$ the above algorithm does not give stable results.

2.4 Edge Ringing Level Estimation

This enables us to consider the value

$$R_E^*(f,d) = TV(f, m_\alpha d, w_{\alpha d})$$

as edge ringing level. For the case of edges with an arbitrary height, we perform a normalization. For $\sigma = d$, the values $g_*(\sigma)$ and $g_d^*(\sigma)$ are close, so it is natural to divide the value R_E^* by $TV(f, d, w_{\alpha d})$. Ringing value takes the form:

$$R_E(f,d) = \frac{TV(f, m_\alpha d, w_{\alpha d})}{TV(f, d, w_{\alpha d})}.$$

To make a decision about the presence of ringing effect, we compare the calculated ringing level R_E with threshold functions

$$G_*(d) = g_*(m_\alpha d, d),$$
$$G^*(d) = g^*(m_\alpha d, d).$$

If $R_E \geq G^*(d)$, we assume that the edge has ringing artifact. If $R_E \leq G_*(d)$, we assume that the edge does not have ringing artifact. The decision in the case $G_*(d) < R_E < G^*(d)$ needs additional analysis for each specific image class.

The calculated threshold functions $G_*(d)$ and $G^*(d)$ for $n_0 = 0.1$ and $\alpha = 3$ for different d are shown in Fig.4.

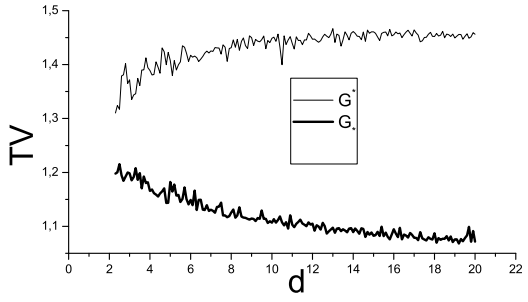


Figure 4: Threshold functions $G_*(d)$ and $G^*(d)$ for $n_0 = 0.1$ and $\alpha = 3$ for different d.

2.5 Ringing Quasiperiod Estimation and Edge Width Analysis

In real situations, we do not know the information about ringing quasiperiod d. This value has to be estimated, but we do not directly estimate d. Instead of this, we calculate edge width w and take this value as an approximation of ringing quasiperiod $d = w$. Using edge width instead of ringing quasiperiod enables to perform ringing level estimation even if ringing artifact is absent. Thus, edge width is considered as possible ringing quasiperiod.

The term 'edge width' does not have a certain definition. The simplest approach used in [1] is to find local minimum and local maximum near the edge center. This approach does not provide stable results for blurred and noisy edges. In [12], the edge is modeled by a special function, but it does not fit our needs.

For one-dimensional edge

$$\overline{f}(x) = \begin{cases} f_0, & x \leq x_0, \\ f_0 + \frac{(f_1 - f_0)(x - x_0)}{x_1 - x_0}, & x_0 < x < x_1, \\ f_1, & x \geq x_1 \end{cases} \qquad (3)$$

we define edge width as $w(\overline{f}) = x_1 - x_0$.

To define edge width for an arbitrary edge $f(x)$, we approximate it by the edge $\overline{f}(x)$ (3). We seek for minimum f_0 and maximum values f_1 of $f(x)$ in a neighborhood of the edge center, reducing for simplicity the edge to the case $f_0 = 0$, $f_1 = 1$.The size of this neighborhood is chosen a priori and represents the maximum considered edge width.

Next we seek for coordinates of intersections of $y = f(x)$ with $y_0 = 1/4$ and with $y_1 = 3/4$, draw a line through these points and find x_0 and x_1 as it is shown in Fig.5. In the case of multiple intersections we take the average of intersection points. We consider the obtained value as the edge width estimation.

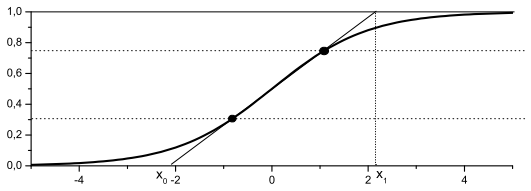


Figure 5: Edge width estimation illustration. For the shown case, the estimated edge width is 4.3 pixels.

3. 2D IMAGE RINGING LEVEL ESTIMATION

Ringing effect is located near sharp edges. It is natural to make a decision about ringing effect by analyzing the areas near sharp edges. So, we estimate image ringing level using the analysis of ringing level of 1D normal cross-sections of image edges obtained by an edge detection algorithm. General purpose edge detection algorithms do not give suitable edges for ringing level analysis. We pose the following statements for edge detection:

1. In the proposed ringing level estimation method, we consider ringing oscillations as parts of the main edge. Thus, ringing oscillations must be distinguished from the main edges.

2. If two sharp edges are positioned closely, ringing effect is interfered. We consider only distant edges to minimize the interference effect. Thus, the edges must be distant enough from each other.

3.1 Basic Edges Selection

We introduce an edge selection algorithm that follows the posed above statements. We call it basic edges selection.

To satisfy the first condition, we use the results of scale-space ringing analysis. The experiments have shown that for σ greater than edge ringing quasiperiod d, the value $TV(f_d * G_\sigma)$ becomes close to 1. It means that if we perform Gauss filtering with radius σ, ringing effect of edges with quasiperiod $d < \sigma$ will be suppressed and edge detection algorithms will not detect false edges along main edges. Thus, we perform Gauss filtering with $\sigma = d$ before edge detection. We use Canny edge detector [13] which uses Gauss filtering.

Parameter d is unknown. To estimate d, we perform Canny edge detection with predefined parameters and analyze edge widths. We consider the case of uniform high-frequency truncation for the whole image. Because d is defined by the cut-off frequency, it is constant for every edge. We approximate d using the analysis of edge widths $w_P = w(f_P(x))$, where $f_P(x)$ is edge normal cross-section function which corresponds to edge point P. For ideal edge

with ringing artifact without noise (1), $w(f_d) = d$, but real edges are not ideal. Blurred edges results in $w_P > d$, noisy edges introduce errors in w_P estimation. To estimate d with taking into account noise and blur outliers, we use the approach from our previous work [14] based on the analysis of density function of w_P distribution.

Using the estimation for ringing quasiperiod d, we take only the edges with w_P close to d: $|\frac{w_P}{d} - 1| > q$ to remove blurred edges and noisy edges with wrongly estimated edge width. Good results are achieved with $q = 0.2$.

Second condition is satisfied by calculating the distance between the edges. To make this, we construct a skeleton M_S of non-edge area and calculate the distance $\rho_P = \rho(P, M_S)$ from edge points to the skeleton. Next, we discard the edge points P with $\rho_P < \alpha d$, where α controls the number of considered ringing oscillations. The distance is calculated using Euclidean Distance Transform (EDT). The details of EDT and skeletonization are presented in following sections.

3.2 Image Ringing Level Estimation

Basic edges selection algorithm gives a set of basic edge points M_{BE} of sharp distant edges. Edge cross-sections f_P have edge width close to the estimated image ringing quasiperiod d. We define image ringing level as the average value of edge ringing values of these edge cross-sections:

$$R_E = \frac{1}{\|M_{BE}\|} \sum_{P \in M_{BE}} R_E(f_P, d),$$

where $\|M_{BE}\|$ is a number of basic edge points.

To improve the algorithm, we take only the first N edge points with maximal gradient value. Eliminating the edges with low gradient value results makes the algorithm more stable to noise and reduces the number of calculations. We use $N = 50$.

The effectiveness of the proposed algorithm of ringing level estimation for synthetic images is demonstrated in [14].

3.3 Euclidean Distance Transform

The distance transform (DT) is a general operator forming a basis of many methods in computer vision and geometry. It finds for each image pixel its smallest distance to the region of interest M:

$$D(p) = \min_{q \in M} \rho(p, q). \quad (4)$$

The simplest metrics used in DT (4) are l_1 metrics (Manhattan distance) $\rho_1(p, q) = |p_x - q_x| + |p_y - q_y|$ and l_∞ (checkerboard distance) $\rho_\infty(p, q) = \max(|p_x - q_x|, |p_y - q_y|)$. The DT for these metrics is calculated easily. These metrics do not preserve distance values during image rotation. A more natural metrics is l_2 metrics (Euclidean distance) $\rho_2(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$. DT with Euclidean distance is called the Euclidean distance transform (EDT). Many fast algorithms for the EDT with linear complexity were developed in two past decades. A survey of these algorithms is given in [15]. We use Meijster's algorithm.

Numerical example of the EDT is shown in Fig.6.

3.4 Skeletonization

The idea of skeletonization is to construct a thin version of a shape that is equidistant to its boundaries.

We perform non-iterative skeletization using EDT. As the area of interest M, we take the edges, and then find all local maximum points of the EDT in horizontal, vertical and two diagonal directions. These points provide a good approximation of the skeleton. Additional skeleton post-processing like regularization or pruning does not significantly change the results of our algorithm of edges selection.

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0
0	0	1	1	1	1	1	0	0
0	0	1	1	1	1	1	0	0
0	0	1	1	1	1	1	0	0
0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0
0	0	1	2	4	2	1	0	0
0	0	1	4	8	4	1	0	0
0	0	1	2	4	2	1	0	0
0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0

Figure 6: Numerical example of the Euclidean distance transform (EDT). Left figure: input data, black pixels (zeroes) form the region of interest. Right figure: the squares of the Euclidean distance of each pixel to the nearest black pixel.

3.5 Edge Selection Results

The processing algorithm is step-by-step illustrated for the images of Lena and Fish in Fig. 7-12. Ideal low-pass filter (high frequency data cut-off, first $1/8$ coefficients of Fourier series were retained) was used to produce ringing artifact. The calculated ringing quasiperiod d was found equal to theoretical quasiperiod value in both cases: $d = 8$.

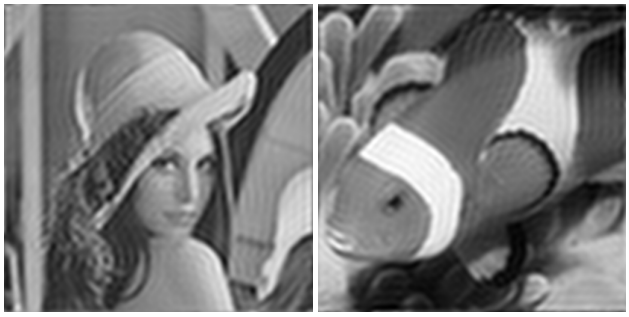


Figure 7: Input images with artificially added ringing artifact using high frequency data cut-off, first $1/8$ coefficients of Fourier series were retained.

Figure 8: Results of Canny edge detector with $\sigma = 1$ to estimate ringing quasiperiod d for the input images from Fig. 7. White edges are important edges with high gradient value. Grey edges are edges with lower gradient values. Both grey and white edges are used for skeletonization, but only white edges will be used for basic edge selection.

Figure 9: Results of Canny edge detector with Gauss radius taken in accordance with ringing quasiperiod $\sigma = d = 8$.

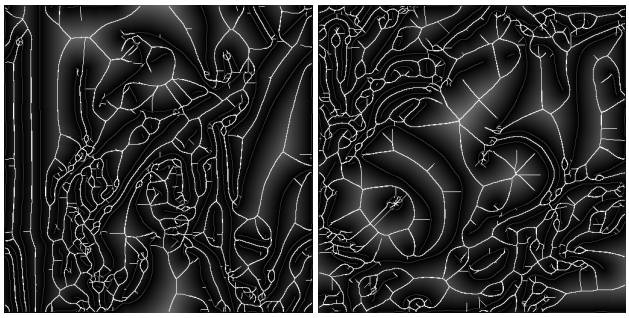


Figure 10: Skeletonization of the result of Canny edge detector from Fig.9. Background intensity level represents the distance to the edges: the higher is the intensity, the more distant to the edges is the point (EDT picture). Grey lines are the edges. White lines belong to the skeleton.

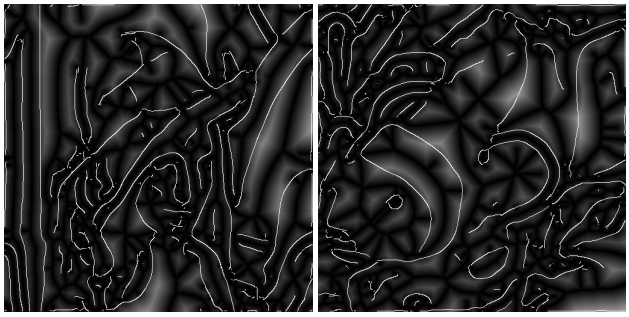


Figure 11: Calculation of the Euclidean distance from the skeleton (Fig. 10) to the edges (Fig. 9). Background intensity level represents the distance to the skeleton (EDT picture). White lines are the edges.

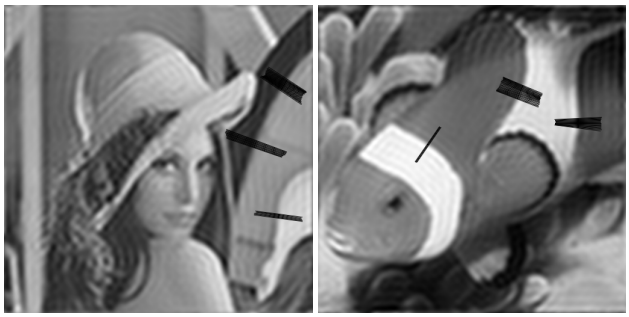


Figure 12: Results of basic edge selection for the input images. The black lines show the selected edges. The length of the line corresponding to a specific edge is equal to 6 edge widths. The number of selected edge points $N = 50$.

4. IMAGE RESTORATION QUALITY ANALYSIS METRICS

Standard metrics based on whole image square error calculation like MSE, PSNR do not correlate good enough with the perceptual image quality. As an example, ringing effect in textured areas is not noticeable while ringing effect near sharp isolated edges is annoying. We introduce an image metrics aimed at quality measurement of image restoration methods like image interpolation or image deringing in edge and edge neighborhood areas.

4.1 Basic Edges Points RMSE

To estimate the quality of edge restoration, we calculate RMSE (root of mean square error) in small neighborhood of edges points obtained by basic edges selection method:

$$BEP(u,v) = \sqrt{\frac{\sum_{M_{BEP}}(u_{i,j} - v_{i,j})^2}{\|M_{BEP}\|}},$$

where u is the reference image, v is the restored image, M_{BEP} is the edge area of the reference image u, $\|M_{BEP}\|$ is the number of points in M_{BEP}. The edge area M_{BEP} is constructed using morphological dilation of basic edges points set M_{BE} with circular structuring element with radius w:

$$M_{BEP} = \{P | \rho(M_{BE}, P) \leq w\},$$

where w is edge width. High value of BEP means that the image restoration method badly reconstructs edges, for example, introduces artifacts like blur or aliasing (jagged edges).

An example of area M_{BEP} to calculate BEP is shown in Fig.13.

4.2 Basic Edges Neighborhood RMSE

We also calculate RMSE in areas where ringing effect is the most likely to appear:

$$BEN(u,v) = \sqrt{\frac{\sum_{M_{BEN}}(u_{i,j}-v_{i,j})^2}{\|M_{BEN}\|}},$$

where M_{BEN} satisfies the condition

$$M_{BEN} = \{P | \rho(M_{BE}, P) > w, \rho(M_E, P) < \alpha w\}.$$

Here the set M_E consists of all edge points obtained by Canny edge detector in basic edges selection method. It contains all edge points from M_{BE} and edge points that do not pass the basic edges conditions and thus do not belong to basic edges.

High value of BEN means that the image restoration method works bad in edge neighborhood, for example, introduces ringing artifact.

An example of area M_{BEN} to calculate BEN is shown in Fig.13.

Figure 13: Illustration for BEN and BEP metrics. Left image is the source image. Middle image is a result of Canny edge detection. White edges are strong edges which pass the high threshold of Canny method, grey edges are weak edges which pass the low threshold. Right image shows the areas of calculation of BEP and BEN metrics. White areas are BEP areas, grey areas are BEN areas.

4.3 Image Restoration Methods Comparison

The quality of image restoration methods can be expressed in terms of the proposed basic edgse metrics. But if one method shows better BEP value, but worse BEN than another method, there is a problem to choose of the best method.

Here we introduce two methods of overall quality estimation of image restoration methods.

1. The first method constructs a relative image quality score for a set of image restoration methods. This is essential for the problem of choosing the best parameter of restoration method. Consider the values BEP_k and BEN_k as corresponding metrics values for k-th image restoration method. Next we find the maximal and minimal values of BEP_k and BEN_k:

$$BEP_* = \min_k BEP_k, \qquad BEP^* = \max_k BEP_k,$$
$$BEN_* = \min_k BEN_k, \qquad BEN^* = \max_k BEN_k.$$

We use the following rules to construct image quality score (basic edges relative quality, $BERQ$):

a. $BERQ = 0$ for $BEP = BEP_*, BEN = BEN_*$;

b. $BERQ = 1$ for $BEP = BEP^*, BEN = BEN_*$;

c. $BERQ = 1$ for $BEP = BEP_*, BEN = BEN^*$;

d. If for two methods $BEP_1 < BEP_2$ and $BEN_1 < BEN_2$, then $BERQ_1 < BERQ_2$.

We use the following definition of $BERQ$:

$$BERQ = \frac{BEP - BEP_*}{BEP^* - BEP_*} + \frac{BEN - BEN_*}{BEN^* - BEN_*}.$$

The lower is the value of $BERQ$, the better is image quality.

2. The second method is calculation of image quality value which does not depend on considered restoration methods. This problem is raised if we have to choose the best method from only two methods. The main problem is to choose the balance between BEP and BEN. We use logarithmic approach

$$BELQ = \log_2 BEP + \log_2 BEN.$$

For better presentation, we suggest BEQ metrics which is equal to $BELQ$ minus constant value $(\log_2 BEP_* - \log_2 BEN_*)$:

$$BEQ = \log_2 \frac{BEP}{BEP_*} + \log_2 \frac{BEN}{BEN_*}$$

where BEP_* and BEN_* are the normalization constants. These constants do not affect the difference between BEQ values of different images. We choose BEP_* and BEN_* as the minimal values of BEP and BEN respectively of the results of image restoration methods for the given image.

4.4 Applications to Image Interpolation

We demonstrate use of the proposed metrics by comparison of a pair of interpolation methods. We took sinc interpolation and a nonlinear combination of sinc and bicubic methods. The combination is based on weighted sum of sinc and bicubic methods, weight coefficients are functions of local TV of source and sinc interpolated images [16].

The results of image interpolation of 'house' image are shown in Fig.14, numerical results are presented in Tab.1.

Figure 14: Interpolation of 'house' image using sinc interpolation and combination of sinc and bicubic methods.

It can be seen that BEP and BEN metrics correlates with the observed quality of the considered interpolation methods: sinc interpolation shows a little bit better results in edge areas while combined method does not introduce ringing artifact in edge neighborhood. Despite MSE is better for sinc interpolation, BEQ corresponds to overall image quality.

More results and details about combined resampling method can be found in [16].

	MSE	BEP	BEN	BEQ
Sinc interpolation	109.5	14.124	4.426	0.181
Sinc + Bicubic	112.1	14.181	3.970	0.029

Table 1: The values of metrics for the images from Fig. 14.

5. IMAGE RINGING SUPPRESSION

We consider the problem of image ringing suppression in two statements: image deringing after resampling and blind image deringing.

5.1 Image Deringing after resampling

Image resampling (interpolation) is a reconstruction of a discrete image on a denser grid. In image deringing after resampling we use the fact that the TV of ideal step edge is constant for any grid. Real images have fractal nature, the TV of real images decreases after downsampling. But most of the resampling algorithms do not take this fact into consideration. So, if after upsampling the TV increases, we assume that it is caused by ringing artifact.

The main idea of the proposed method is to project the upsampled image into the convex set of images with bounded TV:

$$z_R = \arg\min_{z \in M} \|z - z_I\|^2,$$

where $M = \{z : TV(z) \leq sTV(u)\}$, z_I is the interpolated image, u is the low-resolution image, s is the scale factor. The TV is multiplied by s in 2D case, because the number of rows and columns is proportional to s.

More detailed description of this method is presented in [5].

We illustrate the proposed method of image deringing after resampling with image resampled by regularization method [17] with small regularization parameter in Fig.15.

Figure 15: Ringing estimation for image deringing after resampling. Left image: source image upsampled by pixel replication, $R_E = 1.05$. Middle image: image interpolated by regularization-based method with low regularization [17], $R_E = 1.40$. Right image: interpolated image postprocessed by deringing after resampling, $R_E = 1.06$.

5.2 Blind Image Deringing

If the low-resolution image is unknown, we use the proposed ringing metrics to a posteriori control image ringing level. Ringing reduction is performed using the regularization method based on Tikhonov regularization [18]:

$$z_R(\gamma) = \arg\min_z \left(\|z - u\|^2 + \gamma TV(z) \right),$$

where u is the given image with ringing artifact, $\gamma > 0$ is the regularization parameter. The regularization functional is consecutively minimized with increasing γ until the ringing level R_E of the result image $z_R(\gamma)$ is below the ringing threshold G^*.

The results of blind image deringing are shown in Fig.16

Figure 16: Blind image deringing. Left image: original image with ringing effect, $R_E = 1.40$. Right image: the result of automatic deringing with ringing level control, $R_E = 1.18$.

6. CONCLUSION

Adaptive image ringing detection and ringing suppression methods basing on the scale-space ringing effect analysis were developed. Method to choose strong isolated edges which are the best suitable edges for ringing analysis was proposed and used to estimate ringing level for 2D images. New image quality metrics sensible to ringing artifact were introduced. The tests showed that these metrics are promising to be widely used in image enhancement tasks.

Future work on the proposed ringing level estimation method includes additional refinement of the ringing criterion for different image classes. More effective estimation of ringing quasiperiod and edge width is under investigation.

The work was supported by federal target program "Scientific and scientific-pedagogical personnel of innovative Russia in 2009-2013" and RFBR grant 09-07-92000-HHC.

7. REFERENCES

[1] Madhuri Khambete and Madhuri Joshi, "Blur and ringing artifact measurement in image compression using wavelet transform," *Proceedings of World Academy of Science, Engineering and Technology*, vol. 20, pp. 183–186, 2007.

[2] Tao Wang and Guangtao Zhai, "Jpeg2000 image postprocessing with novel trilateral deringing filter," *Optical Engineering*, vol. 47, pp. 027005, 2008.

[3] Jinyong Fang and Jun Sun, *Advanced Intelligent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques*, chapter Ringing Artifact Reduction for JPEG2000 Images, pp. 1026–1034, Springer Berlin Heidelberg, 2007.

[4] Seungjoon Yang, Yu hen Hu, Truong Q. Nguyen, and Damon L. Tull, "Maximum-likelihood parameter estimation for image ringing-artifact removal," *IEEE Trans. Circuits Syst. Video Technol*, vol. 11, pp. 963–973, 2001.

[5] Andrey Krylov and Andrey Nasonov, "Adaptive total variation deringing method for image interpolation," *Proceedings of ICIP'08*, pp. 2608–2611, 2008.

[6] Pina Marziliano, Frederic Dufaux, Stefan Winkler, and Touradj Ebrahimi, "Perceptual blur and ringing metrics: application to jpeg2000," *Signal Processing: Image Communication*, vol. 19, pp. 163–172, 2004.

[7] T. Kartalov, Z. A. Ivanovski, L. Panovski, and L. J. Karam, "An adaptive pocs algorithm for compression artifacts removal," *9th International Symposium on Signal Processing and Its Applications*, pp. 1–4, 2007.

[8] Hantao Liu, Nick Klomp, and Ingrid Heynderickx, "Perceptually relevant ringing region detection method," *16th European Signal Processing Conference Proceedings*, 2008.

[9] Bo-Xin Zuo, "Perceptual ringing metric to evaluate the quality of images restored using blind deconvolution algorithms," *Optical Engineering*, vol. 48, no. 3, pp. 037004, 2009.

[10] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.

[11] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1999.

[12] PeiFeng Zeng and Tomio Hirata, "Distance map based enhancement for interpolated image," *Lecture Notes in Computer Science*, vol. 2616, pp. 429–449, 2003.

[13] J. Canny, "A computational approach to edge detection," *IEEE Trans. PAMI*, vol. 8, pp. 679–714, 1986.

[14] A. V. Nasonov and A. S. Krylov, "Scale-space method of image ringing estimation," *To appear in ICIP'2009*, 2009.

[15] Ricardo Fabbri, Luciano da F. Costa, Julio C. Torelli, and Odemir M. Bruno, "2d euclidean distance transforms: a comparative survey," *ACM Computing Surveys*, vol. 40, no. 1, 2008.

[16] A. S. Krylov, A. V. Nasonov, and A. A. Chernomorets, "Combined linear resampling method with ringing control," *Proceedings of Graphicon'2009*, 2009.

[17] A. Lukin, A. Krylov, and A. Nasonov, "Image interpolation by super-resolution," *Proceedings of GraphiCon'2006*, pp. 239–242, 2006.

[18] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill Posed Problems*, WH Winston, Washington DC, 1977.

About authors

Andrey V. Nasonov is member of scientific staff of CMC MSU.
Email: nasonov@cs.msu.ru

Andrey S. Krylov is associated professor, Head of the Laboratory of Mathematical Methods of Image Processing, CMC MSU.
Email: kryl@cs.msu.ru

Joint Demosaicing and Arbitrary-Ratio Resizing Algorithm for Bayer Color Filter Array Based on DCT Approach

Kuo-Liang Chung[a,*], Wei-Jen Yang[b], Mei-Ju Chen[a], Wen-Ming Yan[b], and Chiou-Shann Fuh[b]
[a]Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology, Taipei, Taiwan R. O. C.
{k.l.chung, M9715007}@mail.ntust.edu.tw.
[b]Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan R. O. C.
{f93035, ganboon, fuh}@csie.ntu.edu.tw.

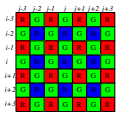


Figure 1: The Bayer CFA structure.

Abstract

This paper presents a new joint demosaicing and arbitrary-ratio resizing algorithm for mosaic images. By using the adaptive heterogeneity projection masks and Sobel- and luminance estimation-based masks, the extracted edge information is utilized to construct the fully populated green color plane, green-red color difference plane, and green-blue color difference plane. Then, based on the composite cosine transform (DCT) technique, the above three constructed planes are resized to the arbitrary sized ones. Further, the resized red and blue color planes are constructed. Based on twenty-four popular testing mosaic images, the proposed resizing algorithm has better image quality performance when compared with two native algorithms.

Keywords: *Arbitrary-ratio resizing algorithm, Color difference, Color filter array, DCT, Demosaicing algorithm, Digital cameras.*

1. INTRODUCTION

Most digital cameras use a single sensor array to capture the color information based on Bayer color filter array (CFA) structure [1] which is depicted in Figure 1. In Bayer CFA structure, each pixel in the captured image has only one of the three primary colors and this kind of image is called the mosaic image. Because the green (G) color plane is the most important factor to determine the luminance of the color image, half of the pixels in Bayer CFA structure are assigned to G color plane; the red (R) and blue (B) color planes share the remaining parts evenly.

Since each pixel in the mosaic image has only one color components, the two missing color components for one color pixel should be recovered as best as possible and such a recovering process is called the demosaicing process which has been studied extensively [2, 3, 4, 5]. Besides the demosaicing issue, how to resize mosaic images is another important research issue. The terms "resize" and "zoom" are used exchangeably. Some resizing algorithms for mosaic images have been developed [6, 7]. Unfortunately, all of them only focus on the quad-zooming process.

In this paper, a new joint demosaicing and arbitrary-ratio resizing algorithm for mosaic images is presented. Utilizing the the adaptive heterogeneity projection masks and the Sobel- and luminance estimation-based (SL-based) masks [4], the edge information can be extracted from the mosaic image directly and accurately. Then, based on the color difference concept and the composite length DCT [8], the mosaic image can be demosaiced and resized to obtain arbitrary-ratio sized full color images. Based on twenty-four popular testing mosaic images, the proposed algorithm has better image quality performance when compared with two native algorithms.

2. THE PROPOSED NEW JOINT DEMOSAICING AND ARBITRARY-RATIO RESIZING ALGORITHM FOR MOSAIC IMAGES

Before performing the proposed resizing algorithm, the tuned horizontal heterogeneity projection value $HP'_H(i,j)$ and the tuned vertical heterogeneity projection values $HP'_V(i,j)$ are computed by Adaptive heterogeneity projection, and then the horizontal gradient response $\Delta I^H_{dm}(i,j)$, the vertical gradient response $\Delta I^V_{dm}(i,j)$, the $\frac{\pi}{4}$-diagonal gradient response $\Delta I^{\frac{\pi}{4}}_{dm}(i,j)$, and the $\frac{-\pi}{4}$-diagonal gradient response $\Delta I^{\frac{-\pi}{4}}_{dm}(i,j)$ are computed by SL-based masks [4]. Our proposed algorithm consists of the following three stages: (1) interpolating the mosaic G plane to construct the fully populated G plane by using the edge-sensing interpolation estimation; (2) interpolating the mosaic G-R and G-B color difference planes to construct the fully populated G-R and G-B color difference planes, respectively; (3) resizing the three constructed planes mentioned above to obtain the arbitrary-ratio size ones, and then based on the three resided planes, recovering the resized R and B planes to obtain arbitrary-ratio resided full color image.

2.1 Stage 1: Interpolating the mosaic G plane

In this subsection, the interpolation for the mosaic G plane I^g_{mo} to obtain the fully populated G plane I^g_{dm} by using the edge-sensing demosaicing algorithm is presented.

For exposition, let us take the central pixel at position (i,j) in Figure 1 as the representative to explain how to estimate the G color value $I^g_{dm}(i,j)$ from its four neighboring pixels with movement $\Omega_g = \{(x,y)|(x,y) = (i \pm 1, j), (i, j \pm 1)\}$. First, according to the tuned horizontal heterogeneity projection value $HP'_H(i,j)$ and vertical heterogeneity projection value $HP'_V(i,j)$ of the current pixel, three cases, namely the horizontal variation as shown in Figure 2(a), the vertical variation as shown in Figure 2(b), and the other variations as shown in Figure 2(c), are considered in the interpolation estimation phase for $I^g_{mo}(i,j)$. The arrows in Figure 2

*Corresponding author. Supported by the National Science Council of R. O. C. under contracts NSC98–2221–E–011–102–MY3 and NSC98–2923–E–011–001.

(a) (b) (c)

Figure 2: Data dependence of our proposed interpolation estimation for $I^g_{mo}(i,j)$. (a) Horizontal variation (vertical edge). (b) Vertical variation (horizontal edge). (c) The other variations.

denote the relevant data dependence.

Further, in order to estimate $I^g_{dm}(i,j)$ more accurately, four proper weights in terms of the gradient magnitude are assigned to the corresponding four pixels in the interpolation estimation phase. Considering the neighboring pixel located at position $(i-1,j)$, if the vertical gradient magnitude is large, i.e. there is a horizontal edge passing through it, based on the color difference assumption [5], it reveals that the G component of this pixel makes less contribution to estimate that of the current pixel; otherwise, it reveals that the G component of this pixel makes more contribution to estimate that of the current pixel. According to the above analysis, the weight of the pixel at position $(i-1,j)$ is given by $w_g(V,i-1,j) = \frac{1}{1+\Delta I^V_{dm}(i-2,j)+2\Delta I^V_{dm}(i-1,j)+\Delta I^V_{dm}(i,j)}$. By the same argument, the weights of the other three neighbors are determined by $w_g(V,i+1,j) = \frac{1}{1+\sum_{k=0}^{2}\delta_k \Delta I^V_{dm}(i+k,j)}$, $w_g(H,i,j-1) = \frac{1}{1+\sum_{k=0}^{2}\delta_k \Delta I^H_{dm}(i,j-k)}$, $w_g(H,i,j+1) = \frac{1}{1+\sum_{k=0}^{2}\delta_k \Delta I^H_{dm}(i,j+k)}$, respectively, where $\delta_k = 2$ if $k=1$; $\delta_k = 1$, otherwise. Consequently, the value of $I^g_{dm}(i,j)$ can be estimated by

$$I^g_{dm}(i,j) = I^b_{mo}(i,j) + \frac{\sum_{(d,x,y)\in\xi_g} w_g(d,x,y) D_{gb}(x,y)}{\sum_{(d,x,y)\in\xi_g} w_g(d,x,y)}$$

$$\xi_g = \begin{cases} \xi_1 & \text{if } HP'_V(i,j) < \alpha HP'_H(i,j) \\ \xi_2 & \text{if } HP'_H(i,j) < \alpha HP'_V(i,j) \\ \xi_1 \cup \xi_2 & \text{otherwise} \end{cases}$$

where $\xi_1 = \{(V,i\pm1,j)\}$ and $\xi_2 = \{(H,i,j\pm1)\}$; for $(d_1,x_1,y_1) \in \xi_1$, $D_{gb}(x_1,y_1) = I^g_{mo}(x_1,y_1) - \frac{1}{2}\sum_{k\in\{\pm1\}} I^b_{mo}(x_1+k,y_1)$; for $(d_2,x_2,y_2)\in\xi_2$, $D_{gb}(x_2,y_2) = I^g_{mo}(x_2,y_2) - \frac{1}{2}\sum_{k\in\{\pm1\}} I^b_{mo}(x_1,y_1+k)$; the parameter α is set to $\alpha = 0.55$ empirically.

Finally, the refinement approach, which combines the concepts of the local color ratios [4] and the proper weighting scheme based on the gradient magnitude, is used to refine the fully populated G plane. For the current pixel at position (i,j), its G value $I^g_{dm}(i,j)$ can be refined by the following rule:

$$I^g_{dm}(i,j) = -\beta + (I^b_{mo}(i,j)+\beta)\frac{\sum_{(d,x,y)\in\xi'_g}\delta_{(d,x,y)} w_g(d,x,y) R_{gb}(x,y)}{\sum_{(d,x,y)\in\xi'_g}\delta_{(d,x,y)} w_g(d,x,y)}$$

where $\xi'_g = \{(H,i,j),(V,i,j),(H,i,j\pm1),(V,i\pm1,j)\}$; $R_{gb}(x,y) = \frac{I^g_{dm}(x,y)+\beta}{I^b_{mo}(x,y)+\beta}$; $\delta_{(d,x,y)} = \frac{1}{2}$ if $(d,x,y) \in \{(H,i,j),(V,i,j)\}$; $\delta_{(d,x,y)} = 1$, otherwise; the parameter β is set to $\beta = 256$ empirically.

After performing the above the edge-sensing interpolation estimation, the fully populated G plane can be constructed. In next subsection, the fully populated G plane will be used to assist the interpolation of the mosaic G-R and G-B color difference planes.

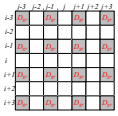


Figure 3: The pattern of the mosaic G-R color difference plane.

2.2 Stage 2: Interpolating the mosaic G-R and G-B color difference planes

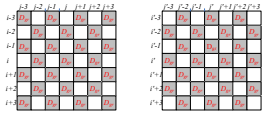
Instead of interpolating the R and B color planes directly, we interpolate the G-R and G-B color difference planes because the color difference plane is much smoother than the original color plane and it would alleviate the amplification of the estimation error in the later resizing stage. Since the interpolation for G-R color difference plane is the same as that for G-B color difference plane, in what follows, we only present it for G-R color difference plane. Based on the mosaic image as show in Figure 1 and the fully populated G plane, the mosaic G-R plane can be obtained by $D_{gr}(i_r,j_r) = I^g_{dm}(i_r,j_r) - I^r_{mo}(i_r,j_r)$. Figure 3 illustrates the pattern of the obtained mosaic G-R color difference plane for the position depicted in gray cells.

The G-R color difference plane interpolation estimation for the other positions consists of the following two steps: Step 1: interpolating the G-R color difference values of the pixels at positions $(i\pm2m, j\pm2n)$ in Fig. 6; Step 2: interpolating the G-R color difference values of the pixels at positions $(i\pm2m, j\pm2n+1)$ and $(i\pm2m+1, j\pm2n)$. For simplicity, the central pixel at position (i,j) in Fig. 3 is taken as the representative to explain the G-R color difference plane interpolation in Step 1. Similar to the interpolation for the mosaic G plane mentioned in last subsection, the G-R color difference value $D_{gr}(i,j)$ can be estimated from its four neighboring pixels with movement $\Omega_r = \{(x,y)|(x,y) = (i\pm1, j\pm1)\}$. In order to estimate $D_{gr}(i,j)$ more accurately, the gradient magnitudes of four diagonal variations are considered to determine the proper four weights. Given a pixel at position (x,y), its $\frac{-\pi}{4}$-diagonal and $\frac{\pi}{4}$-diagonal weights can be determine by $w_{gr}(\frac{-\pi}{4},x,y) = \frac{1}{1+\sum_{k=-1}^{1}\delta_k \Delta I^{\frac{-\pi}{4}}_{dm}(i+k,j+k)}$ and $w_{gr}(\frac{\pi}{4},x,y) = \frac{1}{1+\sum_{k=-1}^{1}\delta_k \Delta I^{\frac{\pi}{4}}_{dm}(i-k,j+k)}$, respectively, where $\delta_k = 2$ if $k=0$; $\delta_k = 1$, otherwise. Thus, the four weights of the four diagonal neighbors of the current pixel at position (i,j) are selected by $w_{gr}(\frac{-\pi}{4},i-1,j-1)$, $w_{gr}(\frac{\pi}{4},i-1,j+1)$, $w_{gr}(\frac{\pi}{4},i+1,j-1)$, and $w_{gr}(\frac{-\pi}{4},i+1,j+1)$, respectively. Based on the four weights, the G-R color difference value at position (i,j) can be estimated by

$$D_{gr}(i,j) = \frac{\sum_{(d,x,y)\in\xi_{gr}} w_{gr}(d,x,y) D_{gr}(x,y)}{\sum_{(d,x,y)\in\xi_{gr}} w_{gr}(d,x,y)}$$

where $\xi_{gr} = \{(\frac{-\pi}{4},i-1,j-1),(\frac{\pi}{4},i-1,j+1),(\frac{\pi}{4},i+1,j-1),(\frac{-\pi}{4},i+1,j+1)\}$.

After performing Step 1, the current pattern of the G-R color difference plane is illustrated in Fig. 4(a). For easy exposition, the central pixel at position (i',j') in Fig. 4(b), which is obtained by shifting Fig. 4(a) one pixel down, is taken as the representative to explain the G-R color difference plane interpolation in Step 2. Referring to Fig. 4(b), it is not hard to find that the pattern of the G-R color difference plane at present is the same as that of the G plane in the mosaic image as shown in Fig. 1. Therefore, the interpolation estimation approach described in last subsection can be directly used to estimate the G-R color difference value at position



(a) (b)

Figure 4: Two patterns of the G-R color difference plane. (a) The pattern of the G-R color difference plane after performing Step 1. (b) The pattern shifting (a) one pixel down.

(i', j'). Consequently, the G-R color difference value of the current pixel can be estimated by

$$D_{gr}(i', j') = \frac{\sum_{(d,x,y)\in\xi'_{gr}} w_{gr}(d, x, y) D_{gr}(x, y)}{\sum_{(d,x,y)\in\xi'_{gr}} w_{gr}(d, x, y)}$$

$$\xi'_{gr} = \begin{cases} \xi_1 & \text{if } HP'_V(i', j') < \alpha HP'_H(i', j') \\ \xi_2 & \text{if } HP'_H(i', j') < \alpha HP'_V(i', j') \\ \xi_1 \cup \xi_2 & \text{otherwise} \end{cases}$$

where $\xi_1 = \{(V, i' \pm 1, j')\}$ and $\xi_2 = \{(H, i', j' \pm 1)\}$; $w_{gr}(H, x, y) = \frac{1}{1+\sum_{k=-1}^{1} \delta_k \Delta I^H_{dm}(x,y+k)}$ and $w_{gr}(V, x, y) = \frac{1}{1+\sum_{k=-1}^{1} \delta_k \Delta I^V_{dm}(x+k,y)}$ where $\delta_k = 2$ if $k = 0$; $\delta_k = 1$, otherwise; the parameter α is set to $\alpha = 0.55$ empirically.

After constructing the fully populated G plane, G-R color difference plane, and G-B color difference plane, in next subsection, the three constructed planes will be resized to the arbitrary-ratio sized ones, and then the arbitrary-ratio resided full color image is followed.

2.3 Stage 3: Resizing the three related planes

Based on the composite length DCT [8], this subsection presents the arbitrary-ratio resizing stage for the fully populated G plane I^g_{dm}, G-R color difference plane D_{gr}, and G-B color difference plane D_{gb}. Since the resizing stage for I^g_{dm} is the same as that for D_{gr} and D_{gb}, we only present it for I^g_{dm}. Let $DCT(B^{(a)})$ and $IDCT(B^{(a)})$ be the DCT and inverse DCT on the block $a \times a$ $B^{(a)}$. For the fully populated G plane I^g_{dm} with size $M \times N$, we first divide it into a set of the image blocks, each with size 8×8, and the obtained image block set is denoted by $\Phi_{I^g_{dm}} = \{B^{(8)}_{m,n} | 0 \le m \le \frac{M}{8} - 1, 0 \le n \le \frac{N}{8} - 1\}$ where $B^{(8)}_{m,n}$ denotes the $(\frac{mN}{8} + n)$-th block in $\Phi_{I^g_{dm}}$. Then, the DCT is performed on each 8×8 image block $\{B^{(8)}_{m,n}$ to obtain the set of transformed blocks, $\hat{\Phi}_{I^g_{dm}} = \{\hat{B}^{(8)}_{m,n} | 0 \le m \le \frac{M}{8} - 1, 0 \le n \le \frac{N}{8} - 1\}$. If we want to resize the $M \times N$ G plane I^g_{dm} to the one with size $\frac{q}{p}M \times \frac{p}{q}N$, the resizing ratio is said to $\frac{p}{q}$ According to the resizing ratio $\frac{q}{p}$, first p^2 blocks in $\hat{\Phi}_{I^g_{dm}}$ are collected to be an active unit which are surrounded by dashed lines in Fig. 5. In order to achieve resizing ratio $\frac{q}{p}$, the p^2 blocks in each active unit should be increased or decreased to q^2 blocks.

For the active unit $\hat{\Psi}_{Act} = \{\hat{B}^{(8)}_{m,n} | 0 \le m, n \le p - 1\}$, the $\frac{q}{p}$-fold resizing procedure consists of the following four steps:

Step 1: Each 8×8 DCT coefficient block $\hat{B}^{(8)}_{m,n}$ in the active unit is expanded to a $(8 + z) \times (8 + z)$ block $\hat{B}'^{(8+z)}_{m,n}$ by the following zero padding rule:

$$\hat{B}'^{(8+z)}_{m,n}(x, y) = \begin{cases} \hat{B}^{(8)}_{m,n}(x, y) & \text{if } 0 \le x, y < 8 \\ 0 & \text{otherwise} \end{cases}$$
$$\forall x, y \in \{0, 1, \ldots, z - 1\};\ \forall m, n \in \{0, 1, \ldots, p - 1\}$$

where z denotes the smallest nonnegative integer satisfying the condition: $p(8 + z) = Cq$, $C \ge 8$. Consequently, we have the set of zero padded DCT coefficient blocks $\hat{\Psi}'_{Act} = \{\hat{B}'^{(8+z)}_{m,n} | 0 \le m, n \le p - 1\}$.

Step 2: For each $\hat{B}'^{(8+z)}_{m,n}$ in $\hat{\Psi}'_{Act}$, the $(8 + z) \times (8 + z)$ IDCT is performed on it to obtain the upsized image:

$$B'^{(8+z)}_{m,n} = IDCT(\hat{B}'^{(8+z)}_{m,n}),\ 0 \le m, n \le p - 1.$$

After performing the $(8 + z) \times (8 + z)$-sample IDCT's on all the zero padded DCT coefficient blocks in $\hat{\Psi}'_{Act}$, the upsized subimage I_{up} $(= \bigcup_{0\le m,n\le p-1} B'^{(8+z)}_{m,n})$ is constructed.

Step 3: Then, the upsized subimage I_{up} is divided into q^2 blocks, each block with size $\frac{p}{q}(8+z) \times \frac{p}{q}(8+z)$, and the set of the resampled image blocks is denoted by $\Psi_{Re} = \{R^{(\frac{p}{q}(8+z))}_{m,n} | 0 \le m, n \le q - 1\}$. Next, the DCT is performed on each $R^{(\frac{p}{q}(8+z))}_{m,n}$ in Ψ_{Re} to construct the resampled DCT coefficient block $\hat{R}^{(\frac{p}{q}(8+z))}_{m,n}$

$$\hat{R}^{(\frac{p}{q}(8+z))}_{m,n} = DCT(R^{\frac{p}{q}(8+z)}_{m,n}),\ 0 \le m, n \le q - 1.$$

We thus have the set of resampled DCT coefficient blocks $\hat{\Psi}_{Re} = \{\hat{R}^{(\frac{p}{q}(8+z))}_{m,n} | 0 \le m, n \le q - 1\}$.

Step 4: Finally, the high-frequency DCT coefficients of each block in $\hat{\Phi}_{Re}$ are truncated by the following rule:

$$\hat{R}'^{(8)}_{m,n}(x, y) = \hat{R}^{(\frac{p}{q}(8+z))}_{m,n}(x, y)$$
$$\forall x, y \in \{0, 1, \ldots, 8\};\ \forall m, n \in \{0, 1, \ldots, q - 1\}$$

where $\hat{R}'^{(8)}_{m,n}$ denotes the left-upper 8×8 subblock of $\hat{R}^{(\frac{p}{q}(8+z))}_{m,n}$; $\hat{R}^{(\frac{p}{q}(8+z))}_{m,n}(x, y)$ and $\hat{R}'^{(8)}_{m,n}(x, y)$ denote the DCT coefficients of the pixels at position (x, y) in $\hat{R}^{(\frac{p}{q}(8+z))}_{m,n}$ and $\hat{R}'^{(8)}_{m,n}$, respectively. Consequently, the resized active unit $\hat{\Psi}'_{Re} = \{\hat{R}'^{(8)}_{m,n} | 0 \le m, n \le q - 1\}$ is constructed.

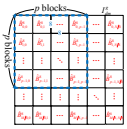


Figure 5: An example of the active unit in $\hat{\Phi}_{I^g_{dm}}$.

After performing the above resizing procedure on all the active units in I^g_{dm}, the new set of the DCT coefficient blocks $\hat{\Phi}_{Z^g_{dm}} = \{\hat{B}^{(8)}_{m,n} | 0 \le m \le \frac{\frac{q}{p}M}{8} - 1, 0 \le n \le \frac{\frac{q}{p}N}{8} - 1\}$ can be obtained. Consequently, the $\frac{q}{p}M \times \frac{q}{p}N$ sized G plane Z^g_{dm} can be obtained by performing the 8×8-sample IDCT on all the DCT coefficient

Figure 6: The twenty-four testing images from Kodak PhotoCD.

Table 1: Average CPSNR comparison for the three concerned algorithms.

	CPSNR			ΔE^*_{ab}		
	$\frac{q}{p}$			$\frac{q}{p}$		
	2	$\frac{4}{3}$	Average	2	$\frac{4}{3}$	Average
A_1	28.41	32.16	30.29	3.529	2.624	3.077
A_2	28.61	32.57	30.34	3.416	2.503	2.960
Ours	**29.12**	**33.58**	**31.35**	**3.194**	**2.342**	**2.718**

blocks in $\hat{\Phi}_{Z^g_{dm}}$. By the same argument, the resized G-R and G-B color difference planes, ZD_{gr} and ZD_{gb}, can be also obtained. Finally, the resized R and B planes can be constructed by

$$Z^r_{dm}(i_z, j_z) = Z^g_{dm}(i_z, j_z) - ZD_{gr}(i_z, j_z)$$
$$Z^b_{dm}(i_z, j_z) = Z^g_{dm}(i_z, j_z) - ZD_{gb}(i_z, j_z)$$
$$\forall i_z \in \{0, 1, \ldots, \frac{q}{p}M - 1\}, \forall j_z \in \{0, 1, \ldots, \frac{q}{p}N - 1\}\}$$

where $Z^r_{dm}(i_z, j_z)$, $Z^g_{dm}(i_z, j_z)$, and $Z^b_{dm}(i_z, j_z)$ denote the three color components of the pixel at position (i_z, j_z) in the resized full color image Z_{dm}, respectively; $ZD_{gr}(i_z, j_z)$ and $ZD_{gb}(i_z, j_z)$ denote the G-R and G-B color difference value of the pixel at position (i_z, j_z) in ZD_{gr} and ZD_{gb}, respectively.

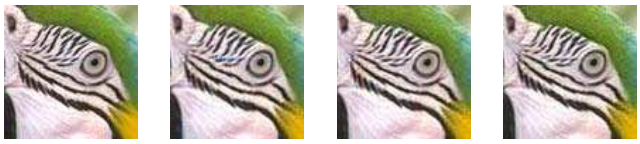
3. EXPERIMENTAL RESULTS

In this section, based on twenty-four popular testing mosaic images, some experimental results are demonstrated to show the applicability and quality advantages of the proposed algorithm. Fig. 6 illustrates the twenty-four testing images from Kodak PhotoCD. In our experiments, the twenty-four testing images, each with size 512×728, are first downsized to the $\frac{p}{q} \cdot 512 \times \frac{p}{q} \cdot 728$ sized ones, and then the downsized images are down-sampled to the mosaic images.

In the two native resizing algorithms to be compared with our proposed algorithm, one of the two demosaicing methods proposed in [5] and [3], respectively, is first utilized to obtain the demosaiced images and then the resizing method proposed in [8] is applied to construct the resized full color images. For convenience, the three native algorithms based on the demosaicing methods proposed in [5] and [3] are called A_1 and A_2, respectively.

Here, we adopt two objective color image quality measures, the CPSNR and the S-CIELAB ΔE^*_{ab} [4], and one subjective color image quality measure, the color artifacts, to justify the the quality advantage of of the proposed algorithm. Based on the twenty-four testing images, among the four algorithms, Table 1 demonstrates the image quality comparison in terms of the average CPSNR and the average S-CIELAB ΔE^*_{ab}, respectively. From Table 1, it is observed that the proposed resizing algorithm produces the best image quality in terms of CPSNR and S-CIELAB ΔE^*_{ab}.

Next, the subjective image visual measure, color artifacts, is adopted to demonstrate the visual quality advantage of the proposed resizing algorithm. For saving space, we just show the visual comparison when the resizing ratio $\frac{q}{p}$ is $\frac{4}{3}$. Four magnified subimages cut from the testing image No. 23 are adopted to examine the visual effect among the three concerned algorithms. Figures 7 (a)–(d) illustrate magnified subimages cut from the resized images obtained by original image and the three concerned resized algorithms, respectively. From the visual comparison, it is observed that based on the same resizing ratio, the proposed algorithm produces less color artifacts when compared with the other two algorithms.



(a) (b) (c) (d)

Figure 7: For image No. 23, when the resizing ratio $\frac{q}{p}$ is $\frac{4}{3}$, four magnified subimages cut from the resized images obtained by (a) original image, (b) A_1, (c) A_2, and (d) the proposed algorithm.

4. CONCLUSIONS

In this paper, a new joint demosaicing and arbitrary-ratio resizing algorithm for mosaic images has been presented. Based on the color difference concept and the composite length DCT, the mosaic image can be demosaiced and resized to the arbitrary-ratio sized full color image. Based on twenty-four popular testing mosaic images, the proposed new resizing algorithm has better image quality performance when compared with three native algorithms.

5. REFERENCES

[1] B. E. Bayer, "Color imaging array," *U.S. Patent# 3 971 065*, 1976.

[2] H. A. Chang and H. H. Chen, "Stochastic color interpolation for digital cameras," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 17, no. 8, pp. 964–973, 2007.

[3] K. H. Chung and Y. H. Chan, "Color demosaicking using variance of color differences," *IEEE Trans. Image Processing*, vol. 15, no. 10, pp. 2944–2955, 2006.

[4] K. L. Chung W. J. Yang W. M. Yan and C. C. Wang, "Demosaicing of color filter array captured images using gradient edge detection masks and adaptive heterogeneity-projection," *IEEE Trans. Image Processing*, vol. 17, no. 12, pp. 2356–2367, 2008.

[5] S. C. Pei and I. K. Tam, "Effective color interpolation in ccd color filter arrays using signal correlation," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no. 6, pp. 503–513, 2003.

[6] R. Lukac K. N. Plataniotis and D. Hatzinakos, "Color image zooming on the bayer pattern," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 15, no. 11, pp. 1475–1492, 2005.

[7] L. Zhang and D. Zhang, "A joint demosaicking-zooming scheme for single chip digital color cameras," *Computer Vision and Image Understanding*, vol. 107, no. 1-2, pp. 14–25, 2009.

[8] Y. S. Park and H. W. Park, "Arbitrary-ratio image resizing using fast dct of composite length for dct-based transcoder," *IEEE Trans. Image Processing*, vol. 15, no. 2, pp. 494–500, 2006.

Combined linear resampling method with ringing control

Andrey S. Krylov, Andrey V. Nasonov, Alexandra A. Chernomorets
Laboratory of Mathematical Methods of Image Processing, Faculty of Computational Mathematics and Cybernetics,
Lomonosov Moscow State University,
{kryl, nasonov}@cs.msu.ru

Abstract

New method to combine different linear interpolation algorithms is suggested. It uses total variation analysis to suppress ringing artifact of the combination. This method enables to construct fast edge adaptive resampling methods. Its usage is illustrated with combinations of sinc, Papoulis and bicubic interpolation algorithms using new image metrics for interpolation methods quality analysis. The method can be also used to combine non-linear methods.

Keywords: *Image interpolation, ringing artifact, deringing, total variation, image metrics.*

1. INTRODUCTION

Every linear resampling method has its own trade-off between three types of artifacts: ringing, aliasing and blur, illustrated in Fig.1. For example, 'ideal' interpolation method which reconstructs the image by a sampling theorem processes edges good, but introduces strong ringing effect. On the other hand, bilinear interpolation adds blur and aliasing to the edges, but does not add ringing artifact. The detailed overview of linear image interpolation methods can be found in [1].

Blur Aliasing Ringing

Figure 1: Typical artifacts of linear interpolation methods.

One of the ways to improve the quality of linear interpolation methods is to perform additional postprocessing to suppress the artifacts. Lots of general purpose deblur and deringing algorithms were developed. Most of the deblurring algorithms are based on deconvolution problem with regularization [2, 3]. In [4], image is decomposed into cartoon-like component and texture component and different methods are used to process these components. Regularization approach is also used for image deringing [5].

Another way is to embed artifact suppression method in the interpolation algorithm. It can be done by a method that constructs high-resolution images using a combination of two interpolation methods, where one of the methods works well in the edge area while the other method shows better results in the rest of the image. At the same time the edge detection procedure to detect edge areas is time consuming and often suffers from noise. In our approach, we use total variation (TV) analysis instead of edge detection.

TV value is closely related with ringing effect [6]. If TV increases after interpolation, we assume that the method causes ringing effect.

2. INTERPOLATION

We consider only the case of one-dimensional interpolation. For two-dimensional images, we perform interpolation first by rows, then by columns.

Generally, linear interpolation of a function $F(x)$ given on a discrete set $\{x_i = ih\}$ looks as:

$$f(x) = \sum_{i=-\infty}^{+\infty} F(ih)K(i - x/h), \tag{1}$$

where $K(x)$ is an interpolation kernel.

In this work the following interpolation methods are used:

1. Sinc (or 'ideal') interpolation. If the input function satisfies the condition of the Shannon-Kotelnikov sampling theorem, then it can be reconstructed using (1) with

$$K(x) = \mathrm{sinc}x = \frac{\sin \pi x}{\pi x}.$$

2. Bicubic interpolation:

$$K(x) = \begin{cases} 1, & \text{for } x = 0, \\ (a+2)|x|^3 - (a+3)|x|^2 + 1, & \text{for } 0 < |x| < 1, \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a, & \text{for } 1 < |x| < 2, \\ 0, & \text{otherwise.} \end{cases}$$

We use bicubic interpolation with $a = -0.5$.

3. Interpolation using Papoulis sampling theorem [7, 8]. If the original signal is represented as a series of its samples and its first and second derivatives (we call it Papoulis-2), it can be reconstructed as

$$f(x) = 3\sin^3(\pi x) \sum_{i=-\infty}^{+\infty} \Big[F(i)\frac{2 + \pi^2(x-i)^2}{6\pi^3(x-i)^3} + \\ + F'(i)\frac{1}{3\pi^3(x-i)^2} + F''(i)\frac{1}{6\pi^3(x-i)}\Big]. \tag{2}$$

We do not know the derivatives of $F(x)$ in sampling points, so we use the following approximation:

$$F'(i) = \frac{F(i+1) - F(i-1)}{2},$$
$$F''(i) = \frac{F(i+2) - 2F(i) + F(i-2)}{4}.$$

Thus, the formula (2) can be interpreted as linear interpolation method (1) with

$$K(x) = 3\sin^3(\pi x)\Big[\frac{1}{3\pi^3}\Big(\frac{1}{x^3} + \frac{2x}{(x^2-1)^2} + \\ \frac{1}{x(x^2-4)}\Big) + \frac{1}{6\pi x}\Big], \tag{3}$$

$K(0) = 1, K(1) = K(-1) = K(2) = K(-2) = 0.$

The kernels of these methods are illustrated in Fig.2.

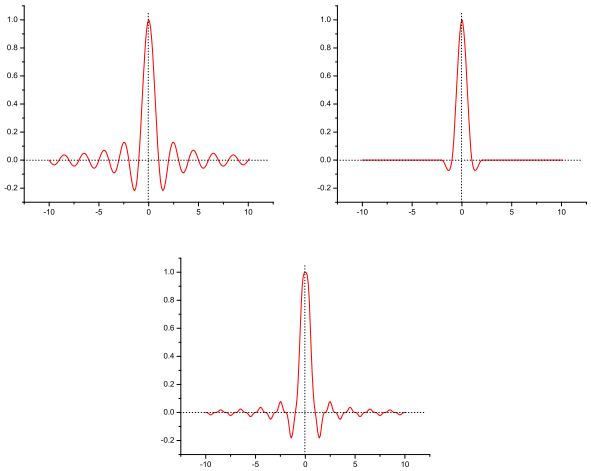


Figure 2: Interpolation kernels for sinc interpolation (left graph), bicubic interpolation (right graph) and Papoulis-2 interpolation (bottom graph).

3. COMBINED METHOD

We construct the interpolated function $f(x)$ as a combination of two functions

$$f(x) = \alpha(x)s(x) + (1 - \alpha(x))r(x), \tag{4}$$

where $r(x)$ is the interpolation of $F(x)$ using the method which produces ringing effect but processes edges well, and $s(x)$ is the result of interpolation method which produces smooth image with smaller ringing effect.

3.1 Total Variation Approach

The coefficient $\alpha(x)$ in (4) is calculated using the analysis of total variation (TV) of $F(x)$ and $r(x)$. In one-dimensional case, TV functional is defined as

$$TV(f, a, b) = \int_a^b |f'(x)|dx.$$

In the discrete case $\{x_i = ih\}$, it is computed as:

$$\begin{aligned} TV(f, a, b) = \sum_{i=i_a}^{i_b-1} |f((i+1)h) - f(ih)| + \\ +(i_a h - a)|f(i_a h) - f((i_a - 1)h)| + \\ +(b - i_b h)|f((i_b + 1)h) - f(i_b h)|, \end{aligned} \tag{5}$$

where $i_a = \lceil \frac{a}{h} \rceil$ (the less integer value greater than or equal to a/h), $i_b = \lfloor \frac{b}{h} \rfloor$ (the greater integer value less than or equal to b/h).

We calculate the ratio between TV of the input image row (column) and TV of its interpolation

$$k(x) = \frac{TV(r, x - nh, x + nh)}{TV(F, x - nh, x + nh)}, \tag{6}$$

where the parameter n defines the half-size of the window of TV calculation. Experiments have shown that good results are obtained with $n = 3$.

If we consider the problem of interpolation of $F(x)$ on a discrete grid with interpolation factor q, we do not need to construct the continuous function $r(x)$, and the formula (6) takes the form

$$k(x) = \frac{TV(R, x - nh, x + nh)}{TV(F, x - nh, x + nh)}, \tag{7}$$

where $R(x)$ is $r(x)$ defined on the grid $\{x_i = ih/q\}$.

We use the following criteria for $\alpha(x)$ computation:

1. If $k(x) \geq B$, we assume that the ringing effect is noticeable, and we take $\alpha(x) = 1$.

2. If $k(x) \leq A$, we assume that the ringing effect is unnoticeable, and we use $\alpha(x) = 0$.

3. if $A < k(x) < B$, we use $\alpha(x) = \frac{k(x)-A}{B-A}$.

Parameters A and B are the thresholds.

3.2 Threshold Analysis

We estimate the parameters A and B experimentally using minimization the 2D image quality metrics from [9]. BEP (Basic Edges Points RMSE) value measures the root of the average square error in edge points area. BEN (Basic Edges Neighborhood RMSE) value calculates the error in edge neighborhood where ringing effect usually appears. As an example, sinc interpolation shows good BEP and bad BEN while bicubic interpolation results in good BEN and bad BEP. To find a balance between these two metrics, we use BEQ metrics:

$$BEQ = \log_2 \frac{BEP}{BEP_*} + \log_2 \frac{BEN}{BEN_*},$$

where BEP_* and BEN_* are the normalization constants. These constants do not affect the difference between BEQ values of different images. We choose BEP_* and BEN_* as the minimal values of BEP and BEN respectively of the results of interpolation methods for the given image.

The proposed combined method was analyzed with different pairs of interpolation methods. The analysis consisted in choice of threshold values A and B by minimization of the BEQ value.

Experiments show that the parameters A and B depend on the used pair of interpolation methods. We have calculated threshold values A and B for combinations of sinc, Papoulis-2 and bicubic interpolation methods on a set of standard test images Lena, Barbara, Peppers, Boat, House. The results are presented in table.1.

Interpolation Methods	threshold A	threshold B
Sinc + Bicubic	1.10	1.31
Sinc + Papoulis-2	1.09	1.19
Papolis-2 + Bicubic	1.20	1.32

Table 1: Optimal threshold values A and B for different combinations of interpolation methods.

3.3 Method speed-up

The performance of TV calculation of the proposed method can be improved using the additive property of TV

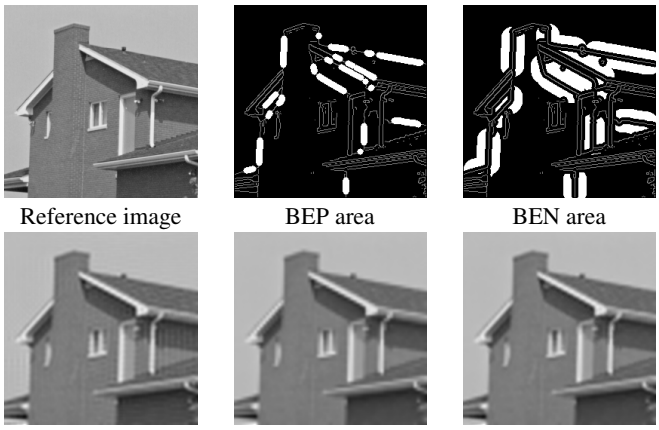
$$TV(f, a, b) + TV(f, b, c) = TV(f, a, c)$$

for $a \leq b \leq c$. This means, that if we shift the window of TV calculation by x, we need just to update the previously calculated TV value

$$TV(f, a+x, b+x) = TV(f, a, b)+TV(f, b, b+x)-TV(f, a, a+x).$$

4. RESULTS

The proposed method is illustrated with 'house' image from the test image set in Fig. 3. The values of MSE, BEP, BEN and BEQ metrics for these images are shown in Tab. 2. It can be seen that the combined method keeps both BEP and BEN metrics low. Despite of slightly increased MSE value, the results of the combined method looks better than of pure sinc or bicubic methods. BEQ metrics correlates with the perceptual image quality.



Reference image | BEP area | BEN area

Sinc interpolation | Bicubic interpolation | Combined method

Figure 3: Illustration of the proposed method by the test image with sinc and bicubic methods. Grey lines in BEP and BEN areas images are the edges.

	MSE	BEP	BEN	BEQ
Sinc interpolation	109.5	14.124	4.426	0.181
Papoulis interpolation	112.1	14.247	4.032	0.059
Bicubic interpolation	125.4	15.178	3.971	0.128
Sinc + Bicubic	112.1	14.181	3.970	0.029
Sinc + Papoulis	109.4	13.894	4.039	0.025
Papoulis + Bicubic	113.3	14.302	3.981	0.046

Table 2: The values of metrics for the images from Fig. 3.

5. CONCLUSION

It was shown that total variation analysis concept enables to combine different linear interpolation algorithms to suppress ringing artifact. New image metrics for interpolation methods quality analysis were used to numerically evaluate the results of the proposed method. Experiments show that the combined method parameter selection using these metrics correspond well to visual image quality enhancement. Fast and effective implementation of the proposed method which is useful for embedded solutions has been suggested. The algorithm can be extended using the combination of more than two methods which are not necessary linear.

The work was supported by federal target program "Scientific and scientific-pedagogical personnel of innovative Russia in 2009-2013" and RFBR grant 09-01-92470-MHKC.

6. REFERENCES

[1] P. Thevenaz, T. Blu, and M. Unser, "Interpolation revisited," *IEEE Trans. Medical Imaging*, vol. 19, no. 7, pp. 739–758, 2000.

[2] Hiroyuki Takeda, Sina Farsiu, and Peyman Milanfar, "Deblurring using regularized locally adaptive kernel regression," *IEEE Trans. Image Proc.*, vol. 17, no. 4, pp. 550–563, 2008.

[3] Li Chen, , and Kim-Hui Yap, "A soft double regularization approach to parametric blind image deconvolution," *IEEE Trans. Image Proc.*, vol. 14, no. 5, pp. 624–633, 2005.

[4] Takahiro Saito, Yuki Ishii, Haruya Aizawa, and Takashi Komatsu, "High-quality image interpolation via nonlinear image decomposition," *Proceedings of the SPIE*, vol. 6812, pp. 68120V–68120V–12, 2008.

[5] V. B. S. Prasath and A. Singh, "Ringing artifact reduction in blind image deblurring and denoising problems by regularization methods," *International Conference on Advances in Pattern Recognition*, pp. 333–336, 2009.

[6] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1999.

[7] F. Deschenes A. Hore, D. Ziou, "A new image scaling algorithm based on the sampling theorem of papoulis and application to color images," *ICIG 2007*, pp. 39–44, 2007.

[8] A. Papoulis, "Generalized sampling theorem," *IEEE Trans. on Circuits and Systems*, vol. 24, pp. 652–654, 1977.

[9] A. V. Nasonov and A. S. Krylov, "Adaptive image deringing," *Proceeding of Graphicon'2009*, 2009.

About authors

Andrey S. Krylov is associated professor, Head of the Laboratory of Mathematical Methods of Image Processing, CMC MSU.
Email: kryl@cs.msu.ru

Andrey V. Nasonov is member of scientific staff of CMC MSU.
Email: nasonov@cs.msu.ru

Alexandra A. Chernomorets is a student of CMC MSU.

3D scenes simulation, animation, and synchronization in training systems with force back-coupling

Aleshin V, Klimenko S, Astskhov J, Bobkov A, Borodina M, Volegov D, Kazansky I, Novgorodtsev D, Frolov P

Department of General and Applied Physics

Moscow Institute of Physics and Technology, Moscow, Russia
e-mail: aleshin_vl@mail.ru, nov-dmitrij@yandex.ru

Abstract

In this project, we are trying to create a training system with force back-coupling and realistic visual entourage imitation. All existing training systems either have no visual environment (e.g. high-g training centrifuges for astronauts) or do not provide sufficient dynamic loads (e.g. flight and driving training systems) [1].
We designed the method of 3D environment creating and rendering and synchronized the resulting 3D scenes with a training system with force back-coupling. For our investigations, we chose the alpine skiing training system produced by the SkyTec Interactive company as this training system involves physical activity with complex coordination. For 3D scenes rendering, we used the OpenSceneGraph high performance open source 3D graphics toolkit. Our method provides deeper user immersion into the virtual environment and as a result more efficient training and corresponding skills mastering. The project was supported the Russian Foundation of Basic Researches grant 08-07-00399

Keywords: *Training system, 3D visual environment, simulation, animation, synchronization*

1. INTRODUCTION

Training systems with force back-coupling are often used by astronauts, sportsmen, in military training, and for entertainment. [2] Such training systems imitate real power loads and allow improving necessary skills. However, most of them either have poor visualization or have no visualization at all.
The virtual environment technology allows to experience deep immersion in the visual scene and train more efficiently.
In this project, we used the active alpine skiing training system produced by the SkyTec Interactive Russian company [3, 4]. This training system imitates a wide range of real physical loads that a sportsman experiences while skiing. The system simulates centripetal acceleration loads that the sportsman experiences while performing a turn. At the same moment (while performing a turn), the sportsman shifts sideward alone the training system platform the same as it happens on a real slope. The total range of sideward shifting is 5.5 m. This training system exceeds all its analogues in its characteristics.
For 3D scenes rendering, we used the OpenSceneGraph high performance open source 3D graphics toolkit. OpenSceneGraph uses the OpenGL programming interface. For 3D objects controlling, OpenSceneGraph uses a full-featured and industry-proven scene graph. Using the scene graph feature provides the 3D objects rendering and animation adaptive management.

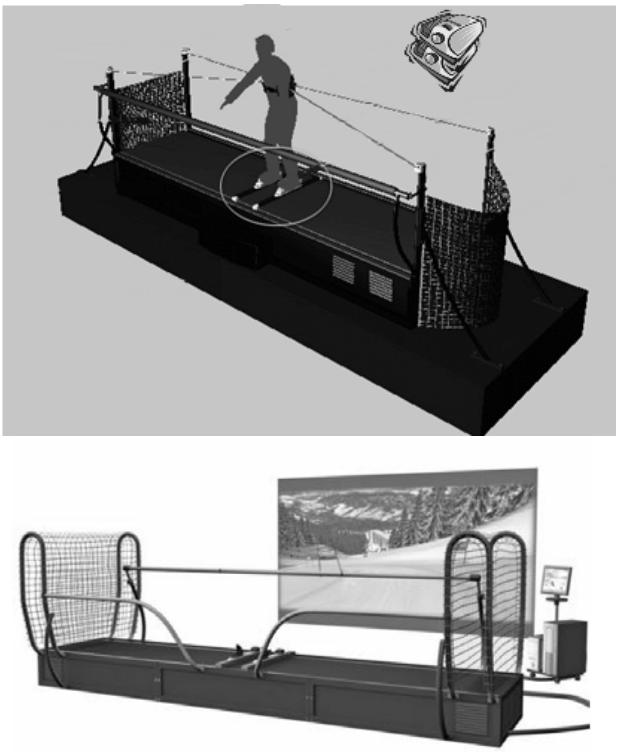


Figure 1: The training system scheme.

The designed system consists of two main parts: the 3D visualization subsystem and the sportsman motion modeling subsystem.
The 3D visualization subsystem creates a stereoimage of a skiing track, surrounding landscape, obstacles, and virtual sportsman (avatar).
The sportsman motion modeling subsystem provides interaction between the sportsman and the training system and computes physical aspects of the process: it receives data about the sportsman position on the training platform, computes his position on the skiing track, computes power loads on the sportsman, and sends this data back to the training system. This subsystem also receives signals from the head tracking system and computes the sportsman head position. After that, it sends the new position and orientation of the camera to the visualization subsystem.
Besides these two main subsystems, there is also the controlling subsystem. This subsystem is connected to the operator display. Through this subsystem, the operator can

choose the skiing track and weather conditions, adjust motion parameters, control the strength of power loads, etc. The operator can observe different statistics: the sportsman velocity, his position on the track, the time elapsed from the start, etc.

2. 3D SCENE DESCRIPTION

A 3D skiing track scene consists of the territory surface, the skiing track attributes (flags, nets, ski lift), other objects (trees, houses, spectators), and the landscape in the background. On the base of this 3D scene, the visualization subsystem creates a stereoimage that the sportsman sees when training. The sportsman motion modeling subsystem also uses the 3D scene for computing the sportsman position and orientation on the track and detecting his collisions with virtual obstacles (flags, a net, trees).

We designed the 3D scene description format that can be used by both main subsystems. The terrain surface is described in one file, and the skiing track and 3D objects position in another separate file.

Our systems supports landscapes and surfaces created and described in different ways and applications.

The first way is to create the surface on the base of a regular grid of vertexes. Such surface can be created in 3D editors: 3D Max, Blender, etc. In these 3D editors, we can change the surface by changing positions of separate vertexes and observe the result at once. In this way, we can create surfaces with the most complicated geometry. But the performance suffers greatly because the motion modeling subsystem has to work with the whole set of vertexes.

The second way is to create the surface on the base of an altitude map on the fly. An altitude map is a map image where pixel brightness stands for the corresponding point altitude. The visualization subsystem creates a regular grid of vertexes where one vertex stands for one pixel in the altitude map. The subsystem defines the altitude of a pixel from its brightness and transfers this altitude value to the *Z* coordinate of the corresponding vertex in the grid. In most cases, it is easier to keep one large regular grid and work with it. However, in the case of large landscapes, it is preferably to keep a small grid connected to the camera position and dynamically update the vertexes *Z* coordinates on the camera motion. The altitude map also simplifies the process of computing the sportsman position on the track in the motion modeling subsystem [5].

An altitude map can be created manually in either graphical editor. However, there are dedicated graphical applications such as Terragen or Bryce. Altitude maps of real landscapes made from satellites can be also used.

Instead of processing altitude maps manually, we can use VirtualPlanetBuilder, a special library produced by the OpenSceneGraph developers. VirtualPlanetBuilder transforms the received altitude map into the 3D model with different detail levels. The resulting 3D model appears to be a graph that consists of different parts of the scene with different detail levels. VirtualPlanetBuilder saves this model in the native OpenSceneGraph format. During the training system operation, OpenSceneGraph automatically renders parts of the scene with the corresponding detail levels. This method leverages performance greatly. The motion modeling subsystem uses the IntersectionVisitor class built in OpenSceneGraph. Due to this class, the subsystem receives the altitude value of each point. Using the VirtualPlanetBuilder library is reasonable in the case of very large landscapes.

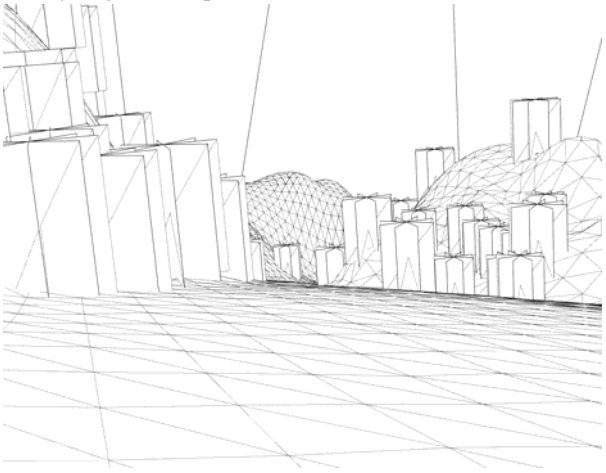


Figure 2: A polygon model.

Figure 3: A 3D mountain landscape.

After the surface is created, we should create and describe a skiing track and different 3D objects position. We can use one surface for many different skiing tracks.

The skiing track description is saved as a list of checkpoints in a separate file. Each checkpoint (x,y) is written on a new line. The skiing track and 3D objects preferences are saved in another file.

The system reads these files before loading the scene. Using checkpoints position and the track width, the system calculates the track borders and flags positions. Using the 3D surface description, the system computes the *Z* coordinate of all its points. Then the system puts the 3D models of flags and nets designed in either 3D editor to the calculated coordinates. The system also adds trees, houses, and other 3D objects to the scene. These objects position can be set explicitly or using a list of rules.

For example, for setting trees, we can use two rules. First, the density of trees depends on the area altitude. The higher the area, the less trees should be placed. Second, the trees density cab be defined using a map by analogy with the altitude map. In this “tree” map, the pixel brightness stands for the trees density in this point.

Figure 4: A skiing track with textures and obstacles.

Besides static 3D scenes, we can use dynamically generated endless scenes. In such endless scene, the surface is generated on the flight during the sportsman descent. The surface consists of a sequence of elements. Each element is created from a set pattern by applying noises to it.

3. TRAINING SYSTEM SYNHRONIZATION

The training system is connected to the computer through the special input/output cardboard. The computer receives the training system events signals from the cardboard in a separate flow. The events on the training system appear when the training system carriage changes its position or the inclination angle changes. Using the inclination angle, the current sportsman position on the track, the surface description, and some other parameters, the computer calculates the next sportsman position and the power loads on him. Then the computer sends the power loads values back to the I/O cardboard [6-8].

Before beginning to render the next frame, the visualization subsystem receives the new camera position. Thus, the sportsman experiences the training system force back-coupling and observes his sliding alone the skiing track.

When computing the next sportsman position, the motion modeling subsystem checks for collisions with other objects of the scene: flags, nets, and borders. Depending on the strength and type of a collision, the motion modeling system can either stop the training or send a message to the visualization subsystem to change the crashed obstacle position. In the later case, the visualization subsystem will start the corresponding animation in the next frame (e.g. a flag will begin to bend).

Figure 5: The experimental setup.

Figure 6: The training process.

The lateral shift and the edge angle are measured, the obtained data is digitized and transmitted to the virtual environment system [1,3,5]. A tracking system can be used to detect the location of the trainee's eyes. The virtual environment system models the skier's trajectory and determines the skier's position in the 3D virtual environment scene on the basis of the data obtained. The main synchronization problem is taking into account physical laws of a skier/snowboarder turning movements. The good analysis of this problem is presented in P. A. Federolf [6]. For the ideal carving turn, it is possible to use the approximation of D. Lind, S. P. Sanders [8]:

$$\tan\phi = \frac{v^2}{gR\cos\alpha} \mp \tan\alpha\cos\beta$$

where ϕ - is a inclination angle, (a tilt angle), v -is a centre mass velocity of a skier, R - is a turn radius, α - is an angle of the slope, β - is a ski traversing angle at a fixed turn point, g - is a gravitational acceleration. A turn radius R at the first approximation will be determined by the edging angle and the interaction of a ski and snow [8]. An inclination angle ϕ differs from edging angle because of the angulation. The basic problem of synchronization is connected with taking into account all foregoing parameters and with definition of the skier position in an induced virtual scene. An individual problem is formation of a signal to a training system according to the position of the skier in an induced virtual scene.

4. CONCLUSION AND RESULTS

In this article, we provided the description of the system that simulates and visualizes the sportsman sliding down a skiing track. The system based on the SkyTec Interactive training system for skiers and snowboarders. This system allows to create different 3D surfaces, scenes, objects, to set skiing tracks, and to stuff them with obstacles. Using the training system events tracking, our system computes the sportsman position on the skiing track and generates power loads on him.
Nowadays, we perform tests on our system with artificial mountain landscapes and skiing tracks. The next stage of our work is to create 3D models of real skiing resorts using stereoscopic topographical survey. Results of work were presented in Russian multimedia [9, 10].

4. REFERENCES

[1] *Aleshin V., Afanasiev V, Klimenko S., et al*. (2004) System for Visualization of Induced Virtual Environment, Proc. 14th International Conference GraphiCon-2004, Moscow, Russia, pp.12-15

[2] *Stanislav Klimenko*, Virtual Planetarium: Educational Application in Virtual Environment, talk at ED-MEDIA 2001 (World Conference on Educational Multimedia, Hypermedia & Telecommunication, June 25-30, 2001, Tampere, Finland)

[3] *Vladimir Aleshin, Stanislav Klimenko, Mikhail Manuilov, Leonid Melnikov,* Alpine Skiing And Snowboarding Training System Using Induced Virtual Environment // Proc. .of the 4th International Congress on Science and Skiing, ICSS 2007, St. Christoph, Austria, publ. 2009, ISBN:978-1-84126-255-0, pp. 137-144, Meyer&Meyer Sport.

[4] Trainer SkyTec Interactive: www.skytec-interactiv.com

[5] *Aleshin V, Klimenko S, Astakhov J, Bobkov A, Volegov D, Borodina M, Kazansky I, Novgorodcev D, Frolov P.* System for Visualization of Induced Virtual Environment fot the Skiing Training System //Proc. Of the Internal Congress "Dialog 2008", ISBN 978-5-88835-020-1, –M., 2008.

[6] *P. A. Federolf*, Finite Element Simulation Of a Carving Snow Ski, PHD Thesis, 2005

[7] *Aleshin V., Danilin V.* (2002). Basics of Alpine Skiing technique, chapter of the book "Alpine Skiing and Snowboarding", pp.35-46

[8] *David Lind, Scott P. Sanders*, The Physics of Skiing, 2004, Springer.

[9] Virtual sport, newspaper "Arguments and facts", № 27, 01.07.2009, p.48

[10] *Andrey Negru, Alpine skiing without leaving home, 10.07.2009, Utro-Russia,* www.utro-russia.ru/news.html?id=44786

About the author

Vladimir Aleshin is a docent at Moscow Institute of Physics and Technology, Department of General and Applied Physics. His contact email is aleshin_vl@mail.ru.

Stanislav Klimenko is a professor at Moscow Institute of Physics and Technology, Department of General and Applied Physics. His contact email is stanislav.klimenko@gmail.com.

Astskhov Juriy is a post-graduate student at Moscow Institute of Physics and Technology, Department of General and Applied Physics.

Bobkov Alexander is a student at Moscow Institute of Physics and Technology, Department of General and Applied Physics. His contact email is archimag@dgap.mipt.ru

Borodina Maria is a student at Moscow Institute of Physics and Technology, Department of General and Applied Physics. Her contact email is mari.borodina@gmail.com

Volegov Dmitriy is a post-graduate student at Moscow Institute of Physics and Technology, Department of General and Applied Physics.

Kazansky Ilia is a post-graduate student at Moscow Institute of Physics and Technology, Department of General and Applied Physics. His contact email is railway_station@mail.ru.

Novgorodcev Dmitry is a post-graduate student at Moscow Institute of Physics and Technology, Department of General and Applied Physics. His contact email is nov-dmitrij@yandex.ru.

Frolov Pavel is a post-graduate student at Moscow Institute of Physics and Technology, Department of General and Applied Physics.

Applying one approach to geometric modelling

Vasyl Tereshchenko
Faculty of Cybernetics
National Taras Shevchenko University , Kiev, Ukraine
v_ter@ukr.net

Abstract

The current state of computer technology allows formulate of new problems and develop effective solutions for them. This requires developing of complex computer models and ways of their formation.Thus, when designing and building a visual model of thermal processes at welding turbine blades of aircraft engines, must be extremely accurate model the welding bath, the welding seam and region adjacent to them. Getting the exact solution of this problem requires simultaneous solution the whole complex of heat physical and geometrical problems which demand significant computing resources. This paper's main result is presenting a generalized effective parallel-and-recursive algorithm with the optimal bound of time complexity $O(log^2N)$ which solves in unified manner the variety of interrelated geometrical problems for the construction of visual models of complex phenomena and processes. This algorithm belongs to the class of the solvability NC_2.

Keywords: *parallel-and-recursive algorithm, visual model,* interrelated geometrical problems.

1. INTRODUCTION

The paper's main result is presenting the generalized effective parallel algorithm which solves in unified manner the variety of interrelated geometrical problems for the construction of visual models of complex phenomena and processes. Such an algorithm is based on well-known technique "divide-and-conquer". An algorithm consists of two basic stages: recursive dividing, which is general for the whole set of problems, and recursive merging. Steps of the merge stage could be executed simultaneously for each problem. The parallel computing model is asynchronous, so merge stage is executed independently for each problem. An application of this algorithm can be, for example, the modeling of thermo-physical and thermo-mechanical processes which arise up at welding. In this case it is necessary to build the exact visual model of the welding bath, the welding seam and region adjacent to them. To have an exact model of welding bath at every moment of time there is the need to simultaneously solve the following geometrical problems: constructing convex hull, triangulation, Voronoi diagram, searching for all nearest neighbors, nearest pairs, geometrical search, spline approximation and so on. Figure 1 shows a fragment modelling welding bath.

In this paper a paradigm is offered - an effective algorithm of parallel solution the set of interrelated geometrical problems for multitask modeling of complex dynamic processes. This paradigm is illustrated on the examples of Voronoi diagram and "all nearest neighbors" problem. It is important to note that main idea of this paper is not to describe the well-known paradigm "divide-and-conquer". In works [1-4] there have been described efficient parallel algorithms for solving some problems of computational geometry, including the mentioned above technique.

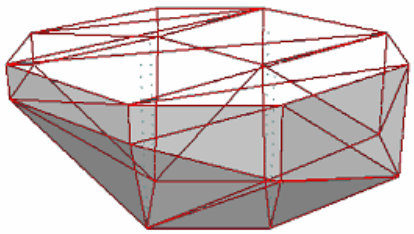


Figure 1: Fragment modeling welding baths: convex hull and triangulation.

For accurate modelling it is necessary to solve simultaneously a set of interrelated problems. The best strategy in this case would be one that uses a common tool for implementing the tasks: data structures, steps of procedures and presentation of results. The most suitable in this case, in our opinion, may be technique "divide-and-conquer". In this technique, the stage of dividing is a common and unified for all tasks, and at the merge stage is proposed to use a common and unified data structure (weighted concatenable queue) at which the procedures are performed quickly. Moreover, the results of the individual steps of some procedures used by other procedures, that ensures high efficiency.

Problem. Let S be the given set of N points in space E^d. It is necessary to develop the generalized effective parallel-and-recursive algorithm for simultaneous solving problems of computational geometry, which are defined on single set of data S, for which the low bound of complexity is $\Omega(NlogN)$ (for one processor machines).

2. THE GENERALIZED ALGORITHM

One of main application problems of the "divide-and-conquer" technique for solving problems of computational geometry is nonlinearity of the merge stage and linear inseparability of the set of points. In the considered approach for problems, due to the composition of successful data structures at the stage of preliminary processing and the use of parallel processing at stages of dividing and merge it is possible to construct the effective parallel-and-recursive algorithm which removes the restrictions specified above. We will consider the technique of the algorithm for the two dimensions case.

2.1 Mathematical model of the algorithm

The mathematical model of the offered parallel algorithm consists of such basic stages: *preliminary processing, dividing set*

of points (recursive descent)and recursive merge of results for subsets (recursive ascent)

Stage 1. **Preliminary processing.** Let S be the given set of N points in the plane $S = \{P_1, P_2, \ldots, P_N\}$ and there are $O(N)$ processors. The ordered array of points $U = \{P_{ij};\ i, j = 1,\ldots, N\}$ is formed at the stage of preliminary processing. Here, i is an index, which specifies the number of points in the list sorted by the x coordinate, and j is an index, which specifies the number of points in the list sorted by the y coordinate. Constructing the sorted lists for $O(N)$ processors can be carried out by means of one of the algorithms described in details in articles [5-7]. An array formed by such a method is given to the input of the algorithm. The graph of this algorithm it is binary tree, Figure2. In this graph, every node is marked by an integer number k. Number k divides the list of points in nodes into two lists of equal power, on the median, after comparison of the first indexes of points in the array U. And every number of node k is determined by the single iteration on a tree, if the quantity of points of the set is known.

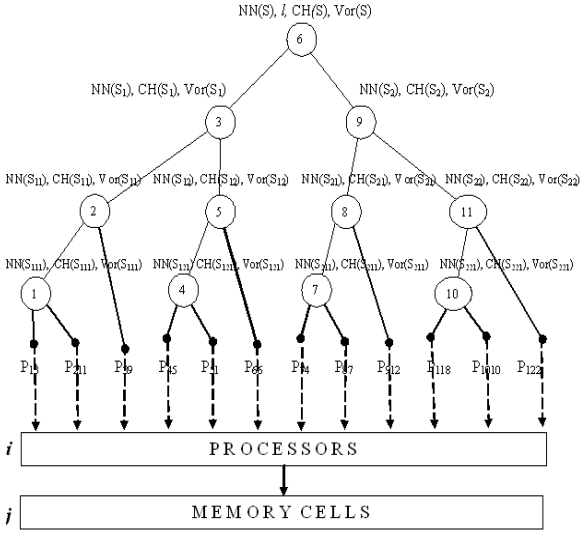


Figure 2: Graph of algorithm. $NN(S)$, $Vor(S)$, $CH(S)$ – merge procedures; l - median.

Stage 2. **Dividing the set of points (recursive descent).** This stage of an algorithm consists of dividing the set of points in the form of list U into equipollent subsets U_1, U_2, searching for a median l and transferring U_1, U_2 at the next step of recursion. The search median in sorted by x indexed array U is performed in constant $O(1)$ time. Time necessary for recursive descent in the parallel algorithm is defined by a following lemma.

Lemma 1. Using $O(N)$ processors it is possible to execute in time $O(logN)$: a stage of recursive separating the set S from N points on equal capacity of subsets S_1 and S_2 in the plane, search of a median l and transfer of subsets S_1 and S_2.

Proof. Let the given set of points to be presented in the form of an indexed two-dimensional ordered array $U = \{Pij\ ;\ i, j = 1,\ldots, N\}$. For constructing of such a structure of data it is possible to take advantage of parallel algorithm of sorting with complexity $O(logN)$, offered Colle [7]. Such a representation of points set allows constructing a tree of dividing, if the quantity of points N in the list U is known. According to the algorithm, first index i of each point P_{ij} is associated with processor number, and second index j is associated with the number of a memory cell in which a point is stored according to orderliness on y coordinate. On each step of dividing corresponding processors synchronously compare the first indexes from the list of points and dispatch points in corresponding nodes of algorithm, keeping thus the order of an arrangement of points which is defined by their order in memory cells. Considering precise orderliness of points P_{ij} in array U by both indexes and interrelation between processors and the memory elements, time of performance of merge process in each node of a tree will not exceed constant $O(1)$. Thus, general time of dividing will not exceed $O(logN)$ for the worst data input. As it was necessary to prove.

Stage 3. **Recursive merge of results for subsets (recursive ascent)**. At this stage, the merge procedures of related problems are running in each node of the algorithm graph. These procedures are building a general solution of problems. Process comes to the end with result of merge in root node. In the presented paper, due to the limits on the pages number, and that the step of division is common to all the problems we will consider example of merger procedures construction for "Voronoi diagram" and "all nearest neighbors". The main feature of the proposed procedures - is to use a common data structure "weighted concatenable queue". This data structure allows to perform all actions within the logarithmic time.

2.2 Constructing merge procedures

The merge stage of algorithm for constructing Voronoi diagram differs from the merge stage in the convex hull only on a finishing step. In the first case dividing chain is built, which connects the Voronoi diagrams of sons, and in the second case - it is finding of bridges (common tangent segments) to the convex hulls of sons. Thus the results of convex hulls constructing and bridges got for a problem "convex hull", is used for next steps of building Voronoi diagram. In addition, construction of dividing chain for the Voronoi diagram in parallel allows to solve the problem of finding all nearest neighbors.

2.2.1. Constructing merge procedure for "Voronoi diagram" problem. At every step of recursive ascent starting from the second, Voronoi diagrams **vor**(S_L) and **vor**(S_R) for subsets of points from the left and right sons, accordingly, are fed to the input of parent node v of the tree. It is necessary to build Voronoi diagram for the node v. Since the basic step of the constructing merger procedures for the Voronoi diagram and all nearest neighbor, in this algorithm, is building a monotone dividing chain, we offer one of the possible algorithms for its construction.

Constructing the dividing chain. The dividing chain constructing process is executed using $O(N)$ processors for the subsets of points, which are contained in a zone near dividing vertical line l (Figure 3). These subsets are located to the left and to the right from l, and belong to the mutually convex chains of convex hulls of sons and points determined by the edges of diagrams $vor(S_L)$, $vor(S_R)$, which cross the edges of these chains.

The time required for the construction of the dividing chain is determined by the following lemma.

Lemma 2. Constructing the dividing chain $\sigma(S_1, S_2)$, which "sews" together the Voronoi diagrams **vor**(S_L), **vor**(S_R) at every step of the merge stage can be completed in $O(logN)$ time using $O(N)$ processors.

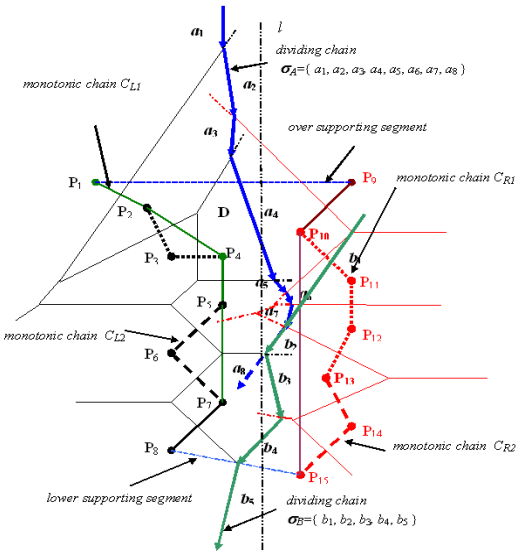


Figure 3. The merge step of two dividing chains (upper σ_A = { a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8 } and lower σ_B={ b_1, b_2, b_3, b_4, b_5 }chains) for the pairs of monotonous chains (C_{L1}, C_{R1}), (C_{L2}, C_{R2}), accordingly.

Proof. Between the upper and lower supporting edges of two convex hulls of Voronoi diagrams for sons **vor**(S_L), **vor**(S_R) of some node v of algorithm graph we have two mutually convex chains. They are determine the region of constructing the dividing chain, D. Each of these chains determines the ordered set of the Voronoi diagram edges, which are directed in into the region D, and they cross the edges of mutually convex chains CH_L, CH_R. Each edge of chains CH_L, CH_R can cross with one or a few edges of the Voronoi diagram and consequently, determines the set of points parted by these edges. We will note the set of vertexes as $B_L(S_1)$ ($B_R(S_2)$). It is consists of vertexes the convex chain $CH_L(CH_R)$ and the points, which are determined by edges Voronoi diagram that are crossing the chain. We will name this set as *left-maximum (right-maximum)* ordered set of points (or a list). If we connect the points in the lists $B_L(S_1)$ and $R(S2)$ consistently, we get lists of edges $E_L(S_1)$ and $E_R(S_2)$, which form chains S_L and S_R, respectively.

Lemma 3. Chains S_L and S_R are monotone in relation to direct l.

Proof. We will prove from opposite. It is known, that dividing chain $\sigma(S_1, S_2)$ is necessarily monotone in relation to direct l. Lets at least one of chains S_L and S_R will be not monotone in relation to l. Then there are edge this chain which will have the angle of rotation in relation to an OX axis with beginning at the end of this edge greater than π. It follows that the corresponding edge Voronoi diagram will not get into the domain D, and the dividing chain will not be monotone, which contradicts the condition.

It is important to mark that merge procedure in every node of algorithm tree can be executed on several processors independently and parallel. In order to execute such actions it is necessary: to determine a data structure which would support a convex hull and Voronoi diagram in every node, would allow to find supporting points, uncouple and couple parts of convex hulls, to conduct supporting segments and to build a dividing chain. As a data structure which would execute the operations mentioned above for logarithmic time, we have chosen the concatenable queue, the same data structure as in problem "Convex hull", with defined procedure MERGE (U_L, U_R), which allows to find supporting points and supporting segments, to build convex hull in $O(logN)$ time and with procedure which would allow to build a dividing chain. For organizing the process constructing the dividing chain, on the basis of lists $B_L(S_1)$ and $B_R(S_2)$ we will create the proper data structures (Figure 4), loading them with needed data.

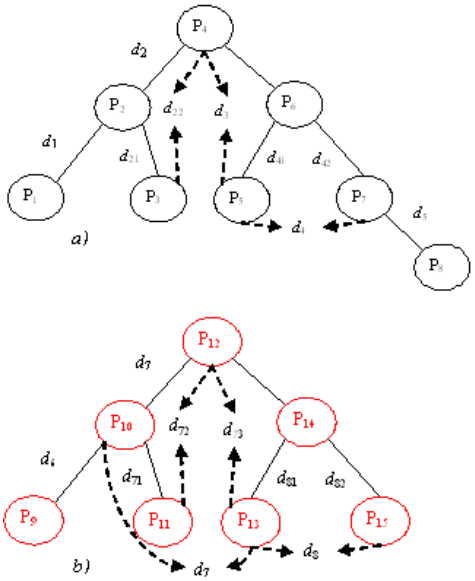


Figure 4. Data structures: the concatenable queues for left C_L and right the C_R chains of merge region D of the Voronoi diagrams: Vor(S_1), Vor(S_2) of figure 3, accordingly.

The concatenable queues of both monotone chains are binary trees with a root, in which nodes we have the coordinates of vertexes, and the arcs of which are the proper edges ek (k = 1,…, N) from the lists of chains $E_L(S_1)$ and $E_R(S_2)$. In addition, nodes are loaded by pointers on the proper edges Voronoi diagram (we will note them through d_{ij} ; i, $j \in N$). Such data structures allow to build a dividing chain $\sigma(S_1, S_2)$ using $O(N)$ processors in $O(logN)$ time. Graph algorithm for constructing a dividing chain can be represented as binary tree. In the leaves tree, each with $O(N)$ processors builds dividing chain for the corresponding pairs of edges (e_k, e_l) ($e_k \in E_L(S_1)$; $e_l \in E_R(S_2)$). The results are given to the next level of the tree where the step of merge is carried out. As a result, a dividing chain is built as connection the dividing chains of sons. The merge process of dividing chains sons requires in $O(1)$ time in every node tree. As we see all transactions for the

constructing of dividing chains require no more than $O(logN)$ time using $O(N)$ processors.

After determining the supporting points and uncoupling convex hulls of node v sons by them, the left and right parts of trees that support the convex hulls U_L and U_R between the supporting points are deleted respectively. The balanced trees, which will support the upper and the lower convex hull of node v are formed by merge parts of trees which remained. All operations are performed in $O(logN)$ time. The obtained trees support a convex hull and allow to execute the constructing procedures of dividing chain in $O(logN)$ time.

2.2.2. Constructing of merge procedure for "all nearest neighbors" problem. At each stage of recursive ascent, starting from the second, at the input of parent node v of the algorithm graph there are Voronoi diagrams (VD) from the left and right sons $vor(S_L)$, $vor(S_R)$, and the nearest neighbor for each point from subsets S_L and S_R. Voronoi diagram is built for the node v, and the new neighbors at the border of subsets of S_L and S_R are being determined simultaneously, relatively to median l. At the merging stage the algorithm, during the constructing the dividing chain of Voronoi diagram, we find simultaneously nearest neighbors among the points from sets S_L and S_R, which form the current pair of a chain edge $\sigma(S_L, S_R)$, Figure 5.

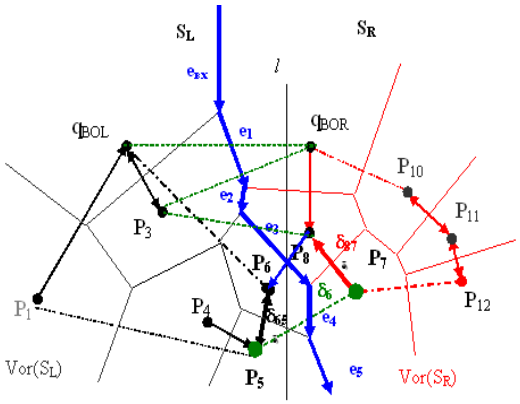


Figure 5. Search nearest-neighbors to pairs of points (P_3, q_{BOR}), (P_3,P_8), (P_6,P_8), (P_6,P_7), (P_5,P_7). For $P_8 \in S_R$ is found a new nearest neighbor $P_6 \in S_L$.

Thus, the edge of the dividing chain determines the next pair of points, which is being checked for the presence of the new nearest neighbors. Let $NN(S)$ be set of pairs the nearest neighbors for set S and $NN(S_L)$, $NN(S_R)$ - for sets S_L and S_R, respectively. The algorithm graph in this case is a binary tree, with the only difference that every node of a tree loaded except for the orderly array of points, median l; Voronoi diagrams $vor(S_L)$ and $vor(S_R)$ children, sets of the pairs points sons of nearest neighbors $NN(S_L)$ and $NN(S_R)$, respectively.

Lemma 4. The stage recursive merging of search results the nearest neighbor to each point of the set S of N points on the plane, using $O(N)$ processors, can be executed in $O(log^2N)$ time.

So as for constructing dividing chain in step merger it is enough $O(logN)$ time, then total complexity of step merger will be the same. This is because at each step constructing edge of dividing chain we must find the nearest neighbors for pair of points which he separates. For it we should compare only two distances.

3. IMPLEMENTATION OF THE ALGORITHM

For implementation of the algorithm is applied MPI and PAROS (Parallel Asynchronous Recursively Operated Systems) technologies (for PRAM model). Those technologies allow to simply and effectively implementing parallel-and-recursive algorithms for the solution of complex problems both on multiprocessing machines, and in a computer network.

4. CONCLUSION

In the given article is proposed approach which allows to develop effective and convenient means of automation the geometric modelling of complex phenomena and processes. The main feature of the implementation approach is that the parallel algorithm simultaneously executes both different steps of one procedure on many processors, and different procedures in one node. It allows develop the generalized algorithm of solution for number geometric problems by single technology.

5. REFERENCES

[1] A. Aggarwal, B. Chazelle, L. Guibas, C. O'Dunlaing, and C.C. Yap. *Parallel computational geometry*. Algorithmica 3. 1988. 293-327.Springer-Verlag New York Inc.

[2] N.M. Amato, M.T. Goodrich, and E.A. Ramos. Parallel algorithms for higher-dimensional convex hulls. In Proc. 35th Annu. IEEE Sympos. Found. Comput. Sci., pages 683-694,1994.

[3] M.J. Atallah and D.Z. Chen. Parallel computational geometry. In A.Y. Zomaya, editor, Parallel Computations: Paradigms and Applications, pages 162-197, International Thomson Computer Press, Boston, 1996.

[4] J.E. Goodman and J. O'Rourke, eds., Handbook of Discrete and Computational Geometry, Second Edition, Chapman and Hall/CRC Press, 2004.

[5] M. Ajtai, J. Komlos, and E. Szemeredi (1982). An O(nlog(n)) Sorting Network. Proc. 15th ACM Symposium on Theory of Computing, pp.1-9. Also in Combinatorica, 3(1)(1983), pp. 1-19.

[6] T. Leighton (1984). Tight bounds on the complexity of parallel sorting. Proc. 16th ACM Symposium on Theory of Computing, pp.71-80.

[7] R. Cole (1986). Parallel merge sort. Proc. 27th IEEE FOCS Symposium, pp. 511-516.

About the author

Vasyl Tereshchenko is a professor at National Taras Shevchenko University of Kyiv, Faculty of Cybernetics. His contact email is v_ter@ukr.net.

The Experience of The "Dust" C Code Implementation on PS3

Vladimir Savchenko
Faculty of Computer and Information Sciences, Hosei University
Tokyo, Japan
vsavchen@k.hosei.ac.jp

Abstract

This paper describes an experiment in learning computer graphics algorithms through parallelizing three C code examples: Ray Tracing, Voxel Visualization, and Image Warping on PlayStation3 (PS3). An implementation of geometry subdivision of data for PS3, which is a physically distributed memory machine, provides, for example, display of a 120×120×120 voxel slab at 30 frames per second. We notice that all the code conversions were done by novices without Cell programming experience in a quite short time.
Keywords: PlayStation3, Cell, Educational approach

1. INTRODUCTION

Computer graphics (CG) education is increasingly important to supply the growing needs of the film, games, and virtual reality industries. Learning goals in CG and concepts that would be generally expected in even a beginning course could be covered at several levels, from the algorithmic and mathematical to a general conceptual treatment followed by programming realization.

During the last few decades, many educational practitioners have increasingly turned their attention to *constructivist* models of learning. Constructivist pedagogies have been successfully used in learning situations where the acquiring of a deep understanding of a subject is required [1],[2].

A common educational approach is to use books and other educational components to give students a theoretical foundation, and a low-level programming library like OpenGL or use Java – developing of animated applets – for practical exercises and projects. This paper describes our experiment of using a constructivist-oriented approach in computer graphics education. Our main premise or "learning model" is that the fundamental concepts and algorithms of CG can be acquired through the experience with parallelizing old C codes related to CG. Also, we present time performance results for running three CG applications using parallel processors of PlayStation3 (PS3) to achieve almost linear speed up of calculations.

Chips with two processors, or cores, are now standard, and four-core chips are increasingly common. But right now, even with Dual-Core machines being common, most PC programmers don't know how to use parallel processing. In spite of the obvious fact that microprocessor companies take a huge risk in adopting the multicore strategy, recently, two rather cutting-edge hardware consoles (XBox 360 and PS3) with heavy parallel processing power have been launched.

The Xbox 360 (the second video game console produced by Microsoft) provides uniform memory and three completely similar PowerPC processors. The PS3 uses the cell microprocessor developed by IBM, Sony and Toshiba as its CPU, utilizing seven of the eight "synergistic processing elements" (SPEs). In PS3, six SPEs are only available because one SPE is reserved by PS3 OS and the other is disabled for improving yield. At peak performance, the chip reaches up to 200 GFlops. It has a heterogeneous multicore architecture that consists of a Power PC processor along with SPEs that are specialized for data processing. PS3 is not only one of the most advanced gaming devices in the world, oriented on polygon processing, but it is thought that PS3 is also the perfect tool for CG and image processing applications having much quantity of data processing in such applications such as the real time rendering of high-definition images, medical image processing, encoding and/or decoding a digital data stream, and in many various applications not related to games. Moreover, an improved version of the cell chip is also the basis of Roadrunner - the fastest computer in the world [3].

Recently, many universities (for instance, the Department of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology [4]) in order to provide a framework for research and teaching in the areas of multicore development and parallel programming have started projects on the PS3. The U.S. Air Force bought 300 PS3s for a research purpose [5]. IBM is also running a programming contest for college and university students in 25 different countries.

Our goal is three-fold, first of all, as we mentioned above our goal is to attain a deep understanding of a subject, in particular CG algorithms. Second, is to give undergraduate students the chance to get hands-on experience on PS3 hardware. Many processing algorithms such as voxel visualization, image processing are "embarrassingly parallel" or almost ideally suited to parallelization while even a simple ray tracing or ray marching for implicitly defined objects creates a problem with load balancing - distributing work dynamically. The fact that Cell-based systems are attractive for CG applications stimulated us to implement PS3 for educational and research purposes.

Finally, we are interested in the use of so called "dust code" in finding a way to make that old our programming code useful to provide a framework for research and teaching in the areas of geometry modeling and CG.

Parallelization of CG applications is mainly achieved by inserting directives in the source code.

The remaining part of this paper is arranged as following: information and related literature review on the key techniques used is given in section 2. Section 3 presents the benchmark models. PS3 implementation is discussed in Section 4. The benchmark results are given in Section 5. And Section 6 concludes the paper.

2. BACKGROUND INFORMATION AND RELATED WORK

Multiple data (SIMD) ray kernels and distributing work dynamically can produce ray-traced images (without detail-

adding of reflective maps and shadows) at interactive frame rates for complex scenes containing more than one million polygons [6], see also [7].

PS3 Cell contains one Power Processor Element (PPE—a standard PowerPC CPU) and eight Synergistic Processor Elements (SPEs). SPEs live inside *elements* on the Element Interconnect Bus, which also talks to the Memory Interface Controller and I/O controllers. This path is very fast, DMA between elements and the 256 MByte XDR system memory achieves a peak rate of 25.6 GBytes/sec. The Cell's SPEs provide very fast execution, but their local store holds only 256 Kbytes. Typically processing on the SPE involves shuttling data to and from main memory using DMA. DMA double buffer allowing the SPE's memory flow controller to coordinate the best order of operations for loading and storing is used. For more information, see [8],[9].

Cell Broadband Engine processor provides a cost-effective alternative to current ray-tracing options and helps free studios from the limitations of low-fidelity, raster-based graphics approximations in their final production frames, see [10].

3. MODELS

For some computer graphics applications, sequential implementations are too slow since computation grows as the square or cube of a modeled space. Faster approaches often rely on subdivision of 2D or 3D space where a partitioning of space is used to accelerate calculations. This partitioning - often simple geometry subdivision of 2D screen - in many applications forms the basis of scalable multi-processor approaches to visualize a scene or simulation.

3.1 EXAMPLE I. RAY TRACING

Ray tracing is one of the techniques to draw an image of a geometry scene. In our implementation of a network of Cell processors, a renderer calculates and renders a scene that consists of multiple implicitly defined spheres that includes coordinates of geometry objects, positions of a source of light and viewpoint, and others. The renderer can finely express reflectance, transparency, and refractions. This stage is the most complicated and CPU-intensive: calculation of lighting and shadowing of the objects. The ray tracing procedure is pretty simple in some way. It goes from one pixel to another; however, computational complexity naturally increases with respect to the image size. Also, when one ray communes with a lot of objects, hidden surface removal performed by projecting nearest point from a viewpoint on plane of projection, calculating reflectance maps and many others related to ray tracing techniques are computationally expensive.

3.2 EXAMP LE II . VOXELIZATION AND VOXEL VISUALIZATION

In many CG, animation, and geometry applications continuous volume is represented by a 3D grid of voxels, see, for example, [11]. This representation is convenient for a number of applications such as mixing synthetic objects into medical imagery, 3D-visualization of tomography volume data, reverse engineering, and scientific visualization. In these fields, 3D data is measured or computed at a large number of points in 3D space and then rendered to produce informative images. The main weakness of the voxel-based approach is that it is expensive in terms of computation and memory. Interaction is an effective technique for guiding the production of these images, allowing a doctor or scientist to navigate to various regions of interest and adjust the transfer function used to map the data into colors. Unfortunately, due to the size of these 3D datasets and the processing required to render them, interaction has only been possible with relatively small data sets and not of the most general organization. In particular, regular rectilinear grids of volume elements, or *voxels*, are well-suited to acceleration by modern graphics hardware. This hardware can take the form of a dedicated volume processor [12] or a more generic 3D graphics accelerator [13], see, also [14].

For data of the right form and size, this hardware makes interaction possible. Nevertheless, volume data often comes in many sizes and forms that creates difficulties of using specific volume processors. Also, interaction is an effective technique for guiding the production of new volume models, see, for instance, [15] and references therein, that is considered as a continuation of the project.

3.3 EXAMPLE III. IMAGE WARPING

One of the image processing techniques is the image warping, see, for instance [16]. Image warping is the process of digitally manipulating an image where the image coordinates are transformed from pixels (i,j) to coordinates (x,y) of destination image. Early interest in this area dates back to the mid-1960s when it was introduced for geometric applications, for example, distortion correction of lenses in remote sensing. It has found a new surge of interest from the CG field. This is largely due to increasingly powerful computers that make warping a viable tool for image synthesis and special effects. The most usual technique used in warping algorithms is based on spline interpolation. So, conventional image processing takes much time because the computer calculates a mapping over an image by calculating a spline; in our application, we apply warping technique based on the use of radial bases functions [17] that allows effectively produce forward and inverse mappings. Thus it seems reasonable and interesting for students to investigate opportunities of the use of PS3 and study numerical algorithms for calculating maps from the destination image to the source image in parallel and to check up speedup with respect to processing speed on a single processor.

4. PS3 IMPLEMENTATION

Load balancing is a central theme of multiprocessor-based systems for managing memory layout and assigning processors to jobs. In considered here applications, fortunately, there is no spatial and time dependency between processed data, therefore processors can be updated in parallel in any order. The master-slave (or host-node) paradigm is used for load balancing to speed up calculations. It means that a separate "control" program (on PPE) called the master is responsible for process spawning, initialization, collection, and display of results. The slave programs (on SPEs) perform the actual computation involved; their workloads are allocated by the master dynamically. There is no room to show relative simplicity of software development. Nevertheless, let us notice that the main master program includes two functions – startRayTracing00 and getJob – which provide , first of all, a preparation for SPE by spe_image_open, spe_context_create, spe_program_load, and pthread_create functions which are included in startRayTracing00. Secondly, ray tracing is performed by SPEs where each thread takes control of sending and getting necessary data. PPE is also used for ray tracing as well as SPEs. Processing data are distributed by getJob

function. Synchronization is taken on each thread by using the mutex method in getJob function that provides correct distribution of job assignments. SPEs use DMA transfer that allows to access system memory for writing RGB data directly to a pixels array. It makes possible performing a complete image directly by SPEs without composing a final image by PPE.

The DMA transfer controller is used to send data going to and from the main memory. 16 bytes multiple sequence alignment is used both on local store and main memory to provide the DMA transfer. Moreover, the size of data that can be sent by one time is up to 16 Kbytes. Because information transmitted in the ray tracing application is integer type (four bytes that show coordinates of the pixel and the color of the pixel) 4,000 of data or less can be sent at once. The automatic generation of accurate multiple alignments is potentially a daunting task. Nevertheless, experiments show that an improvement of the processing speed is not achieved when the number which can be sent is increasing from 1,000 to 4,000 while the processing speed is improved when this value is increasing from 1 to 1,000. Thus, size of parcel data to be sent at once was chosen to be 1,000. Let us notice that Phong [18] model was used.

The SPEs are the Cells short-vector SIMD workhorses. It is important to exploit SIMD properties of SPEs as efficiently as possible to speed up the processing speed of a SPE program. While it is not quite clear how we can harness the power of the Cell and PS3 by using SIMD operations in ray tracing application, existing voxelization/visualization C code can be easily modified to add new capabilities for model voxelization and real time visualization.

5. PERFORMANCE EVALUATION

The processing speed measurements were produced for the Cell processor and the Pentium processor for the examples shown in Figures 1, 2, 3.

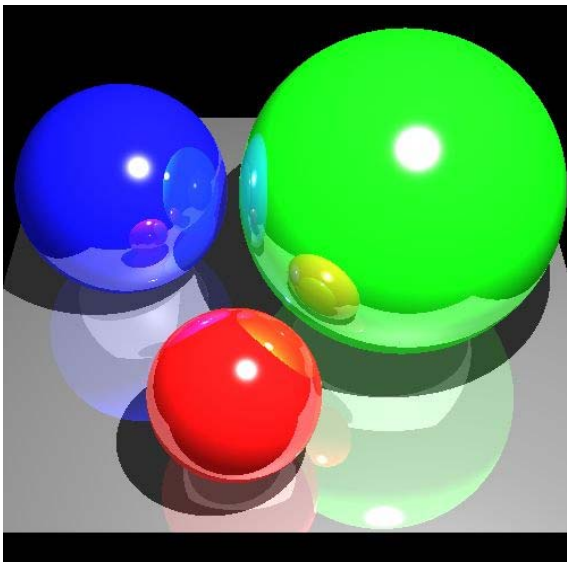


Figure 1. Ray-traced objects are rendered with detail-adding of reflective maps and shadows

The performance results are shown in Tables 1,2,3; time is given in seconds.

Table 1. Measurement results for Ray Tracing

#PUs	Processing time
Pentium 4 3.2GHz	7.81
Pentium M 1.6GHz	14.27
PPE	5.60
PPE + 1SPE	3.36
PPE + 2SPE	2.44
PPE + 3SPE	1.93
PPE + 4SPE	1.60
PPE + 5SPE	1.36
PPE + 6SPE	1.18

The image of a voxel model (defined by subtraction and union operations of three implicitly defined spheres) produced by the developed software algorithm is shown in Figure 2. The execution time of voxelization and vizualization by using 6 processors is 0.1097 sec. The execution time of processing without using Cell was 1.3075 sec. Thus, the speed of processing using the cell is about order of magnitude higher than without using the cell processor that is explained by using SIMD operations for voxelization of the model.

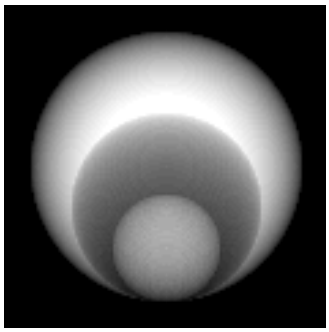


Figure 2. Result of voxel visualization. Size of the voxel slab is 120x120x120

Table 2. Processing characteristics

#PUs	Time of voxelization	Time of visualization
1SPE	0.4268	0.1103
2SPE	0.2177	0.0688
3SPE	0.1438	0.0473
4SPE	0.11	0.0459
5SPE	0.0891	0.0428
6SPE	0.07	0.0397
PPE	1.1938	0.1137

Table3. Warping test results

#PUs	Processing time
1SPE	1.96
2SPE	0.99
3SPE	0.65
4SPE	0.50
5SPE	0.41
6SPE	0.35
PPE	3.05

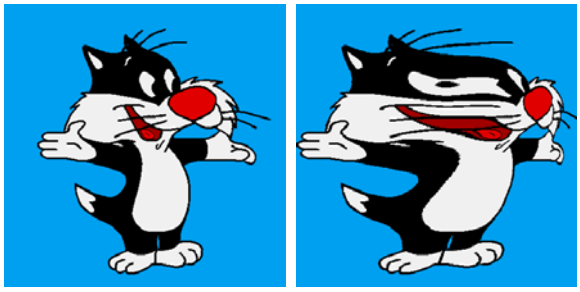


Figure 4. Warping of "Neko" image. Image size is 407×411

Let us notice that almost similar programming approach discussed above for Ray Tracing was used for voxel vizualization and image warping.

6. CONCLUSION REMARKS

The PS3 Cell processor has unique architecture; therefore a special treatment of programs developed for conventional computers is needed. Comparison of performance of Pentium 4 3.2GHz processor and using 6 Cell processors shows that the Cell processor system demonstrates about 7 times performance speed up of Pentium 4 processor. It has been said that the processing speed of Cell processor is from several to tens times faster than Pentium D processor. Nevertheless, we were not able to attain ten times acceleration of ray tracing by the use of Cell processor. Without any doubt, we can state that we succeeded in the using Cell processor with respect to Pentium 4 which performance is lower than Pentium D. Cell processor shows about 7 times performance speed up of Pentium 4 processor and about 12 times performance speed up of Pentium M processor. Moreover, because the processing speed has improved by increasing the number of used SPEs almost linearly, it seems that multiple data alignment procedure is reasonable and accurate.

However, the improvement of the processing speed by SPEs was less than our expectation. We suppose that there are two reasons of it. First one is related to uneasy question how to assign processing operations. SPE works well on simple operations, that is conditional branches must be eliminated. The second is that SIMD operations were not used in the software development of ray tracing and image warping algorithms. In future, we are planning to rewrite the source code of the SPE`s program which uses only scalar operations by using SIMD operations. It is a serious problem because of the initial code was not accustomed to the idea of the vector data. It seems that without proper compiling tools SIMDimization process can be a serious obstacle for a wide application of Cell processors.

The results presented are from a short initial evaluation. There is still much scope for optimization and these results should be considered as representative of the performance that can be obtained from the first naïve parallelizing of old C codes.

The main lesson learned is the fact that students with no background in Cell parallel programming, were able to get their projects done from scratch in just about three months. This largely goes to show that we can expect rather simple migration of "embarrassingly parallel" CG applications to the Cell processor.

ACKNOWLEGEMENTS

I would like to thank my former undergraduate students Y. Mukai, T. Suganami, and I. Tatsuma, for their encouragement to participate in this project.

REFERENCES

[1] C.T. Fosnot, Ed. Constructivism. Theory, Perspectives, and Practice. Teachers College P, 1996.

[2]E. Von Glasersfeld. Radical Constructivism and Teaching,2001,http://www.umass.edu/srri/vonGlasersfeld/onlinePapers/html/geneva/

[3] http://www.lanl.gov/roadrunner/

[4] http://www.ps3cluster.org/index.html

[5] https://www.fbo.gov/

[6] C. Benthin, I. Wald_, M. Scherbaum, H. Friedrich, Ray Tracing on the Cell Processor**, ,** IEEE Symposium on Interactive Ray Tracing**,** 2006, pp. 15-23.

[7]Real-Time Ray Tracing on the Playstation3 Cell Processor, http://eric_rollins.home.mindspring.com/ray/ray.html

[8] IBM, Cell Broadband Engine Programming Handbook, Version 1.1, April 2007.

[9] IBM, Cell broadband Engine Architecture, Version 1.01, October 2006.

[10]Interactive Ray Tracer for Cell Broadband Engine, http://www.alphaworks.ibm.com/tech/irt?open&S_TACT=105AGX59&S_CMP=GRsite-jw18&ca=dgr-jw18awirt

[11] B. Lichtenberg, R. Crane, and S. Naqvi, Introduction To Volume Rendering, Prentice Hall PTR, 1998.

[12] H. Pfister, J. Hardenbergh, J. Knittel, H. Lauer, and L. Seiler. The VolumePro Real-Time Ray-casting System. Proceedings of SIGGRAPH 99. pp. 251-260.

[13] B. Cabral, N. Cam, and J. Foran. Accelerated Volume Rendering and Tomographics Reconstruction Using Texture Mapping Hardware. Proceedings of 1994 ACM Symmposium on Volume Visualization. pp. 91-98.

[14] U. Neumann, Interactive Volume Rendering on a Multicomputer, Proceedings of the 1992 Symposium on Interactive 3D Graphics, SI3D '92, March 29 - April 1, 1992, Cambridge, MA, USA. ACM, 1992, pp. 87-93.

[15] T. A. Galyean and J. F. Hughes, Sculpting: An Interactive Volumetric Modeling Technique, Computer Graphics, Vol. 25, No. 4, 1991, pp 267-274.

[16] G. Wolberg, Digital Image Warping (Systems), Wiley-IEEE Computer Society, 1990.

[17] V.A. Vasilenko, Spline Functions: Theory, Algorithms,, Programs, Nauka Publisher, Novosibirsk, 1983, (in Russian)

[18] J. F. Blinn, "Models of Light Reflection for Computer Synthesized Pictures". Proc. 4th annual conference on computer graphics and interactive techniques, 1977, pp. 192-198.

Анализ развития концепций и методов визуального представления данных в научных исследованиях задач вычислительной физики

А.Е.Бондарев, В.М.Чечеткин
Институт прикладной математики им. М.В.Келдыша РАН
Москва, Россия
bond@keldysh.ru; chech@gin.keldysh.ru

Аннотация

В данной работе рассматриваются основные стадии развития научной визуализации как самостоятельной научной дисциплины. Рассматривается эволюция и перспективы развития основных концепций, методов и подходов визуального представления результатов численных исследований задач вычислительной физики, в первую очередь, задач механики жидкости и газа.

Ключевые слова: *научная визуализация, концепция визуального представления, методы визуализации .*

1. ВВЕДЕНИЕ

Данная работа представляет авторский анализ эволюции и перспектив развития концепций и методов визуального представления результатов численных исследований задач вычислительной физики, в первую очередь, в механике жидкости и газа. Данный анализ основан преимущественно на отечественных разработках в области научной визуализации.

Интенсивное развитие вычислительной техники и методов математического моделирования сформировали новую научную дисциплину, которая получила название ***научная визуализация*** (***Scientific visualization).*** Развитие научной визуализации как самостоятельной дисциплины, востребованной во всех без исключения областях знания, науки и инженерных технологий, заняло два десятилетия. Это обусловлено тем, что развитие методов, алгоритмов и подходов визуального представления численных данных происходило на каждом этапе как ответ на реальные потребности науки и техники к этому моменту. Данное развитие соответствовало уровню сложности решаемых задач, уровню развития вычислительной техники, уровню развития математических методов.

Одновременно развивались концепции визуального представления численных данных, соответствуя когнитивному уровню и потребностям науки и техники в соответствующий период. Данная работа рассматривает процесс исторического развития научной визуализации как отдельной дисциплины и развития концепций визуального представления численных данных. Также рассматриваются современные потребности и перспективы развития методов визуального представления.

2. РАЗВИТИЕ НАУЧНОЙ ВИЗУАЛИЗАЦИИ КАК ОТДЕЛЬНОЙ ДИСЦИПЛИНЫ

Термин ***Научная Визуализация*** (***Scientific visualization)*** понятен на интуитивном уровне – всем ясно, что речь идет о визуальном представлении результатов научных исследований с помощью средств компьютерной графики.

Научная визуализация переводит результаты научных исследований, выраженные в численной форме, в визуальные образы. Визуальное представление численных данных облегчает работу с информацией и обмен этой информацией. Но главная цель – увидеть, то, что раньше нельзя было увидеть. Иначе говоря, увидеть - невидимое.

Увидеть невидимое хотели всегда – задолго до появления компьютеров. Большая часть экспериментов в физике, механике жидкости и газа, теории упругости имела своей целью не только получить и измерить количественные характеристики явления, но и увидеть их, сделать физические процессы видимыми. Можно вспомнить в ядерной физике камеру Вильсона, в которой исследуются траектории ядерных частиц с использованием фотографий этого явления. Можно вспомнить экспериментальные работы в механике жидкости и газа, где использовалось подкрашивание ламинарных потоков в каналах, наклеивание длинных шелковых нитей на модели и макеты, размещаемые в аэродинамических трубах.

Существует аналогия между численным и физическим экспериментом. В физическом эксперименте во многих случаях именно оптическая картинка является основным первичным результатом исследований. В области аэродинамики фотография картины обтекания является основным или единственным первичным результатом исследований на некоторых экспериментальных установках. В испытательных камерах аэробаллистических установок производится искровая фотография модели с применением теневого прибора, прибора Теплера или с применением интерферометра. Эта фотография часто является единственным первичным результатом эксперимента. Научный результат получают путем качественной и количественной обработки фотографии. В теневом приборе интенсивность почернения на фотографии пропорциональна второй производной плотности. Это позволяет выделить скачки уплотнения, волны разрежения, границы областей отрывного течения. Прибор Теплера реагирует на первую производную плотности, а интерферометр реагирует на саму величину плотности и позволяет выделить линии равной плотности.

Отметим, что задолго до появления компьютеров и численных экспериментов, сама суть получения результата в физическом эксперименте основана на анализе поля течения, отражающемся в действии экспериментального прибора. Основные положения, результаты и постулаты механики жидкости и газа были во многом получены таким экспериментальным путем. Многие из них отражены в известном альбоме течений жидкости и газа Ван-Дайка [1].

Физическая визуализация в экспериментах была основным источником информации о явлениях и процессах до появления математического моделирования. С появлением и развитием численного эксперимента, который стал более дешевым и маневренным инструментом познания физической картины мира, физический эксперимент остался основным средством контроля и верификации численных исследований. Продолжают развиваться разнообразные экспериментальные методы визуализации прозрачных сред [2]. Согласно [2] «…Известные достоинства оптических методов - высокая чувствительность, бесконтактность применения, отсутствие влияния на параметры исследуемого явления, возможность проведения качественных и количественных измерений - способствовали использованию их при визуализации и изучении неоднородностей в прозрачных средах…Задача визуализации неоднородностей в прозрачных средах состоит в том, чтобы изменения фазы проходящего через них волнового фронта преобразовать в изменения освещенности изображения исследуемого объекта, которые легко регистрируются …»

С появлением компьютеров возникла возможность проводить численные эксперименты, основанные на численном решении уравнений, используемых в математической модели исследуемого физического явления или процесса.

Роль методов и алгоритмов визуального представления результатов численного моделирования в вычислительной физике изначально рассматривалась как вспомогательная. Визуализация имела две основные вспомогательные функции:

а) обеспечение контроля и лучшего понимания численных результатов;

б) иллюстративная функция, облегчающая запоминание и ориентацию в обсуждениях и дискуссиях.

В настоящее время из иллюстративного и по своим функциям вспомогательного инструмента научная визуализация превращается в полноправный инструмент познания, более того, зачастую становится единственным инструментом способным прояснить суть моделируемого физического процесса.

Это превращение происходило следующим образом. Соответствующие времени технические требования к расчетам вызывали необходимость развития вычислительных методов для решения все более сложных задач. Параллельно с этим развивались методы визуализации численных решений в этой области знания.

Реальное рождение научной визуализации и компьютерной графики было обусловлено появлением во второй половине XX века графопостроителей. Это дало возможность не только автоматически представлять графики на бумаге, но и породило развитие программных средств визуального представления численных данных. Именно в это время появляются такие пакеты графических программ как Графор[3], СМОГ [4], Графал, Plot10.

Здесь начинается развитие программного обеспечения, позволяющего представлять результаты научной и конструкторской работы визуально.

На этом этапе в вычислительной механике жидкости и газа решались трехмерные стационарные задачи и двумерные задачи, носящие, хотя и нестационарный характер, но устанавливающиеся в процессе расчета к устойчивой картине течения. Появившаяся возможность графического представления этих решений привела к развитию инструментов визуализации.

Необходимо отметить неоценимую роль для развития отечественной науки и техники в советский период, которую сыграл программный комплекс Графор, разработанный в Институте прикладной математики им. М.В. Келдыша РАН под руководством Ю.М. Баяковского. Представленный в виде библиотек этот программный комплекс позволял квалифицированному пользователю не только иметь первичный инструментарий для визуального представления двумерных результатов и одномерных графиков функций, но и реализовывать на его основе собственные надстройки, позволяющие визуализацию широкого круга задач в самых различных областях знания.

Однако с течением времени по мере совершенствования вычислительной техники и усложнения решаемых задач стало очевидно, что для получения реального результата уже нельзя сначала просто рассчитать его, а потом применить изобразительные средства. Стало понятно, что применение и развитие средств визуального представления численных данных является самостоятельной ветвью науки, равно необходимой во всех остальных областях знания.

Именно этот момент и следует рассматривать как точку становления научной визуализации как самостоятельной научной дисциплины. Об этом рассказывал на проводившемся в 2000 году в рамках 10-й Международной конференции ГРАФИКОН-2000 круглом столе «Применение методов научной визуализации в прикладных задачах» С.В.Клименко (Институт физики высоких энергий, Протвино):

«…все начиналось с проблемы визуального представления научных данных и результатов расчетов. Эта проблема была сформулирована на дискуссии в 1987 году, которая состоялась в феврале. Эта дискуссия была собрана по инициативе ACM SIGGRAPH IEEE Computer Society Technical Committee of Computer Graphics. На этой дискуссии собрались люди, и они сформулировали основные задачи этой новой науки, которая возникла в связи с тем, что в Штатах тогда начали довольно мощно использовать суперкомпьютеры, и они поняли, что то количество цифр и чисел, которое выдают суперкомпьютеры невозможно переварить без использования каких-то новых методов, новых технологий, новых средств…» [5].

В 1987 г. национальным научным фондом США была создана Инициатива ViSC (Visualization in Scientific Computing), нацеленная на создание систем визуализации. Эти события положили начало развитию отдельной научной дисциплины – научной визуализации, обладающей своими научными форумами, своими органами научной печати, своими профессиональными научными сообществами.

На современном этапе развития параллельно с развитием средств вычислительной техники значительно усложнились задачи вычислительной физики. Это, как правило, двумерные и трехмерные нестационарные задачи, обладающие сложной геометрией. Часто подобные задачи предполагают слияние различных физических и математических моделей. Данные математические модели могут быть реализованы на сложных неструктурированных сеточных разбиениях. Особое

значение получают алгоритмы решения обратных задач, имеющих своей целью обычно оптимизацию явлений, процессов, определяющих параметров задачи, различных конструктивных параметров и т.п. Эти задачи предъявляют совершенно новые требования к роли визуализации в процессах численного решения задач математической физики.

Следует особо отметить, что развитие программных средств визуализации всегда носило догоняющий характер. Визуальное представление численных решений зависело от возможностей вычислительных ресурсов, в первую очередь, а во вторую – от уровня решаемых задач математического моделирования и используемых при этом численных методов.

В попытках найти новые пути развития в 2000, 2003 годах в рамках Международной конференции ГРАФИКОН были организованы круглые столы по проблеме развития и применения научной визуализации в прикладных задачах. Инициатором этих форумов являлся Институт прикладной математики им. М.В.Келдыша РАН, организаторами и модераторами выступали А.Е. Бондарев, Ю.М.Баяковский, В.А.Галактионов, С.В. Клименко. Эти мероприятия собрали специалистов их разных стран из разных областей науки и техники: от ядерной физики до биологии, медицины и фармацевтического производства. По ходу обсуждений было отмечено, что с развитием вычислительных средств, математических методов и сложности решаемых задач изменилась основная цель применения средств визуального представления: «Не проиллюстрировать уже полученный результат, а получить и понять его с помощью методов и подходов научной визуализации». Также отмечалось, что одной из важных функций научной визуализации является сохранение и накопление научного опыта решения задач в различных областях науки и техники.

Одним из важнейших результатов данных форумов было обсуждение выработки общих критериев визуального представления. Был выдвинут и поддержан следующий тезис: **«Необходимым признаком научной визуализации является наличие визуальной концепции».** Подобно тому, как по мере развития вычислительных средств, исследователю было необходимо обосновывать физическую и математическую модели явления, выбор численного метода, счетной области, сеточного разбиения, так сейчас и в будущем необходимо заранее разрабатывать и обосновывать визуальную концепцию будущего представления и анализа результатов.

Было введено важное понятие – **информационной ценности визуального образа**. Наряду с выразительностью и корректным отображением физического явления или процесса визуальный образ должен обладать информационной ценностью для исследователя.

3. РАЗВИТИЕ КОНЦЕПЦИЙ ВИЗУАЛЬНОГО ПРЕДСТАВЛЕНИЯ ДАННЫХ

Параллельно развитию вычислительных средств и математических методов, параллельно усложнению задач науки и инженерии проистекало развитие концепций визуального представления численных данных.

В период, когда в вычислительной математике и прикладных областях рассматривались в основном двумерные стационарные и нестационарные задачи, концепция визуального представления численных решений этих задач была предельно проста. Ее можно примерно сформулировать так: «Полученные в расчетах данные существуют в виде двумерных полей скалярных или векторных величин. Эти двумерные поля изображаются соответствующими программными средствами. Скалярные поля в виде изолиний или цветового представления пространства между изолиниями. Векторные поля изображаются стрелками, имеющими масштаб и направление, соответствующими представляемой векторной величине, а также линиями тока».

В определенный период эта концепция удовлетворяла прикладным целям и задачам. Дальнейшая детализация визуального представления была возможна путем различных комбинаций в рамках данной концепции.

Недостатком данной концепции являлось то, что коллективы ученых, конструкторов, инженеров и коллективы разработчиков графических средств визуального представления численных данных существовали в рамках вышеизложенной концепции абсолютно раздельно. Это существенно мешало развитию.

Основные принципы построения системы визуализации для анализа двумерных скалярных и векторных полей были успешно сформулированы в работе [6]. Также в этой работе было перечислено необходимое функциональное содержание подобной системы визуализации.

При усложнении алгоритмов и задач даже для двумерных данных рамки вышеописанной концепции оказались тесны. Требовалось расширить возможность работы с двумерными данными. Стандартное представление двумерного скалярного поля в виде трехмерной поверхности F(x,y) также не позволяло в полной мере выразить структуру полученных расчетных данных. Предлагались различные способы квазитрехмерного визуального представления двумерных данных. Основной целью являлось расширение рамок двумерной концепции.

Для задач обладающих осевой симметрией предлагался следующий вариант псевдотрехмерного представления: создать из двумерной сетки трехмерную фигуру вращения и получить таким образом полезное трехмерное визуальное представление двумерной задачи.

Предлагалось, например, последовательно расположить перпендикулярно оси времени набор двумерных сечений, взятых в разные моменты времени, и, соединив изолинии соседних сечений по времени, получить некую квазитрехмерную картину.

Также предлагался следующий подход. Пусть распределение скалярной величины A будет фоном, на нем представим линиями тока векторную величину B, толщина этих линий будет отображать скалярную величину C, вдоль линий тока цветом изобразим распределение величины D, вспыхивающими маркерами представим E, и так далее.

Основная проблема состояла в том, что практическая реализация подобных подходов продуцировала либо абсолютно нефизичную, искусственную картину, либо образ, перегруженный избыточной нескомпонованной информацией.

В период, когда задачи математической физики, математического моделирования, инженерные задачи в

прикладных областях стали по большей части трехмерными и нестационарными, в научной визуализации произошла естественная попытка автоматически перенести концепцию визуального представления двумерных данных на трехмерный случай, а также автоматически перенести на трехмерный случай уже разработанные методы визуального представления. Естественно, визуальное представление трехмерных нестационарных процессов имело свою специфику и сразу выявило ряд проблем.

В работе [7] описаны традиционные методы визуального представления численных решений трехмерных нестационарных задач для такой области знания как вычислительная аэрогазодинамика. Трехмерность моделируемого нестационарного процесса создает существенные трудности для визуального представления результатов, особенно там, где вычислитель не обладает априорной информацией о характере течения. Современные трехмерные нестационарные задачи вычислительной физики во многом напоминают задачи «черного ящика», когда без применения алгоритмов визуального представления и контроля получаемых в процессе расчета результатов даже для опытного расчетчика трудно полноценно представить, что же в счетной области происходит. Таким образом, для представления решения трехмерной задачи необходимо использование комбинированных алгоритмов и подходов графического представления, иначе обеспечить понимание качественной картины течения сложно.

В качестве попытки преодолеть ограничения, накладываемые автоматическим переносом двумерной концепции визуального представления на трехмерный случай, как направление развивалось создание ориентированных на конкретный класс задач вычислительных комплексов. Эти комплексы обладали модулями научной визуализации, предназначенными покрывать все возможные потребности вычислителя в данном ограниченном классе задач. В подобные модули стремились сконцентрировать все существующие средства визуального представления численных данных и все возможные комбинации этих средств.

Успешным примером реализации такого подхода можно рассматривать информационно-вычислительную систему, описанную в [8], предназначенную для моделирования двумерных и трехмерных нестационарных задач вычислительной гидродинамики. В среде этой системы реализовано построение сложных композиционных сцен, отражающих расчеты с качественным рендерингом изображений граничных поверхностей, изоповерхностей. Реализована возможность построения сечений, отражающих структуру расчетных сеток, изолиний, 2D – графиков, векторных полей. Реализована возможность расчета треков отмеченных частиц (маркеров) в контрольном объеме для выяснения топологии сложных течений. Для визуализации скалярных полей существует возможность выделения цветом отдельных значений анализируемых скалярных полей и наложения на полупрозрачные изоповерхности одних скалярных величин карт с цветовой индикацией локального распределения на этих поверхностях других скалярных величин. Использование маркеров жидких частиц дает возможность анимации движения потока жидкости с выделением локальных особенностей.

Несмотря на интенсивное развитие в различных классах задач подобных систем, они не могли решить многие проблемы, появлявшиеся по мере дальнейшего развития вычислительной техники и вычислительных методов. Надо отметить, что зачастую стремление реализовать все имеющиеся средства визуального представления результатов, приводило к информационной перегруженности образов. Это существенно понижало информационную ценность визуальных образов для исследователей и пользователей, несмотря на выдающуюся выразительность и привлекательность.

Появлявшиеся новые задачи, численные методы, алгоритмы и подходы, обусловленные развитием вычислительной базы, требовали новых подходов к визуализации данных. Однако эти новые требования не могли быть удовлетворены в рамках старых концепций визуального представления. Суть этой проблемы выражена в выступлении А.К.Алексеева (РКК «Энергия», г.Королев) [9]:

«…за бортом остается вопрос: что же именно нужно представлять визуально, что наиболее необходимо для практики?

Когда к нам приходят специалисты и предлагают комплексы программ для расчета поля течения и средства визуального представления этих полей, то общение напоминает визит фотографа в глухую деревню. Мы восхищаемся представленными полями скоростей, распределениями температуры, плотности, давления. Но на практике эти красивые изображения мы можем лишь повесить на стенку, чтобы любоваться. Для решения практических конструкторских задач нам недостаточно моделировать и визуализировать поле течения, нам недостаточно знать поле скоростей и распределения давления и т.д. На практике нас интересуют конкретные параметры для исследуемого класса задач, определяемого конструкторской целью. Нам необходимо визуальное представление именно этих параметров и прослеживание их изменений в зависимости от вариации тех или иных условий…Программы визуализации дают информацию, которая с точки зрения ее практического использования является с одной стороны сверхизбыточной, с другой стороны - недостаточной. Практическая задача содержит кроме модели течения еще некоторую цель и некоторые управляющие параметры. Практик не может напрямую использовать поле течения, ему нужны рекомендации на языке других специалистов (проектантов, конструкторов и т.д.) по выбору каких-то вариантов из множества допустимых. Это может быть выбор геометрической формы (минимальное сопротивление), выбор управления течением (максимальное смешение) и т.д. В задачах идентификации, по измерениям в поле выбираются коэффициенты, источниковые члены, граничные/начальные условия и т.д. Во многих случаях эти задачи можно поставить в оптимизационном виде. В этом случае решение газодинамической задачи является элементом решения более общей, оптимизационной задачи, содержащей еще и сопряженные параметры…»

Необходимы были новые визуальные концепции, позволявшие отойти от стандартных представлений о научной визуализации, как о графическом отображении наборов скалярных и векторных полей. Необходимы были новые подходы, которые могли бы в своем развитии

обеспечить реальные современные потребности научных исследований и инженерных технологий.

4. АНАЛИЗ СОВРЕМЕННЫХ КОНЦЕПЦИЙ И МЕТОДОВ ВИЗУАЛЬНОГО ПРЕДСТАВЛЕНИЯ ЧИСЛЕННЫХ ДАННЫХ

Основной задачей последнего десятилетия в развитии средств научной визуализации было преодоление зоны взаимонепонимания между разработчиками программных средств визуализации и потребителями их продуктов – исследователями и инженерами. Это взаимное непонимание имело свои корни в концепции, трактовавшей научную визуализацию как простое графическое отображение неких скалярных и векторных полей

Подобная трактовка блокировала дальнейшее развитие, как самой научной визуализации, так и всех научных дисциплин и областей, где применение методов визуального представления могло бы привести к принципиально новым результатам.

Преодоление рамок вышеизложенных представлений – старой концепции – шло по разным направлениям. Современные методы визуального представления численных данных можно с известной долей условности разделить на две группы:

- методы, основанные на повышении выразительности визуального образа. В дальнейшем изложении будем их называть – «Методы выразительности»;

- методы, основанные на анализе данных. В дальнейшем изложении будем их называть – «Методы анализа данных».

В отдельную группу необходимо выделить подходы, связанные с произошедшим за последнее десятилетие интенсивным развитием вычислительной техники. Это явление открывает новые возможности визуального представления численных данных, но порождает новые проблемы. Упомянутые подходы основаны на организации возможности применения уже разработанных ранее методов, алгоритмов, программных средств на новой вычислительной технике, реализации сочетания имеющихся методов с новыми вычислительными возможностями. Следует отметить, что эта группа подходов, как правило, не содержит в себе новых концепций и методов визуального представления, но стремится приспособить известные методы к новой технике.

Методы выразительности

Описываемая группа методов имеет одно общее свойство – они посвящены решению конкретных проблем, возникающих при определенных визуальных представлениях, с помощью усиления и повышения выразительности результирующего образа. Простой и убедительный прием повышения выразительности образа предложен в работе [10]. В данной работе рассматриваются проблемы визуализации течений жидкости с изменчивой пространственно-временной структурой в объеме. В качестве примера рассматривается задача свободной конвекции жидкости в полости с локальным теплообменом на горизонтальных поверхностях. Решение фундаментальной проблемы научной визуализации достигается выделением структурных элементов течения в сочетании с анимацией. «…При анализе течения в полости возникает проблема визуализации течения *как такового*, т.е. течения жидкости в которой ничего нет кроме ее самой. Основная трудность визуализации течения как такового связана с отсутствием у человека априорных образов движущейся в объеме жидкости… Мы видим движение жидкости только через движение каких-либо поверхностей, которые мы связываем с жидкостью…В рассматриваемом течении…характерной поверхностью, восприятие движения которой позволяет получить представление о движении жидкости в объеме, может служить изоповерхность нейтральной температуры. Для четкого выделения в кадре границы раздела областей, занятых холодной и теплой жидкостями, связь между интенсивностью цвета и температуры выбирается нелинейной. Использование нелинейной зависимости приводит к нарочитому заострению границы между областями, занятыми теплой и холодной жидкостями. Тем самым формируется искусственное представление о наличии в объеме двух жидкостей с резкой границей раздела…» [10].

Важным направлением среди методов визуального представления векторных полей является группа многочастичных анимационных методов [11] (далее МЧА-методы). Они основываются на визуальном представлении движения множества маркеров в объектном пространстве. В работе [12] предложении единый подход к рассмотрению и реализации МЧА-методов визуального представления векторных полей. Он заключается в следующем. Рассматривается расчетная область произвольной формы. Внутри области по некоторому правилу задается конечное множество точек, каждой из них ставится в соответствие маркер. Это – точки рождения маркеров. В данном подходе маркер определен своей точкой рождения, временем жизни и жизненным объемом. Большинство разнообразных МЧА-методов можно рассматривать как частные случаи вышеописанного подхода с варьирующимися правилами задания параметров маркера.

Другим важным направлением усиления выразительности визуальных образов являются текстурные методы, основанные на LIC-методе (Line Integral Convolution). Основы LIC-метода и его модификаций, предназначенных для визуального представления двумерных и трехмерных векторных полей подробно описаны в [13]. Выразительный и многофункциональный LIC - метод успешно применяется для рисования линий и трубок тока в трехмерной области. Данный подход помогает представить трехмерную глубину и ориентацию взаимоналагающихся линий тока. В сочетании с комбинациями толщины линий тока и цвета данный подход улучшает восприятие характеристик трехмерного течения. Использование текстур, ориентированных вдоль направления течения, совместно с их анимацией позволяет передавать достаточно полную информацию о течении [14].

Необходимо подчеркнуть тот факт, что, несмотря на внешнюю выразительность образов и на видимые преимущества в исследованиях тонких топологий численных решений, методы выразительности не нашли широкого применения в современных коммерческих программных пакетах визуализации. Причиной этому называют математическую сложность этих методов и высокие вычислительные затраты. Эти аргументы при нынешнем росте возможностей вычислительной техники не представляются существенными. Скорее всего, дело заключается в том, что сами по себе результаты визуального

представления, полученные при помощи методов выразительности, имеют невысокую информационную ценность, несмотря на внешнюю привлекательность. Сочетание с другими методами и подходами может существенно повысить информационную ценность подобных методов.

Методы анализа данных

Описываемая группа методов также имеет одно общее свойство – достижение новых результатов с помощью препроцессингового, параллельного расчету или дополнительного анализа основного поля данных. Как правило, такой подход позволяет применение известного и разработанного в других областях знания мощного математического аппарата. Суть новой концепции визуального представления можно сформулировать так: **получение основного поля данных для визуального представления; дополнительные анализ и обработка основного поля данных; получение новой информации, как результата дополнительного анализа и обработки; применение средств визуализации к полученной новой информации и к основному полю данных**.

Подобная концепция имеет разные формы и методы применения и реализуется в самых разных направлениях в настоящее время.

Простой, красивый и эффективный метод, основанный на анализе данных, был предложен в работе [15]. При визуальном представлении скалярного поля при помощи тоновой заливки задается закон, устанавливающий соответствие между цветами палитры и диапазоном изменения представляемой скалярной величины. При линейном задании этого закона можно потерять существенную часть информации о решении, особенно в областях сильных изменений величины в узких зонах. Во избежание этого проводится статистический анализ распределения значений изображаемой величины с целью определения нового закона соответствия, что помогает визуально представить полученное решение без потери ценной информации.

Другой способ выделения структур в решении, основанный на применении алгоритмов обработки изображений к полю данных, полученных в процессе решения задач математической физики, предложен в работе [16]. Цель данного подхода состоит в автоматизации обработки результатов проведенного численного моделирования газодинамического течения для получения структуры его сильных разрывов. В данном подходе газодинамические функции рассматриваются как интенсивность изображения, а значения функций в каждой точке как элементы изображения (пиксели). Применяется детектор перепадов и проводится классификация обнаруживаемого разрыва при помощи дискретных аналогов газодинамических соотношений, выполняемых на разрыве. Изложенный подход не зависит от конкретного вида решаемой задачи и не требует никакой априорной информации о течении. Он применим к результатам расчетов, полученных любым методом сквозного счета, ускоряет обработку результатов численного моделирования, а также повышает объективность интерпретации получаемых результатов.

Это направление получило новое дальнейшее современное развитие с применением мощного современного математического аппарата вейвлет-анализа. Работа [17] посвящена выделению и классификации сингулярностей газодинамических полей – ударных волн, слабых и контактных. Алгоритм [17] основан на разложении исходных газодинамических полей по вейвлетным базисам с последующим анализом полученных коэффициентов.

Одним из важных направлений исследования пространственно-временных структур в поле численного решения является анализ зон циркуляционного течения. Подобные ситуации возникают при обтекании потоком препятствий, при взаимодействии струй и во многих других случаях. В нестационарных задачах нужно прослеживать не только наличие циркуляционной (вихревой) зоны, но и отслеживать ее развитие во времени: процесс зарождения, развития и распада вихря. В работе [18] предложен подход обнаружения и прослеживания центра вихря в расчетной области, основанный на анализе собственных значений якобиана матрицы компонент скорости. Этот метод был успешно применен к ряду задач. В работе [19] этот метод был обобщен на случай вязких течений и применен к исследованию развития циркуляционных зон в задаче о взаимодействии вязкого потока со спутной струей. Трассировка центров вихрей в процессе моделирования течения позволяет следить за эволюционной картиной течения.

Подход, представленный в работе [20], предлагает проводить визуальное представление не только параметров основного поля результатов расчетов, но при решении сопряженных задач получать новую дополнительную информацию из визуального представления сопряженных параметров. В задачах управления течением, идентификации течения сопряженные параметры используются для расчета градиента целевого функционала. Визуализация сопряженных параметров в этих задачах может быть естественно использована для поиска зон наиболее эффективного контроля или для выбора зон измерения. Поле сопряженных параметров позволяет рассчитать и визуализировать вклады ошибки исходных данных, ошибки физической модели и погрешности дискретизации в погрешность целевого функционала. Это может использоваться для уменьшения погрешности и для получения решения с необходимой гарантированной точностью.

Еще одним важным направлением среди методов, базирующихся на анализе данных, является метод [21]. Данный метод рассматривает проблемы визуального представления определяющих параметров задач оптимизации, основывающихся на решении обратных задач. В общем, в практических целях обратные задачи формулируются следующим образом: найти при каких определяющих параметрах в классе задач будет возникать интересующее практического исследователя явление. Определяющие параметры варьируются до наступления события (физического эффекта) в процессе многократного расчета обратных задач. Результирующие взаимозависимости определяющих параметров представляются визуально в виде поверхностей, объемов, к которым можно применить геометрические преобразования. Рассматривая пространство определяющих параметров как набор базисных векторов, обратная задача формулируется как нахождение в этом пространстве всех подобластей, где событие наступило. Подобная постановка важна с точки зрения выработки

визуальной концепции для построения визуального представления решений обратных задач. Организация многократного расчета обратной задачи соответствует идеологии многопроцессорных параллельных расчетов. Данный методологический подход был успешно применен в работе [21] для оптимизации вычислительных свойств гибридных разностных схем (в качестве события рассматривалось возникновение нежелательных осцилляций).

Особым и крайне важным направлением является развитие систем визуального представления экспериментальных данных, которые помогают накапливать, верифицировать, сравнивать и представлять экспериментальные данные различных типов. Прототип подобной системы ExVis был описан в работе [22], где приведены основные принципы построения и функционального наполнения подобной системы, способной оперировать с большими объемами экспериментальных и вычислительных данных.

Подходы, обеспечивающие соответствие современной вычислительной технике

В отдельную группу, как было указано в начале данного раздела, необходимо выделить подходы к визуальному представлению данных и получению численных решений, основанные на использовании новых возможностей вычислительной техники. Типичным примером таких подходов можно назвать [23], где содержатся принципы и пример организации компьютерной виртуальной лаборатории на базе возможностей персональных компьютеров. Компьютерная лаборатория COMGA [23] включает исследование задач вынужденной, естественной и термокапиллярной конвекции несжимаемой жидкости на базе уравнений Навье-Стокса в приближении Буссинеска в областях простой формы. Визуализация данных играет важную роль на всех этапах решения задачи.

Другим важнейшим на сегодняшний день направлением являются работы по организации визуального представления численных данных с помощью высокопроизводительной вычислительной техники – параллельных вычислительных систем. В качестве примеров типичных разработок в этой области можно привести [24, 25]. Здесь основной задачей является организация отображения результатов с помощью традиционных методов визуализации при ограниченной пропускной способности каналов связи между рабочей станцией пользователя и суперкомпьютером для получения данных. Системы, описанные в вышеупомянутых работах, призваны разрешить эти проблемы путем организации оптимального потока данных между пользователем и суперкомпьютером.

5. ПЕРСПЕКТИВНЫЕ НАПРАВЛЕНИЯ РАЗВИТИЯ КОНЦЕПЦИЙ И МЕТОДОВ ВИЗУАЛЬНОГО ПРЕДСТАВЛЕНИЯ

По мере роста возможностей высокопроизводительных вычислений будут исследоваться все более сложные задачи и, следовательно, будут развиваться математические методы вычислений и математический аппарат в целом. Одним из необходимых условий дальнейшего развития научной визуализации станет **адаптация существующих концепций, методов и подходов визуального представления численных данных к применению на высокопроизводительной технике** – параллельных вычислительных системах. Здесь основная задача – сохранение наработанного опыта и перенос его на высокопроизводительные вычислительные системы.

Основным направлением развития концепций и методов визуального представления численных данных становится дальнейшая **разработка методов анализа данных, их расширение на другие прикладные области и синтез с методами выразительности.**

Важным направлением, имеющим в будущем большое значение, становится **развитие систем обучаемости – создание в классах задач баз экспериментальных и расчетных данных в численной и визуальной форме, позволяющих хранение, накопление, обработку и сравнение результатов в автоматизированном режиме**. Развитие подобных систем в разных областях знания и в разных классах задач позволит резко ускорить и качественно улучшить процессы верификации и валидации результатов, как вновь полученных, так и накопленных ранее.

Еще одним важным направлением развития систем визуализации станет **развитие систем обеспечения возможности анализа и контроля сложного физического или технологического процесса с разных ракурсов, разных величин в разные отрезки времени в автоматическом режиме**. Организация подобных систем, обеспечивающих, подобно автоматическим датчикам слежения, контроль расчетного процесса математического моделирования физического явления должна сыграть важную роль в научных исследованиях

В связи с этим становится очень важным и актуальным направлением **создание и развитие систем специализированных интерфейсов в задачах научной визуализации, позволяющих обработку и визуальное представление численных данных в режиме автоматического сценария**. Прообраз подобной системы описан в работе [26]. Использование таких систем позволит отлаживать наиболее нужный исследователю вид, метод, алгоритм визуального представления для объекта или явления в классе задач, создавать сценарий визуального представления и автоматически переносить этот сценарий на другой объект из этого класса задач.

6. ЗАКЛЮЧЕНИЕ

Научная визуализация становится катализатором интенсивной разработки новых открытий и технических решений в различных областях современной науки и техники. Системы научной визуализации помогут не только достойно представить результаты вычислений, но также объединять и анализировать результаты вычислений и экспериментов, накопленные ранее, легко и свободно оперировать огромными объемами информации. Для достижения этого на новом уровне развития научной визуализации предстоит решить такие глобальные задачи, как:

- выработка новых концепций и степени условности визуального представления физических процессов;
- разработка единых критериев качественной оценки визуального представления вычислительных и экспериментальных результатов;
- согласование алгоритмов визуального представления

численных решений с законами физических процессов.

7. ЛИТЕРАТУРА

[1] Альбом течений жидкости и газа. Под ред.М.Ван-Дайка. М., Мир, 1986, 182 с.

[2] Белозеров А.Ф. Оптические методы визуализации газовых потоков. Казань, Изд.КГТУ, 2007, 747 с.

[3] Баяковский Ю.М., Галактионов В.А., Михайлова Т.Н. « ГРАФОР. Графическое расширение Фортрана» М., Наука, 1985

[4] Математическое обеспечение графопостроителей. II уровень. СМОГ. Инструкция по программированию/ Под ред. Ю.А.Кузнецова,. Новосибирск: ВЦ СО АН СССР, 1976. 78 с..

[5] Стенограмма выступления С.В.Клименко (Институт физики высоких энергий, Протвино)// «Применение методов научной визуализации в прикладных задачах», Сб. науч.тр.,Москва, МГУ, 2000, с.4

[6] Могиленских Д.В., Павлов И.В., Федоров В.В., Мельникова С.Н., Сапожникова Е.Э. Принципы построения и функциональное содержание системы визуализации для анализа скалярных и векторных полей, заданных на двумерных регулярных сетках // Препринт РФЯЦ – ВНИИТФ, Снежинск, N 172, 2000, 26 стр.

[7] Бондарев А.Е., Бондарев Е.Н. Функции визуализации в вычислительной аэрогазодинамике// Общероссийский научно-технический журнал «Полет», Москва, «Машиностроение», N 10, 2000, с.53-60

[8] Горячев В.Д. Визуализация результатов расчетов в I&CS$_2$ – сетевой информационно-вычислительной системе моделирования термогидродинамических процессов // «Применение методов научной визуализации в прикладных задачах», Сб. науч.тр., Москва, МГУ, 2000, с.39-42.

[9] Стенограмма выступления А.К.Алексеева (РКК «Энергия», Королев) // «Применение методов научной визуализации в прикладных задачах», Сб. науч.тр.,Москва, МГУ, 2000, с.6-8.

[10] Гудзовский А.В., Клименко С.В. Визуализация свободно-конвективных течений жидкости в полости // Труды международной конференции по компьютерной графике ГРАФИКОН-1998, Москва, МГУ, 1998, с.31-38.

[11] Stolk J., J.J. van Wijk . Surface-Particles for 3D Flow Visualization, Advances in Scientific Visualization, Springer Verlag, 1992, pp.119-130.

[12] Сельвачев А.Ю., Аксенов А.А., Клименко С.В. Анимационная визуализация трехмерных векторных полей // Труды международной конференции по компьютерной графике ГРАФИКОН-1998, Москва, МГУ, 1998, с.53-56.

[13] Forsell L.K., Cohen S.D. Using Line Integral Convolution for Flow Visualization: Curvilinear Grids, Variable-Speed Animation and Unsteady Flows // IEEE Transactions on Visualization and Comp. Graphics, 1995, 1(2), pp.133-141.

[14] Anikanov A.A., Potiy O.A. Texture Advection for 3D Flow Visualization // The 13th International Conference on Computer Graphics: GraphiCon'2003, Moscow, Russia. Sept. 5-10, 2003, p.100-105.

[15] Mogilenskikh D.V. Nonlinear Color Interpretation of Physical Processes // Proc. 10th International Conference on Computer Graphics & Vision GRAPHICON'2000, Russia, Moscow, 2000, p.202-211.

[16] Базаров С.Б. Применение цифровой обработки изображений для визуализации результатов газодинамических расчетов // «Применение методов научной визуализации в прикладных задачах», Сб. науч.тр., Москва, МГУ, 2000, с.39-42.

[17] Афендиков А.Л., Левкович-Маслюк Л.И. , Луцкий А.Е. , Плёнкин А.В. Локализация разрывов в полях газодинамических функций с помощью вейвлет анализа // Математическое Моделирование, 2008, том 20, №7, страницы 65-84.

[18] D.Kenwright, R. Haimes « Automatic Vortex Core Detection», IEEE Computer Graphics and Applications, v.18, N 4, 1998, p.70 - 74.

[19] Бондарев А.Е., Бондарев Е.Н. О трассировке вихревых структур // «Научная визуализация в прикладных задачах», Сб. науч.тр., Москва, МГУ, 2003, с.4-13

[20] Алексеев А.К., Бондарев А.Е. Визуализация переноса погрешности при расчете поля течения // «Научная визуализация в прикладных задачах», Сб. науч.тр., Москва, МГУ, 2003, с.4-13.

[21] Бондарев А.Е. Применение методов визуализации для оптимизации гибридных разностных схем с учетом влияния вязкости и турбулентности // Тр.1-й межд. конференции «Трехмерная визуализация научной, технической и социальной реальности. Кластерные технологии моделирования»/ УдГУ – Ижевск, 2009. – Т.1, С. 120 – 123.

[22] Uselton S. «ExVis: Developing A Wind Tunnel Data Visualization Tool» Proc.IEEE Visualization 97, ACM press, New York, Oct.1997, pp.417-420.

[23] http://www.comga.ru

[24] Нестеров И.А. Интерактивная визуализация векторных полей на распределенных вычислительных системах // Математическое моделирование, 2008, т.20, N 6, сс. 3-14

[25] M. Iakobovski, I. Nesterov, P. Krinov. Large distributed datasets visualization software, progress and opportunities // Computer Graphics & Geometry, 2007, Vol. 9, N 2, pp. 1-19.

[26] Ma Kwan-Liu «Visualizing Visualizations» IEEE Computer Graphics and Applications, v.20, N 5, 2000, p.16 - 19.

Analyzing the Evolution of Scientific Visualization Concepts and Methods for Computational Physics

Abstract

This article describes the main stages of scientific visualization evolution as original field of science. The development perspectives for visual concepts, methods and approaches are considered for problems of computational physics (mainly CFD problems).

Keywords: *Scientific Visualization, Visual Concept, Visualization Methods*

Об авторах

Бондарев Александр Евгеньевич – к.ф.-м.н., старший научный сотрудник ИПМ им. М.В. Келдыша РАН
E-mail: bond@keldysh.ru

Чечеткин Валерий Михайлович – д.ф.-м.н., профессор, главный научный сотрудник ИПМ им. М.В. Келдыша РАН.
E-mail: chech@gin.keldysh.ru

Графические средства для научной визуализации в кристаллографии

А.Ю. Бреднихина[1], В.А. Дебелов[2]
[1]Новосибирский государственный университет,
[2]Институт вычислительной математики и математической геофизики СО РАН, Новосибирск, Россия
{brednikhina| debelov}@oapmg.sscc.ru

Аннотация

Данная работа посвящена обзору основных графических представлений, которые применяются в учебном процессе и исследовательской работе в области кристаллографии и кристаллооптики и входят в функциональность универсальных современных кристаллографических и минералогических Интернет-ресурсов. Обзор касается не вообще средств научной визуализации, а только тех базисных графических представлений данных, которые присущи в основном упомянутым дисциплинам. Это, как показано в данной работе, следующий набор: внешняя форма кристалла, индексы Миллера, центр/оси/плоскости симметрии, простые формы, стереоскопическая и гномостереоскопическая проекции, элементарная ячейка и кристаллическая решетка, оптические оси, индикатрисы поглощения, таблица Мишеля-Леви, ортоскопия и коноскопия и т.д. Рассматриваются только идеальные монокристаллы, поскольку все обучающие, справочные и информационные кристаллографические ресурсы в Интернет работают с графическими представлениями идеальных кристаллов. На основе анализа функциональности существующих графических представлений авторы предлагают подход к построению библиотеки графических средств для использования в областях кристаллографии и кристаллооптики (БКГС).

Ключевые слова: *Научная визуализация, интернет-ресурс, кристаллография, кристалл, графические представления, коноскопия, ортоскопия.*

1. ВВЕДЕНИЕ

Благодаря развитию вычислительной техники и систем коммуникации и передачи данных в научно-исследовательскую и образовательную сферу активно внедряются системы дистанционного обучения, образовательные интернет ресурсы и системы поддержки исследований. Это коснулось практически всех областей научной деятельности, в том числе и кристаллографии и наиболее связанной с ней минералогии.

Неотъемлемой частью информационных и образовательных интернет систем являются различные графические модули, предназначенные для иллюстрации данных. Первые графические средства в кристаллографических и минералогических системах появились более десяти лет назад в Mineralogy Database (http://mindat.org), SHAPE (http://shapesoftware.com), Интерактивной системе обучения (ИСО) "Кристалл" (http://ggd.nsu.ru/Crystal) и др. Сначала были достаточно простые приложения, предназначенные только для визуализации одиночных идеальных кристаллов или их отдельных характеристик и свойств. Постепенно эти и другие информационные системы расширяли набор графических средств за счет добавления новых модулей визуализации и расширения функциональности существующих приложений.

Несмотря на разнообразие современных графических средств представления данных о кристаллах, они в большинстве своем носят разрозненный характер: одни визуализирует только геометрию одиночного идеального кристалла (без дефектов решетки), другие – только его элементарную ячейку, третьи – некоторые физические характеристики; четвертые комбинируют вывод формы кристалла и оптических индикатрис и т.д.

Достаточно большая часть данных об одиночных кристаллах имеет геометрическую интерпретацию. Теоретически известна взаимосвязь между свойствами кристалла, что позволяет объединить различные данные о нем в единую модель и создать универсальное средство их графического представления, которое позволило бы визуализировать как отдельные характеристики, так и их комбинации. Такая комплексная модель до сих пор не была реализована ни в одной из доступных информационных систем, что говорит о необходимости анализа существующих моделей. Фактически речь идет о создании универсальной библиотеки графических представлений данных о кристаллах, объединенных в рамках обобщенной модели.

Цели данной работы – представить анализ и обобщение существующих графических представлений данных о кристаллах, имеющих геометрическую интерпретацию, и разработку обобщенной модели представления одиночного кристалла и его параметров. Для этого дается обзор существующих способов графического представления данных о кристаллах и приводятся: разработка модели, алгоритмов и программ комплексного представления и настраиваемой визуализации кристаллографических данных одиночных кристаллов.

2. СВЕДЕНИЯ ИЗ КРИСТАЛЛОГРАФИИ

Кристаллография – это наука, занимающаяся всесторонним изучением многообразия кристаллов, их свойств, условий образования и роста [1]. *Кристаллы* – это твердые тела, обладающие упорядоченной внутренней структурой на уровне атомов и молекул (обладающие трехмерно-периодической атомной структурой – *кристаллической решеткой*). Кристаллы в силу упорядоченности внутренней структуры обладают:

- *анизотропией* (неоднородностью свойств в зависимости от направления; например, зависимость скоростей роста граней от направления нормали к грани, или зависимость окраски от ракурса);
- *симметрией*. Под симметрией в данном случае понимается возможность совмещения фигуры с собой в результате преобразований симметрии: отражения,

вращения или их комбинаций. Оси и плоскости, относительно которых производятся преобразования симметрии, называются элементами симметрии. К ним относится и центр симметрии кристалла. Сочетания элементов симметрии и соответствующие им преобразования симметрии определяют группы симметрии или точечные группы [1, 2].

Для каждого кристалла определена *кристаллографическая система координат* (КСК), как правило, косоугольная. Начало координат – это центр симметрии кристалла.

Внешняя форма кристалла – здесь это некоторый многогранник. Грани кристаллов в кристаллографии описывают с помощью индексов Миллера [1], которые однозначно задают плоскости граней кристалла.

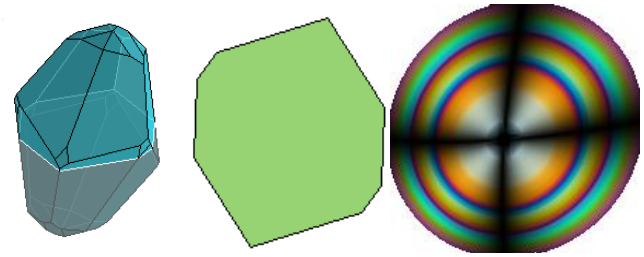
За счет свойства симметрии кристаллов можно выделить семейства граней, которые выводятся друг из друга преобразованиями симметрии. Такие семейства называют *простыми формами*. Для определения простой формы достаточно задать одного представителя. Для описания формы всего кристалла определяют группу симметрии и представителей простых форм, которые задаются с помощью индексов Миллера.

В рамках данной работы мы будем рассматривать только идеальные монокристаллы, поскольку все справочные и информационные ресурсы в интернет работают с графическими представлениями идеальных кристаллов.

Кристаллооптика – дисциплина, посвященная всестороннему изучению взаимодействия кристаллов со светом [3]. Большинство кристаллов являются оптически анизотропными, то есть по-разному взаимодействуют со светом в зависимости от направления, спектральных характеристик и поляризации падающего света. Для анизотропных кристаллов имеется понятие оптической оси. Бывают одноосные и двуосные кристаллы. Как правило, оптические оси показываются в привязке к граням, вершинам и центру симметрии кристалла. В наглядных пособиях наряду с оптическими осями обычно изображают оси индикатрисы в связке с осями кристаллографической системы координат.

Индикатрисы. Одним из основных объектов, с помощью которого описывают оптические свойства кристаллов, является индикатриса. Она представляет собой поверхность коэффициентов, например, преломления, в зависимости от направления падающего света.

Существуют два основных метода для изучения анизотропии кристаллов:



а) б) в)

Рис. 1. Изображения шлифа в поляризованном свете (а – положение среза, б – ортоскопия, в – коноскопия).

- *Ортоскопия*, когда на *шлиф* или *петрографический шлиф* (тонкая пластинка, вырезанная из кристалла или минерала, характерная толщина среза 0.01-0.3 мм) падает параллельный пучок поляризованного света, который затем проходит через второй поляризатор), после которого изображение регистрируется фотоаппаратом или глазом. Типичное изображение приведено на рис. 1б.
- *Коноскопия*. На шлиф падает конус поляризованного света. Пример изображения на рис. 1в.

Петрографический микроскоп. Один из основных инструментов изучения кристаллов на основе их оптических свойств [4]. Он отличается от обычного микроскопа наличием различных поляризационных устройств, за счет которых с его помощью можно наблюдать оптические эффекты, являющиеся следствием анизотропии физических свойств кристаллов. С его помощью изучают петрографические шлифы.

3. КРИСТАЛЛОГРАФИЧЕСКИЕ И МИНЕРАЛОГИЧЕСКИЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ПРОГРАММЫ

В сети Интернет существует множество ресурсов по кристаллографии и минералогии на различных языках. Наиболее представительные ресурсы, содержащие графические модули для визуализации кристаллографических данных, – это: минералогические базы данных [5, 6], Минералогический атлас [7], Минералогическая информационная система [8], Афинская минералогическая система [9], Кристаллографическая и кристаллохимическая база данных для минералов и их структурных аналогов [10], русскоязычный ресурс: веб-справочник или информационная справочная обучающая система "Кристалл" [11, 12]. Как правило (или по максимуму), для каждого минерала имеется описание его характеристик: классификация, залегание минерала (глубина, условия); физические свойства (прочность, прозрачность, окрас, плотность, расщепление); кристаллографические параметры (группа симметрии, параметры кристаллографической системы координат, законы образования двойников); параметры химической структуры, элементарной ячейки; оптические свойства (изотропный или анизотропный); месторождения (география), и другие.

С точки зрения графических средств минералогические ИС очень разнородны. Имеется ряд программных продуктов, которые устанавливаются на платформе конечного пользователя и служат более узким целям – не минералогии вообще, а именно кристаллографии и кристаллооптике. Наиболее продвинутые системы: Shape 7.2 [13], KrystalShaper [14], JCrystal [15], Crystal Shape Laboratory [16], Java Crystal Gallery [17], Smorf [18].

И наконец, нами были рассмотрены геометрические интерпретации и графические представления, применяемые в работах и учебниках по кристаллографии (на примере [1, 3]), и которые не используются в рассмотренных ресурсах. Ряд из них, которые вполне алгоритмизуемы и реализуемы с учетом современного развития вычислительной техники, также было решено включить в функциональность проектируемой библиотеки БКГС. Например, сферические проекции кристаллов. Конечно, за пределами библиотеки были оставлены многие графические представления кристаллов,

например, фотореалистические изображения. Это связано с тем, что решение задачи физически корректной реалистической визуализации анизотропных сред и объектов в настоящее время относится к области фундаментальных исследований [19].

4. ОБЗОР ГРАФИЧЕСКИХ ПРЕДСТАВЛЕНИЙ

Что мы понимаем под наличием функции графического представления, например, формы кристалла? Можно сказать, что OpenGL, DirectX, Java3D и даже GDI (Graphical Device Interface в Windows) позволяют рисовать любые трехмерные формы. Но они понимают эти формы лишь тогда, когда они заданы в виде набора граней. Это очень низкий уровень языка описания объекта для проблемной области, и очевидно, что кристаллограф не должен сам программировать все необходимые трехмерные представления кристалла вместе с аннотирующими элементами. Да и не сможет он вмешиваться внутрь кода программы. Наша проектируемая библиотека графических функций БКГС должна включать геометрически и визуально довольно сложные примитивы, параметризованные на основе кристаллографических понятий.

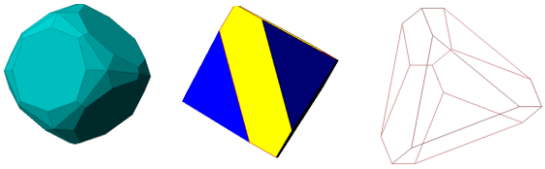


Рис. 2. Визуализация внешней формы кристалла

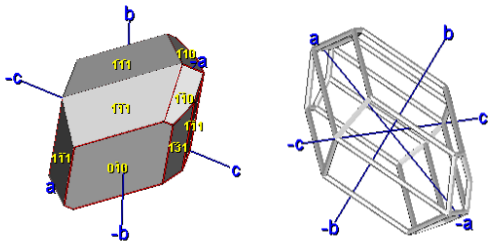


Рис. 3. Визуализация формы кристалла с осями КСК и индексами Миллера у граней

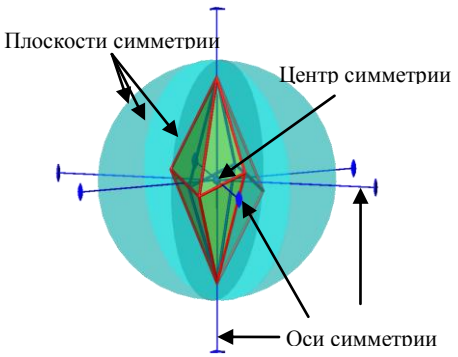


Рис. 4. Визуализация формы одиночного идеального кристалла и элементов симметрии

Необходимо отметить, что многие средства работают в интерактивном режиме, когда пользователь оперативно (возможно при помощи мыши) меняет параметры визуализации исследуемой модели, например, ракурс.

4.1 Исходные данные

Исходный набор данных о конкретном кристалле является вполне обозримым. В качестве примера возьмем данные по кристаллу кальцита: длины векторов репера КСК, углы между осями КСК, группа симметрии, индексы Миллера простых форм, индикатриса, заданная при помощи главных коэффициентов преломления и ортов осей эллипсоида. К этим данным следует добавить также cif-модель [20], из которой понадобятся координаты расположения атомов в элементарной ячейке и связи между ними.

4.2 Графические представления в кристаллографии

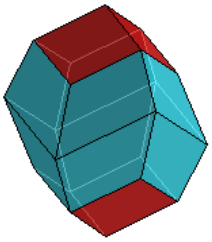


Рис. 5. Динамическая подсветка граней, разным цветом изображены разные простые формы (светлые линии – ребра невидимых граней)

Замечание. Этот небольшой экскурс в реально используемые графические представления наталкивает на мысль, что формировались они эволюционно, т.е. добавлялись в функциональность программы эпизодически, по мере необходимости. Мощность множества графических средств у разных систем сильно различается. Рассматривая используемые изображения, можно сделать замечание об их реализации. Видно, что проектировались конкретные изображения, например, внешняя форма с осями КСК, с индексами Миллера (рис. 3). Разрабатывался и программировался конкретный алгоритм визуализации такого представления с учетом возможного удаления невидимых элементов. Как правило, такие реализации почти не поддаются модификациям, т.е. если разработчик задумал ввести еще какое-либо представление, ему придется провести для него всю разработку целиком.

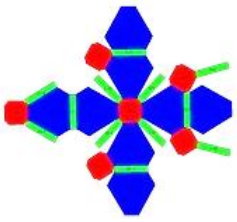
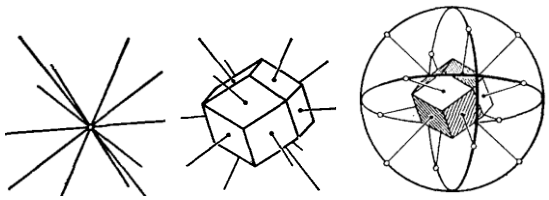


Рис. 6. Плоская развертка формы кристалла

Наряду со стандартным представлением формы кристаллов в виде многогранников, используются также различные кристаллографические проекции: сферические, стереографические, гномостереографические и гномонические проекции, см. [1]. Большинство этих проекций строятся с помощью полярного комплекса – набора векторов единичной длины, выходящих из точки центра симметрии и совпадающих с направлениями векторов

нормалей к граням кристалла (рис. 7а). Точки – концы этих векторов на сфере образуют *сферическую проекцию*. В принципе можно получить сферическую проекцию всего многогранника, а не только векторов нормалей к граням.



а) б) в)

Рис. 7. Графическое представление ромбического додекаэдра: а – его полярный комплекс, б – изображение совместно с нормалями, в – изображение совместно с нормалями и сферической проекцией

Сферическая проекция кристалла (и любой другой поверхности) хотя и наглядна, но трудна в ручном исполнении, поэтому кристаллографы традиционно применяют ее проекцию на плоскость и используют стереографическую или гномостереографическую проекцию.

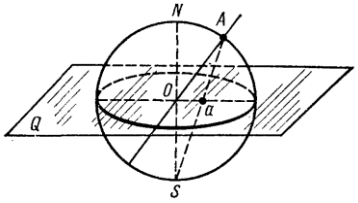


Рис. 8. Построение стереографической проекции на экваториальную плоскость [1]

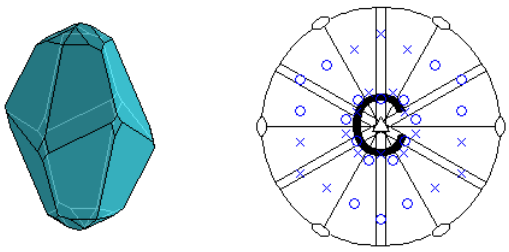


Рис. 9. Слева – внешняя форма кристалла, справа – гномостереографическая проекция с элементами симметрии

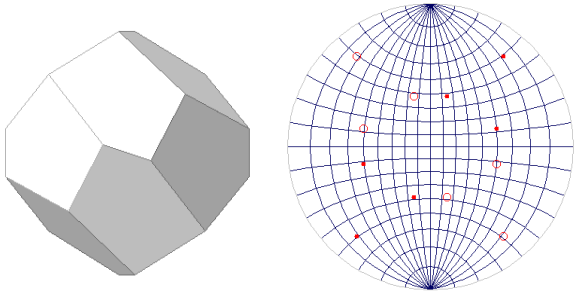


Рис. 10. Слева – внешняя форма кристалла, справа – сетка Вульфа

Стереографическая проекция получается из сферической проекции с помощью отображения на некоторую плоскость. Наиболее распространенной в кристаллографических интернет ресурсах является *гномостереографическая проекция* (часто ее называют просто также стереографической). Она представляет собой стереографическую проекцию полярного комплекса кристалла на экваториальную плоскость. Причем на чертеже часто отображаются не только нормали к граням, но и специальные образы элементов симметрии. В *гномонических проекциях* в качестве проекционной плоскости используют не экваториальную, а касательную к сфере плоскость.

В качестве дополнения к (гномо-)стереографическим проекциям кристаллов для решения различных практических задач используют сетку Вульфа, которая является стереографической проекцией системы параллелей и меридианов, нанесенных на поверхность сферы (в качестве проекционной плоскости выбирается плоскость одного из меридианов). С ее помощью можно как построить стереографическую проекцию, так и восстановить по ней сферические координаты точки или измерить угловое расстояние между точками и т.д.

Гномостереографические проекции кристаллов (иногда вместе с сетками Вульфа, иногда без) реализованы во многих минералогических и кристаллографических системах (рис. 10), и включение алгоритмов их расчета и визуализации в набор средств проектируемой библиотеки БКГС является очевидным. Несмотря на простоту и наглядность, сферические проекции не представлены в минералогических и кристаллографических интернет ресурсах. Но поскольку они являются одним из этапов получения стереографической и гномостереографической проекций, их реализация в библиотеке становится логичной.

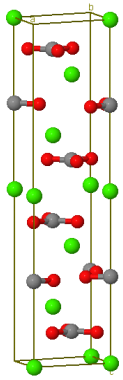


Рис. 11. Элементарная ячейка кальцита – форма, атомы и связи (из [7])

Важной особенностью кристаллов является кристаллическая решетка. Она определяет все свойства, которые выделяют их в обособленный класс объектов. Кристаллохимия – это специальный раздел кристаллографии, занимающийся изучением структуры и свойств кристаллической решетки. Одно из основных понятий кристаллохимии – это *элементарная ячейка*. Она представляет собой наименьший по объему параллелепипед, внутри и в узлах которого располагаются атомы или молекулы, и повторение которого в пространстве позволяет восстановить структуру кристаллической решетки [1, 2]. Элементарная ячейка – это своего рода инвариант и точка привязки, поскольку она выбирается так, чтобы ее симметрия совпадала с симметрией кристалла в целом.

Элементарная ячейка описывает минимальный блок химической структуры кристаллической решетки, и ее часто изображают в виде отдельного параллелепипеда, внутри которого располагаются атомы-шарики со связями или их координационные многогранники (см. рис. 11). Полезная связка микроскопических (элементарная ячейка) и макроскопических (внешняя форма) параметров продемонстрирована на рис. 12, где видно, что симметрия элементарной ячейки и кристалла как макроскопического объекта совпадают, элементарные ячейки представляются в виде сплошных параллелепипедов.

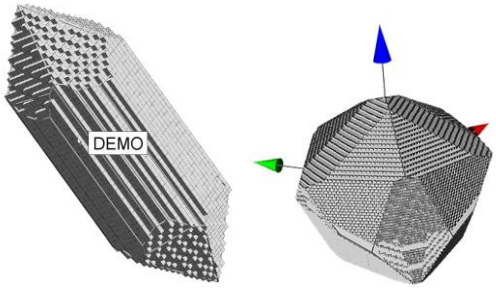


Рис. 12. Блочная застройка формы кристалла: слева – KrystalShaper, справа – разрабатываемая библиотека

4.3 Петрографический микроскоп и оптические свойства кристаллов

Петрографический микроскоп – это сложное устройство, и для эффективной работы с ним необходим большой опыт: специалист должен по интерференционной картине и форме сечения уметь определять кристалл, понимать, как расположена оптическая ось по отношению к сечению и многое другое. В результате возникает целый набор объектов, которые формируют компьютерную модель петрографического микроскопа: оптические оси, плоскость сечения кристалла (шлифа), индикатриса и ее расположение по отношению к сечению, таблица Мишеля-Леви. В кристаллографии сформировались своего рода стандарты графического представления многих объектов и их комбинаций, в том числе и перечисленных выше, что должно быть учтено при разработке графической библиотеки.

4.3.1 Анизотропия оптических свойств и оптические индикатрисы

Большинство кристаллов является оптически анизотропными. Это означает, что они по-разному взаимодействуют со светом в зависимости от направления падающего света и его характеристик. К наиболее ярким проявлениям анизотропии (см. [19]) можно отнести двулучевое преломление (луч падающего света делится на два луча внутри кристалла), плеохроизм (зависимость коэффициентов поглощения от направления и поляризации падающего света).

Поскольку библиотека БКГС реализует комплексное представление кристалла, в ней должна быть представлена связь формы кристалла с ориентацией оптической индикатрисы и оптических осей, что может быть очень полезным при изучении влияния положения индикатрисы на изображения, наблюдаемые в микроскопе.

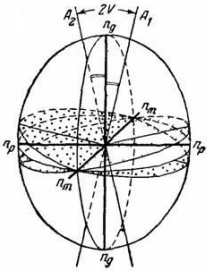


Рис. 13. Индикатриса преломления – классическое кристаллографическое представление в литературе

4.3.2 Таблица Мишеля-Леви

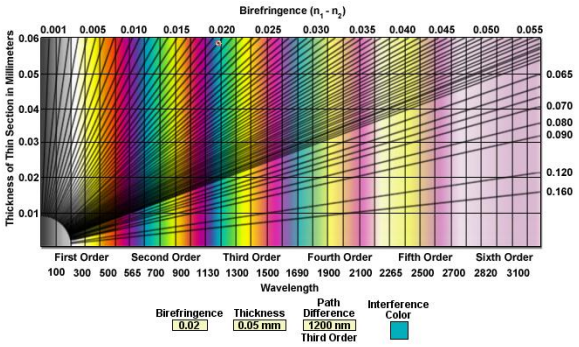


Рис. 14. Таблица Мишеля-Леви [21]

В таблице Мишеля-Леви последовательно отображаются переходы интерференционных цветов, наблюдаемые в петрографическом микроскопе при изменении толщины среза. Она особенно активно используется на этапе начального обучения, когда студенту необходимо накопить достаточный опыт определения цветов и запомнить зависимости цвета от толщины среза для кристаллов различных минералов, что делает ее неотъемлемой частью компьютерной модели петрографического микроскопа.

5. АНАЛИЗ И ОБОБЩЕНИЕ

Таким образом, рассмотрев функциональность всего множества средств научной визуализации в области кристаллографии и минералогии, мы пришли к выводу, что, следуя [22], они формировались по эволюционному пути, т.е. новые возможности (здесь – новые графические представления) добавлялись в пользовательский интерфейс и функциональность программы эпизодически, по мере необходимости и в зависимости от мощности доступной вычислительной техники, графического оборудования и графических библиотек. Также в [22] отмечалось, что на определенном этапе развития продукта необходимо делать революционный скачок, т.е. целиком пересматривать принципы формирования структуры программного продукта, включая его пользовательский интерфейс.

Отметим, что нельзя проследить наличие какой-либо общей модели, лежащей в основе всех рассмотренных графических представлений. Аналогично и в нашем коллективе присутствовал указанный эволюционно-эпизодический подход в течение более чем 10-летней разработки графической функциональности информационной справочной системы (ИСО) "Кристалл". К настоящему времени этот интернет ресурс предоставляет достаточно

богатый набор средств, а цель данной работы – сделать революционный скачок в его развитии: мы поставили задачу создать единую модель для научной визуализации в рассматриваемой области кристаллографии.

5.1 Геометрическое представление

Геометрическая модель кристаллографических понятий строится на наборе геометрических конструктивных элементов для понятий и объектов, введенных в п.2. Мы предлагаем следующий подход к визуализации данных о кристалле:

1. Построение геометрической метамодели кристалла на основе исходных данных конкретного кристалла.
2. Построение геометрического представления кристалла (ГПК).
3. Задание геометрического представления элементов ГПК.
4. Спецификация материалов (цвет, текстура и т.п., см. [23]) и типа отображения элементов, т.е. уточнение графического представления элементов ГПК.
5. Определение способа проецирования и выбор камеры.
6. Добавление специфических графических элементов.

5.1.1 Геометрическая метамодель кристалла

Геометрическая метамодель кристалла – это набор геометрических примитивов для конкретного кристалла в трехмерном пространстве. Данная метамодель рассматривается в правосторонней декартовой системе координат (мировой СК) XYZ, начало которой находится в центре симметрии кристалла. Элементы метамодели:

- Центр симметрии – точка.
- Оси мировой СК XYZ.
- Оси кристаллографической СК ABC.
- Описанная сфера – это сфера с центром в начале координат, описанная вокруг многогранника кристалла. Радиус **R** этой сферы служит единицей измерения для задания различных размеров и положений.
- Внешняя форма – набор простых форм.
- Простая форма – набор граней, соответствующих одному индексу Миллера (ИМ) в описании кристалла. Все простые формы индексируются порядковым номером ИМ в описании.
- Грань – пространственный многоугольник, специфицированный ИМ.
- Каркас – реберное представление формы кристалла.
- Оси симметрии.
- Плоскости симметрии.
- Оптические оси O1 и O2, если есть.
- Нормали граней.
- Полярный комплекс – набор всех нормалей граней, выходящих из одной точки.
- Вектора скоростей роста граней в идеальных условиях.Элементарная ячейка (ЭЯ). Размеры ЭЯ.
- Атомы ЭЯ. Координаты относительно ячейки.
- Связи между атомами.
- Сфера для сферической проекции. Как правило, описанная сфера.
- Сетка Вульфа. Семейство параллелей и меридианов на сфере.
- Индикатриса преломления.
- Индикатриса поглощения.
- Плоскость сечения индикатрисы.
- Круговые сечения индикатрисы, очень важны в случае двуосных кристаллов. Нормали к круговым сечениям индикатрисы совпадают с направлениями оптических осей.
- Плоскость для стереографической проекции. Порт изображения стереографической проекции. Порт – это прямоугольник в пространстве (в мировой СК), на котором размещается требуемое изображение.
- Плоскость для гномостереографической проекции. Порт изображения гномостереографической проекции.
- Плоскость развертки.
- Плоскость шлифа. Порт изображения шлифа.
- Порт изображения ортоскопии шлифа.
- Порт изображения коноскопии шлифа.
- Порт таблицы Мишеля-Леви.

Отметим, что все эти элементы показывают максимально возможный набор геометрических элементов, которые пользователь может использовать в своем представлении кристалла. Часть из них уже может быть рассчитана на основе исходных данных, другие просто резервируются, поскольку требуют дополнительных спецификаций пользователя, например, выбор конкретной плоскости шлифа.

5.1.2 Промежуточный этап построения геометрического представления кристалла

На данном этапе из элементов метамодели строится вспомогательная геометрическая конструкция (ВГК). Те элементы, которые включаются, привязываются к требуемым (задаваемым пользователем) местам в пространстве. При этом один элемент может входить в конструкцию несколько раз. Например, нормаль грани может быть привязана к началу координат, к центру грани, к другому заданному пользователем месту. Оси могут быть привязаны к началу координат или к центру какой-либо грани, или к вершине и т.д. Другими словами, экземпляр оси привязывается к тому месту, к которому приковано внимание пользователя. Ограничений нет, поскольку предполагается, что пользователем управляет здравый смысл. Очевидно, что грани кристалла располагаются единственным образом. Для ЭЯ также возможны варианты: одна, несколько, блочная застройка внешней формы.

На этом этапе все пользовательские параметры должны быть определены.

5.1.3 Уточнение геометрического представления кристалла

Теперь для всех элементов конструкции ВГК специфицируется их геометрическая форма и размеры. Всем элементам одного типа может быть назначена единая форма либо каждому своя. Например, ось A будет представлена отрезком, а ось B в виде телесной стрелки. Напомним, что все размеры назначаются в процентах от радиуса **R**. По окончанию данного этапа мы должны получить описание

всей геометрии пространственной сцены, которую в дальнейшем будет наблюдать пользователь.

Вершины и точки – точки, сферы.

Векторы, оси – отрезок, цилиндр, цилиндр с конусом (телесная стрелка).

Грань – ребра границы, сплошная грань, оба типа вместе. Можно специфицировать показ индекса Миллера – надо установить признак ИМ.

Ребра – это отрезки или цилиндры.

Элементарная ячейка. Поскольку это параллелепипед, то его грани и ребра можно специфицировать аналогично граням и ребрам кристалла. Именно на данном этапе в случае блочной застройки определяются все параллелепипеды, входящие в застройку. Атомы – вершины, связи – ребра.

Ось симметрии, оптическая ось – отрезок, вектор, цилиндр, телесная стрелка.

Плоскость симметрии – правильный многоугольник, лежащий в этой плоскости – грань.

Аналогичным образом специфицируются и остальные элементы ВГК, что мы опустим за недостатком места.

Таким образом, ГМК (геометрическая модель кристалла) сформирована полностью. Это 3D сцена.

5.1.4 Завершение задания сцены и рендеринг

Материалы мы понимаем здесь даже более широко, чем в [23], не только цвет и гладкость, но и типы и толщины линий.

- Отрезок специфицируется цветом, прозрачностью, типом линии, толщиной линии.
- Сплошные поверхности специфицируются цветами внешней и внутренней поверхности, прозрачностью и т.д. Эти спецификации полностью определяются возможностями используемой графической библиотеки: OpenGL, DirectX, Java3D и т.д.

Завершающий этап формирования сцены – это спецификация параметров камеры: положение в мировой СК, точка внимания, "вектор вверх", порт вывода всей сцены, см. [23].

Особенности рендеринга. При наличии в сцене граней с признаком ИМ осуществляется вывод индексов Миллера этих граней.

6. ЗАКЛЮЧЕНИЕ

При формировании функциональности нашей библиотеки БКГС мы выбрали подход, основанный на понятиях и объектах, введенных в п. 2. В данном случае это формирование внешней формы со всеми элементами, которые представимы геометрически, и дальнейшая совместная визуализация формы кристалла, кристаллографических координатных осей, оптических осей, индикатрис, элементов симметрии, аннотация граней кристалла индексами Миллера. Ни одна из известных систем не реализует эти средства в полном объеме. А вот задачи собственно визуализации и удаления невидимых элементов должны решаться универсальной графической библиотекой – OpenGL и др.

Конечно, пользователь может собрать какой-то набор из представленных форм и визуализировать его, например, получить изображение, приведенное на рис. 15. Здесь трудно что-либо разобрать, но мы не ограничиваем общности. Ведь и на языке программирования можно написать синтаксически правильную, но практически бессмысленную программу.

Сложность задания определяется сложностью желаемой геометрической модели. Например, для получения коноскопии шлифа (см. рис. 1) достаточно: а) выбрать плоскость шлифа; б) порт изображения коноскопии шлифа расположить на плоскости XY; в) камеру выбрать в одном из стандартных положений. Для получения развертки (рис. 6): а) грани сделать сплошными; б) ребра представить отрезками; в) задать цвета простых форм; г) задать порт на плоскости XY; д) камеру выбрать в одном из стандартных положений.

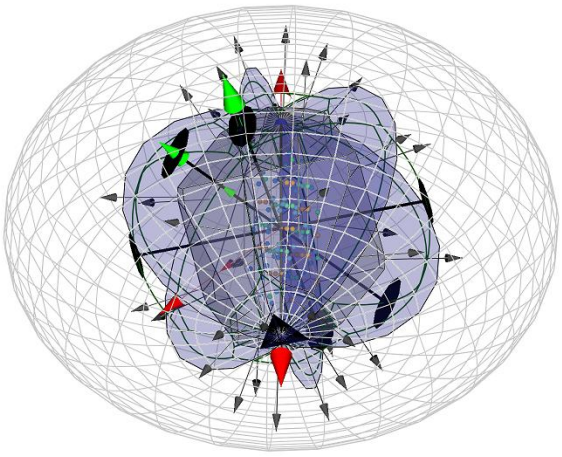


Рис. 15. Совместная визуализация внешней формы кристалла и ряда характеристик

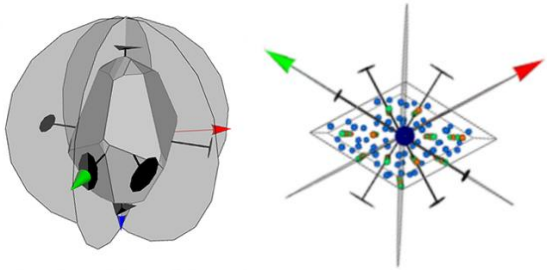


Рис. 16. Визуализация внешней формы (слева) и элементарной ячейки (справа) кристалла кальцита и его элементов симметрии

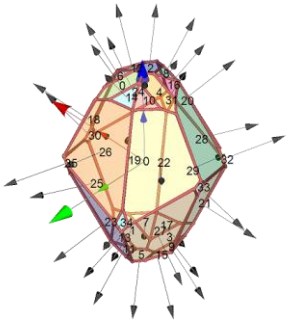


Рис. 17. Внешняя форма кристалла кальцита с кристаллографической системой координат и нормалями к граням и их индексами

Отметим, что все эти параметры могут меняться пользователем в интерактивном режиме, но, как правило, настройка делается один раз и сохраняется между сеансами.

Мы утверждаем, что множество всех изображений кристалла, построенных на основе предложенной выше геометрической модели, мощнее, чем множество графических представлений, используемых в известных минералогических ИС и кристаллографических ресурсах.

Благодарности

Работа выполнена при частичной финансовой поддержке РФФИ, грант № 06-07-89216а, 09-07-00237а.

7. ЛИТЕРАТУРА

[1] М.П. Шаскольская. Кристаллография / М.: Наука – 1984.

[2] Г. Б. Бокий. Кристаллохимия / М.: Наука – 1971.

[3] А.В. Шубников. Оптическая кристаллография // М: Изд-во АН СССР – 1950.

[4] В.А. Дебелов, А.Ю. Рубцова, С.З. Смирнов. Компьютерная модель петрографического микроскопа // Тр. 16-й международной конференции по компьютерной графике и ее приложениям Графикон-2006, Новосибирск, 1-5 июля 2006. – Новосибирск: ИВМиМГ СО РАН, 2006. – С. 298–302.

[5] Минералогический справочник Д.Ральфа http://mindat.org

[6] Минералогическая база данных http://webmineral.com/

[7] Минералогический атлас http://www.mineralienatlas.de

[8] Минералогическая информационная система http://www.minmax.net/

[9] Афинская минералогическая система http://un2sg4.unige.ch/athena/html/athome.html

[10] Кристаллографическая и кристаллохимическая база данных для минералов и их структурных аналогов http://database.iem.ac.ru/mincryst/

[11] ИСО "Кристалл" http://ggd.nsu.ru/Crystal

[12] Debelov V.A., Devyatova A.Yu., Sattarov M.A, Smirnov S.Z., Zhmulevskaya D.R. Technological Aspects of Development of the Web-trainer on Crystallography // Тр. 13-й международной конференции по компьютерной графике и зрению Графикон-2003, Москва, 5-10 сент. 2003. – М.: МГУ, 2003. – С. 192–195.

[13] Shape 7.2 http://shapesoftware.com

[14] Krystal Shaper http://www.jcrystal.com/products/krystalshaper/

[15] JCrystal http://www.jcrystal.com

[16] Crystal Shape Laboratory http://www.crystalgrowing.com/crystal-lab/crystal-lab.htm

[17] Java Crystal Gallery http://www.kristallzuechtung.de/galerie.htm

[18] Smorf http://www.smorf.nl/

[19] Дебелов В.А., Саттаров М.А. Проблемы реалистической визуализации кристаллов // Тр. 13-й международной конференции по компьютерной графике и зрению Графикон-2003, Москва, 5-10 сент. 2003. – М.: МГУ, 2003. – С. 221–227.

[20] Cif-модель. http://www.iucr.org/resources/cif/

[21] Michel-Levy Birefringence Chart http://www.microscopy.fsu.edu/primer/java/polarizedlight/michellevylarge/index.html

[22] Hartson H.R., Hix D. Human-Computer Interface Development. Concepts and Systems for its Management // ACM Computing Surveys. – 1989. – Vol. 21, № 1. – P. 5–92.

[23] Foley J., Van Dam A., a.o. Computer Graphics Principles and Practice / 2nd Edition in C. – Addison Wesley. – 1990.

Об авторах

Анна Юрьевна Бреднихина – аспирантка Новосибирского государственного университета.

E-mail: brednikhina@oapmg.sscc.ru.

Виктор Алексеевич Дебелов – ведущий научный сотрудник Института вычислительной математики и математической геофизики СО РАН, профессор кафедры компьютерных систем факультета информационных технологий НГУ.

E-mail: debelov@oapmg.sscc.ru.

Graphics tools for crystallography

Abstract

The report is devoted to a survey of graphical representations that are used in education and researches in the areas of crystallography and crystallooptics and included in functionality of widespread used modern Internet-resources on crystallography and mineralogy. The given survey concerns to those basic graphical representations of data which are intrinsic mainly in mentioned scientific areas. This is, as is shown in the given work, the following set: an external shape of a crystal, Miller's indices and forms, center/axes/planes/groups of symmetry, a stereoscopic projection, an elementary cell, optical axes, indicatrices, ortoscopic and conoscopic pictures of thin sections, etc. Only ideal monocrystals are considered as all Internet-resources produce graphical representation of data of ideal monocrystals. Basing on analysis of existent graphical representations authors suggest and justify their approach to construction of universal library of graphics tools that should be used in the areas of crystallography and crystallooptics.

Keywords: *Scientific visualization, crystallography, monocrystal, graphical representation, orthoscopy, conoscopy.*

About the authors

Anna Yu. Brednikhina is a Ph.D. student of the Novosibirsk State University. Her contact email is brednikhina@oapmg.sscc.ru.

Victor A. Debelov holds a position of leading researcher at Computer Graphics Lab. of the Institute of Computational Mathematics and Mathematical Geophysics SB RAS, prof. of Novosibirsk State University. His contact email is debelov@oapmg.sscc.ru.

Алгоритм векторизации штриховых изображений отрезками прямых

Максим Стержанов
Кафедра информатики
Белорусский государственный университет информатики и радиоэлектроники, Минск, Республика Беларусь
accept@bk.ru

Аннотация

В статье описывается гибридная методика векторизации штриховых бинарных изображений отрезками прямых, основанная на представлении объектов изображения в виде планарного нагруженного ориентированного псевдографа. Предлагается алгоритм построения графовой модели изображения, кодированного в виде концов серий. Метод является дальнейшим развитием идей Ди Зензо, Монагана, Павлидиса.

Ключевые слова: *Векторизация, RLE, графовая модель.*

1. ВВЕДЕНИЕ

При сканировании широкоформатных чертежей получается растровое изображение большого размера, что накладывает определенные ограничения на применяемые алгоритмы обработки. В частности, необходима структура данных, которая будет обеспечивать компактное хранение изображения, сохраняя его топологию. Рассмотрим сильные и слабые стороны некоторых методов представления растровой информации. Операция утоньшения позволяет представить объекты на растре линиями единичной ширины. Скелетизированное изображение сохраняет топологию, однако оно чувствительно к шуму, места соединений обрабатываются не всегда корректно. Недостатком контурного препарата является то, что по нему трудно построить топологию исходного изображения. Граф смежности строк является удобным способом представления изображения, состоящего из большого числа горизонтальных или вертикальных отрезков. Однако данная структура не хранит информацию о местах соединения, что, безусловно, является серьезным недостатком.

В данной работе предлагается представление каждого объекта изображения в виде планарного нагруженного ориентированного псевдографа, в котором все ребра суть прямолинейные отрезки или дуги плоских кривых, а вершины – точки на плоскости, являющиеся концами отрезков или точками сочленения нескольких отрезков.

2. ИСПОЛЬЗУЕМАЯ ТЕРМИНОЛОГИЯ

Под серией будем понимать последовательность отсчетов, имеющих одинаковое значение[1]. Серия однозначно определяется с помощью четырех значений:

d – направление (вертикальное или горизонтальное);

pos – номер столбца (строки) матрицы изображения, которому принадлежит серия;

beg – номер строки (столбца) матрицы изображения, которому принадлежит первый пиксель серии;

end - номер строки (столбца) матрицы изображения, которому принадлежит последний пиксель серии.

Две серии A и B называются смежными, если выполняются условия (1)-(4):

$$A.d = B.d \quad (1)$$

$$|A.pos - B.pos = 1| \quad (2)$$

$$A.beg \le B.end + 1 \quad (3)$$

$$A.end \ge B.beg - 1 \quad (4)$$

Под путем из серии A к серии B будем понимать последовательность серий $A=A_1, A_2,\ldots, A_n=B$, таких, что A_i является смежной с A_{i+1} для $1 \le i \le n-1$.

Рассмотрим две смежные серии. Они находятся в отношении «родитель-потомок». Серию с меньшим значением pos будем называть родительской. Серию с большим значением pos будем считать дочерней.

Серия будет называться нормальной, если она имеет ровно одного родителя и ровно одного потомка. В противном случае серия является особой.

Под начальной будем понимать серию, не имеющую родителей. Под конечной будем понимать серию, не имеющую потомков. Под серией слияния будем понимать серию, имеющую более одного родителя. Под серией ветвления будем понимать серию, имеющую более одного потомка.

Под полосой будем понимать связное множество нормальных серий. Полной полосой будем называть полосу, которая не является подмножеством другой полосы. Полоса представляет отдельную «ветвь» изображения. Полоса может содержать строго одну серию из столбца изображения, т.е. в полосе отсутствуют случаи ветвления и слияния серий. Под длиной полосы будем понимать количество серий, которые образуют полосу. Под весом полосы будем понимать суммарное число отсчетов ее серий. Если квадрат длины полосы больше ее веса, то полоса закрывается.

Листинг первым ввел понятие линейного скелета[2], который образуется в результате континуального сжатия области (без изменения топологии) с выделением подмножества пикселей единичной толщины. Средняя ось формируется центрами дисков максимального радиуса, помещенных внутрь области[3].

Скелетной кривой (СКР) в непрерывном пространстве будет являться либо линейный скелет либо средняя ось, сохраняющие топологические или геометрические признаки[4]. Мы задаем СКР множеством из N целочисленных точек $pnt_0, pnt_1,\ldots, pnt_{N-1}$. СКР имеет

характеристику ширины. На атрибуты СКР задаются следующие ограничения:

$$N \geq 3 \qquad (5)$$

$$| pnt.x_{i+1} - pnt.x_i | \leq 1, i = (0,1,...,N-2) \qquad (6)$$

$$| pnt.y_{i+2} - pnt.y_i | \leq 4, i = (0,1,...,N-3) \qquad (7)$$

Заметим, что условия (6)-(7) являются симметричными относительно координат x и y.

Закрытием полосы является формирование СКР по центральным точкам серий полосы (рис. 1). Серии, образовавшие СКР, удаляются из полосы. СКР представляются только те серии полосы, чьи центральные точки соответствуют условиям (5)-(7).

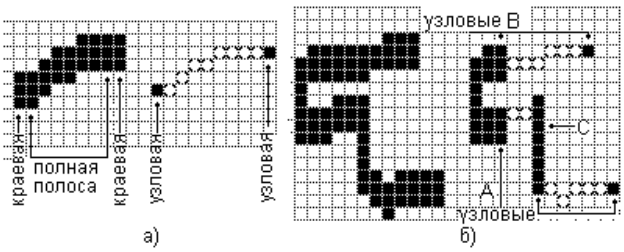


Рисунок 1: Пример закрытия полос: а) узловые серии являются краевыми; б) узловые серии A, B, C отличаются от краевых.

Рассмотрим полную полосу S. Серии, являющиеся родительской и дочерней по отношению к первой и последней сериям S соответственно, назовем краевыми. Обозначим краевые серии через E_1 и E_2. Узловыми будем называть серии, к которым прикрепляются СКР при закрытии полосы. Если краевая серия является начальной или конечной, то она является узловой. В этом случае узловая серия «сужается» до центральной точки. Пусть краевая серия E_1 будет серией слияния или ветвления. Рассмотрим путь E_1E_2. Тогда узловой будет серия E_{1+n}. В данном случае размеры серии не изменяются. Такой выбор узловых серий сохраняет топологию соединения.

Пусть имеется полная полоса S, состоящая из n серий R. Общее число отсчетов серий полосы S равняется m. После закрытия она представляется кривой C. Вычисляются следующие морфологические свойства. Площадь рассчитывается просто как количество пикселей, соответствующих полосе. Длина вычисляется как евклидово расстояние между центрами узловых серий. Ширина W рассчитывается по формуле:

$$W = \frac{m}{\sum_{i=1}^{n} \sqrt{1 + \left(\frac{R_i.beg + R_i.end - \left(R_{i-1}.beg + R_{i-1}.end\right)}{2} \right)^2}} \qquad (8)$$

3. ПОСТРОЕНИЕ ГРАФОВОЙ МОДЕЛИ БИНАРНОГО РАСТРА

Каждый столбец матрицы изображения представляется упорядоченным по возрастанию координат списком серий. Изображение хранится в виде массива списков серий (МСС). Выделим связные компоненты (СК) изображения. Благодаря RLE-кодированию возможно использование эффективного алгоритма[5]. Его основная идея заключается в том, что метка СК ассоциируется не с отдельным пикселем, а с серией. Однако для больших изображений размер таблицы эквивалентности является фактором, снижающим производительность. Мы применяем методику «разделяй и властвуй» для нахождения СК изображений большого размера. Основываясь на работе[6], разделим изображение на NxN частей. Для каждой части применим алгоритм[5], затем выполним процедуру слияния. Каждая СК соответствует объекту на изображении и будет представлена графом.

Нами предлагается использование быстрого алгоритма частичной скелетизации, основной задачей которого является нахождение осевых линий частей объектов простой формы. Более сложные фрагменты будут обработаны отдельно в дальнейшем. Скан-проход алгоритма заключается в следующем: изображение сканируется по вертикали, анализируется связность смежных серий и выделяются полосы. Найденные полосы закрываются.

После вертикального сканирования изображение поворачивается на 90°, снова выполняется скан-проход, затем изображение поворачивается в исходное положение (рис. 2).

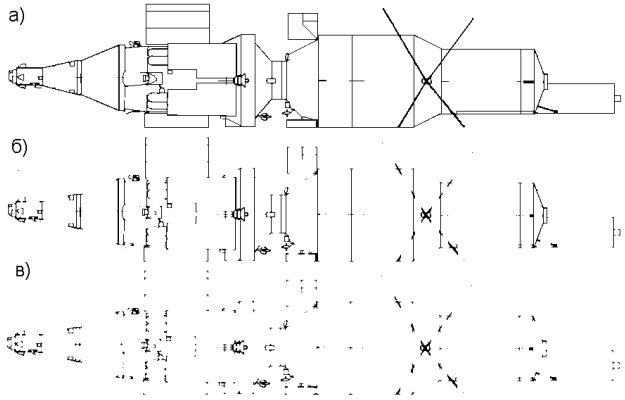


Рисунок 2: Пример работы алгоритма частичной скелетизации:а) исходная СК; б) вертикальный проход; в) горизонтальный проход.

В результате двух скан-проходов прямолинейные отрезки СК заменяются скелетными кривыми (СКР). Группы серий, которые не были заменены СКР на процедуре частичной скелетизации, представляют собой области соединений (например, X, T, Y -типа). Из области соединения (ОС) исходят СКР, аппроксимирующие относительно прямолинейные участки. Для каждой СКР, исходящей из ОС, получим вектор направления, построенный по ее начальным точкам. Найдём точку пересечения векторов направлений ОС и соединим ее отрезками с начальными точками СКР. Пометим точки растра, через которые проходят эти отрезки. Затем применим параллельный алгоритм утоньшения для ОС, который не будет удалять помеченные пиксели (рис. 3).

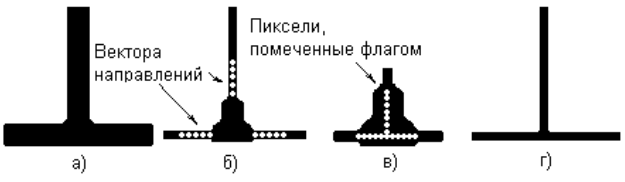


Рисунок 3. Обработка соединений. а) СК б) результат частичной скелетизации в) ОС, белым показаны точки соединения г) результат скелетизации

Таким способом, с помощью аналитической реконструкции обеспечивается корректная обработка соединений.

На следующем шаге анализом последовательности серий единичной ширины выделим группы серий, удовлетворяющие условиям (5-7). Заменим эти серии с помощью СКР. В данном случае СКР будет являться альтернативным способом представления последовательности пикселей единичной толщины. Перейдем к построению графовой модели. По координатам узловых серий (т.е. серий, которые указывают на СКР) сформируем множество вершин V. По точкам СКР построим ребра. Каждая СКР была прикреплена к двум сериям, которые преобразовались в вершины, следовательно, можно найти вершины, из которых исходят ребра. Рассмотрим два смежных ребра. Если они являются коллинеарными (допускается погрешность в 5°), то они могут быть заменены одним ребром.

В результате вышеописанных действий каждая СК изображения представляется нагруженным ориентированным планарным псевдографом, вершинам которого соответствуют концевые и узловые точки отрезков СК, а ребрам – сами отрезки СК, представленные в форме СКР. Независимые подходы к описанию и построению графовых моделей были предложены в[1, 7-8].

Псевдограф G задается парой G = (V, E), где:

V – множество вершин;

E – мультимножество ребер, каждое из которых соединяет две вершины из V, причем изображения ребер из E на плоскости не пересекаются, поэтому (V, E) представляет собой планарный граф. Каждое ребро имеет важные характеристики (длина, ширина, элонгация), которые могут быть использованы при последующем создании векторной модели.

Графовая модель является компактной формой представления СК изображения, которую значительно проще анализировать, чем исходную матрицу изображения. Она описывает топологию СК, связи между отрезками СК (ОСК) и позволяет осуществлять эффективное нахождение графических примитивов.

Под мультиточкой будем понимать точку пересечения как минимум трех отрезков[7]. Нахождение мультиточек является важной задачей при распознавании образов. Вершины графа, из которых исходит более трех ребер, являются мультиточками.

По графовой модели могут быть получены следующие характеристики:

а) отдельной СК: выделение петли на изображении СК; количество ОСК; длина каждого ОСК; общая длина всех ОСК; средняя длина всех ОСК; максимальная и минимальная длина ОСК; средняя элонгация всех ОСК; максимальная и минимальная элонгация ОСК; средняя ширина всех ОСК; максимальная и минимальная ширина ОСК.

б) изображения: количество объектов; количество ОСК всех объектов; суммарное, среднее, максимальное и минимальное значение параметров СК.

Граф может быть преобразован в более компактную форму гиперграфа[9], гиперребра которого состоят из ребер, соединяющих вершины степени 1 и 2 исходного графа (см. рис. 4).

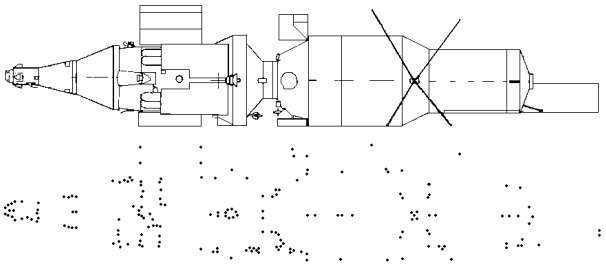


Рисунок 4: Исходное изображение и вершины гиперграфа.

4. ВЕКТОРИЗАЦИЯ

4.1 Нахождение путей векторизации

Для последующей постобработки поместим узловые и концевые точки в отдельные массивы (МУТ, МКТ). Следующим шагом является построение векторной модели на основании имеющегося графа. Из имеющегося псевдографа G получим гиперграф GG. На первом шаге процедуры построения выделим гиперребра графа GG, состоящие из ребер графа G, соединяющих вершины графа G степени один и два. Каждое гиперребро имеет две концевые вершины. Из каждой такой вершины гиперребра GE исходит ноль или более одно ребра E, не принадлежащих гиперребру GE. Рассмотрим 2 ребра E1 и E2, исходящих из вершины степени 3 графа G. Пусть ребра E1 и E2 принадлежат гиперребрам GE1 и GE2 соответственно. Если ребра E1 и E2 являются коллинеарными с некоторой погрешностью (например, 5°), то гиперребра GE1 и GE2 объединяются.

Введем понятие пути векторизации. Под путем векторизации (ПВ) будем понимать последовательность точек $p_i=(x_i, y_i)$, лежащих на средней линии СК. ПВ = $\{p_0, p_1, \ldots, p_n\}$. Если $p_0 = p_n$, то путь является закрытым. Точки ПВ описывают набор отрезков и дуг окружностей. Каждое гиперребро описывает ПВ (рис. 5).

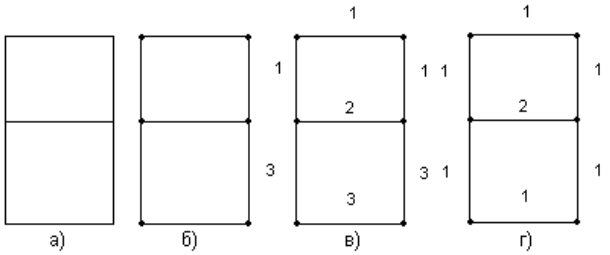


Рисунок 5: а) исходная связная компонента б) полученный граф в) цифрами показаны гиперребра GE на первом шаге построения г) цифрами показаны ПВ.

4.2 Векторизация

Для выделения отрезков применяется метод генерализации Дугласа-Пекера[10]. Пусть имеется путь векторизации ПВ = $\{p_0, p_1, \ldots, p_n\}$. Рассмотрим процедуру аппроксимации. Построим отрезок p_0p_n. Пусть p_k – самая удалённая от отрезка p_0p_n вершина. Если расстояние от p_k до отрезка p_0p_n меньше заданного порога, то p_0p_n аппроксимирует последовательность. В противном случае разобьем ПВ на 2

части: P1 = {p_0, …, p_k} и P2 = { p_{k+1},…, p_n}. Для каждой части рекурсивно применяется процедура аппроксимации. Введем понятия сегмента. Под сегментом будем понимать прямолинейный отрезок, полученный путем полигональной аппроксимации точек пути векторизации. Сегменты используются для нахождения дуг окружностей. Пусть имеется путь векторизации, представленный сегментами S1,S2, …,Sn. Рассмотрим пару смежных сегментов S_iS_{i+1}.Из геометрии известно, что три неколлинеарные точки X,Y, Z на плоскости однозначно определяют окружность. Центр окружности C будет находиться на пересечении перпендикуляров, опущенных на середины отрезков XY и YZ. Затем итеративно тестируются смежные сегменты на принадлежность дуге.

Полученные из сегментов отрезки и дуги окружностей получают характеристику ширины ребра графа G, из анализа которого они были получены.

В качестве постобработки предлагается стыковка отрезков в местах соединений. Это осуществляется анализом МУТ и МКТ. На рис.6 приведен пример работы алгоритма.

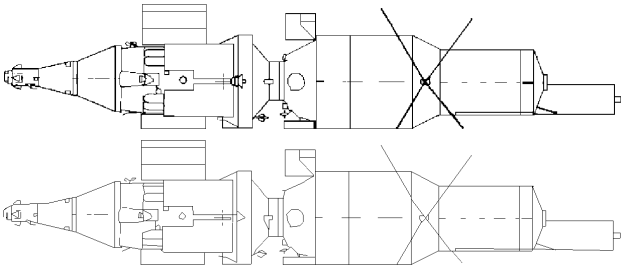


Рисунок 6: Исходное изображение и результат векторизации.

5. ЗАКЛЮЧЕНИЕ

Эффективный метод представления растрового изображения должен обеспечивать достаточную степень сжатия информации и в то же время позволять обрабатывать изображение напрямую в кодированной форме. Например, векторизация часто осуществляется для изображений большого размера, и разбиение изображения на фрагменты с последующей «сшивкой» является источником ошибок.

Мы предлагаем использование графовой модели как компактного средства хранения и описания структуры растра. Достоинством предлагаемого подхода является простота реализации и достаточно высокое быстродействие. Нами предлагается модификация алгоритма нахождения СК, созданная специально для обработки изображений большого размера. Благодаря кодированию концов серий построение скелета осуществляется быстрее, чем при использовании классических методов попиксельного анализа. Для САПР важна высокая точность нахождения линейных сегментов и точек их пересечения. Предлагается методика обработки соединений. Зоны соединений исключаются из процедуры скелетизации, и производится аналитическая реконструкция точек пересечения. Исходная СК может быть реконструирована анализом графа. Недостатком подхода является неполное описание площадных объектов. Полученная графовая модель успешно применяется при векторизации планов зданий, технических чертежей.

6. ЛИТЕРАТУРА

[1] Monagan, G. Appropriate base representation using a run graph./G. Monagan., M. Roosli//Document Analysis and Recognition:Proc. Int. Conf., Japan-p.623-626.

[2] Listing, J.1861, Dercensusraeumlicher complexe oder verallgemeinerungen des euler'schen satzes von den polyeadern., Abhandlungen der Mathematischen Classe der Koeniglichen Gesellschaft der Wissenschaften zu Goettingen.

[3] H. Blum. A transformation for extracting new descriptors of shape, Models for the Perception of Speech and Visual Form (W.Wathen-Dunn, ed.), MIT Press, pp. 362-380, 1967.

[4] G. Klette, "Topologic, Geometric, or Graph-Theoretic Properties of. Skeletal Curves," PhD dissertation, Groningen Univ., 2007.

[5] Linda G. Shapiro, Connected Component Labeling and Adjacency Graph Construction. – Topological algorithms for digital image processing – Elsevier Science B.V. - 1-31 - 1996.

[6] Jung-Me P., Carl G. Looney, Hui-Chuan C., Fast Connected Component Labeling Algorithm Using a Divide and Conquer Technique, CATA 2000 Conference on Computers and Their Applications, pp. 373 – 376, Dec. 2000.

[7] S. Di Zenzo, L. Cinque, S. Levialdi. Run-based algorithms for binary image analysis and processing// Pattern Analysis and Machine Intelligence, IEEE Transactions on PAMI, vol.18 NO.1 - January 1996 - pp. 83-89.

[8] Костюк Ю.Л., Новиков Ю.Л. Графовые модели цветных растровых изображений высокого разрешения // Вестник ТГУ. 2002. № 275, апрель. С.153–160.

[9] Berge, C., Ed. 1973. Graphs and Hypergraphs. American Elsevier, New York.

[10] Douglas, David H. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature./ David H. Douglas, Thomas K. Peucker - The Canadian Cartographer 10, 1973, № 2 - P. 112–122.

Благодарности

Автор выражает благодарность своим родителям, а также профессору Леониду Ивановичу Минченко и доценту Владимиру Яковлевичу Анисимову за ценные замечания и советы, и Борису Борисовичу Гребенщикову за неповторимую атмосферу, создаваемую его музыкой.

Об авторе

Максим Стержанов – аспирант БГУИР. email: accept@bk.ru.

Abstract

In this paper we introduce a hybrid vectorization methodology for line monochrome drawings. A graph model is utilised as an intermediate representation. This form of compression preserves the topology. We are extending further the work of Di Zenzo, Monagan and Pavlidis, amongst others. The algorithm is based on run-length encoding of the image. We examine the modification of extraction methods for connected components. Junctions are analysed separately using a specific procedure for analytical reconstruction.

Keywords: *vectorization, RLE, graph model.*

Моделирование отражательных свойств материалов плоских объектов по фотоизображениям

Андрей Ильин, Андрей Лебедев, Виталий Синявский, Алексей Игнатенко

Факультет Вычислительной Математики и Кибернетики

Московский Государственный Университет имени М.В. Ломоносова, Москва, Россия

{ailyin, alebedev, vsinyavsky, ignatenko}@graphics.cs.msu.ru

Аннотация

Синтез реалистичных изображений является обязательным этапом при проектировании и предварительном расчете освещенности материалов автомобилей, зданий, помещений и т.п. Для физической корректности синтеза необходимо иметь средства для адекватного представления моделей материалов в цифровом виде. В данной работе предлагается физически корректная модель освещения и способ её компактного хранения. Разработанная модель освещения автоматически реконструируется по небольшому числу фотоизображений плоской поверхности объекта. В работе предложена схема интерактивной визуализации модели материала с использованием графического процессора (GPU).

Ключевые слова: модели освещения, реконструкция моделей материалов, интерактивная визуализация, ДФО

1.Введение

В настоящее время существует множество различных подходов для реконструкции моделей материалов. Все они отличаются способом получения данных, используемой моделью освещения и её форматом хранения, схемой визуализации модели материала.

Предлагаемый нами подход основан на получении данных о материале с фотоизображений. При этом поверхность материала считается плоской, что избавляет от необходимости восстанавливать информацию о геометрии поверхности материала. Разработанные нами алгоритмы производят восстановление ДФО (двулучевая функция отражения) и представляют её в специальном формате. Для наиболее полного восстановления разработанной модели материалов требуется порядка 5-6 фотографий. При этом весь процесс реконструкции абсолютно автоматический. От пользователя требуется только набор фотоизображений.

Для материалов с ярким бликом имеется поддержка изображений широкого динамического диапазона, которые можно получить из нескольких обычных фотоизображений, сделанных с разной выдержкой.

Также были разработаны алгоритмы для интерактивной визуализации моделей с использованием графического процессора (GPU).

В данной работе были получены следующие результаты:

- Разработана модель освещения, являющаяся физически корректной, полностью реконструируемая по небольшому числу входных данных и обеспечивающая реалистичность на стадии визуализации;
- Реализован алгоритм для автоматической реконструкции предложенной модели освещения; проведена оценка объема входных данных, обеспечивающих требуемое качество на этапе визуализации;
- Предложен формат для хранения и визуализации моделей освещения, занимающий небольшой объем памяти; для формата предложена схема интерактивной визуализации на GPU;

В следующем разделе приводится краткий обзор существующих методов реконструкции моделей материалов. В разделе 3 производится описание необходимых входных данных для предложенных алгоритмов и их предобработка. В разделе 4 описывается реализованная модель освещения, формат ее хранения и приводится алгоритм восстановления данных моделей по фотоизображениям. Раздел 5 посвящен схеме интерактивной визуализации предложенной модели. В разделе 6 производится анализ необходимых данных. В заключении, в разделе 7 изложены основные результаты.

2.Обзор существующих методов

Все существующие модели освещения можно условно разбить на две группы моделей: аналитических и заданных ДФО (двулучевая функция отражения). Аналитические модели – это параметрическое семейство функций яркости точки материала от положения наблюдателя и источника света. Как правило, они зависят от небольшого числа параметров, что приводит к нереалистичному моделированию реальных материалов. К тому же некоторые аналитические модели (например, модель Фонга) не являются физически корректными, а значит, они непригодны для предварительного расчета освещения в задачах проектирования. В работе [1] представлена система, которая по набору фотоизображений восстанавливает аналитические модели. Из-за неточных и неполных данных о яркости материала, получаемых с фотографий, в оптимизационном процессе происходит неточное определение параметров модели.

Другой подход заключается в вычислении ДФО [2]. Часто ДФО представляется в табличном виде [3]. Доступ к таблице осуществляется по четырем аргументам: двум углам, задающим направление на источник света и двум углам, задающим направление на камеру. Для большинства материалов, необходима достаточно подробная таблица (достаточно малый шаг по углам) и, как следствие, большие объемы для хранения данный, записанных в таблице. Проблема хранения особенно актуальна для современных графических приложений, работающих в реальном времени. В них используют аппаратную поддержку вычислений на графическом процессоре, что требует хранения табличных ДФО в памяти видеокарты. Подобные аппаратные

ограничения сильно сужают класс реконструированных типов материалов.

Для решения проблем с необходимым большим объемом памяти применяют различные способы. Один из них – факторизация 4-х мерной функции на две двумерные функции. При этом используется свойство взаимности света [2]. Этот подход позволяет сильно сократить размеры таблиц. Также для него разработаны эффективные схемы интерактивной визуализации на GPU.

Одним из решений, является предоставление модели материала в виде табличной функции, определенной на неравномерной сетке. При этом сетка является более детализированной в зоне блика материала и менее детализированной в остальных областях. Однако при использовании подобного подхода сильно усложняются алгоритмы аппаратного ускорения визуализации таких сеточных функций.

Наш подход является компромиссом между непосредственным хранением табличной ДФО и параметрическим представлением материалов. При этом удается решить проблемы, связанные с физической корректностью модели, большими объемами памяти для хранения, сложностью реконструкции.

На стадии получения данных используются различные подходы. Некоторые из существующих методов предполагают использование сложных фотометрических установок [4]. Другие же используют фотоизображения объекта, сделанные с разных позиций камеры. При этом в ряде работ предлагается получать изображения с разной выдержкой для последующего восстановления изображений широкого динамического диапазона [5]. Это необходимо для восстановления моделей материалов с очень ярким бликом. В данной работе мы используем именно такой подход. При восстановлении моделей материалов по фотоизображениям критическим фактором является точность калибровки камеры и положения источника света.

В ряде работ помимо восстановления материала реконструируется информация о геометрии объектов. Это достигается путем введения в сцену дополнительных объектов, сделанных из того же материала. В работе [7] используется шарик для определения направления на источник света. Также предполагается, что рассматриваемые материалы хорошо отражают свет. В нашем подходе изначально предполагается, что поверхность материала является плоской.

Некоторые из существующих методов позволяют восстанавливать широкий класс материалов [8]. Наша модель освещения способна качественно задавать только изотропные материалы. Тем не менее, большинство материалов реального мира являются изотропными, поэтому подобные ограничения на не являются критичными.

3.Получение данных

Получение и предобработка данных состоит из трех шагов:

- Фотографирование плоской поверхности материала с разной выдержкой;
- Получение изображений широкого динамического диапазона;
- Восстановление положения источников света и камеры относительно образца материала.

На первом этапе необходимо сфотографировать плоский объект. Обязательным условием последующего корректного восстановления материала является регистрация бликов материала на фотографии. Для этой цели камера и источник располагаются специальным образом относительно образца (Рис.1).

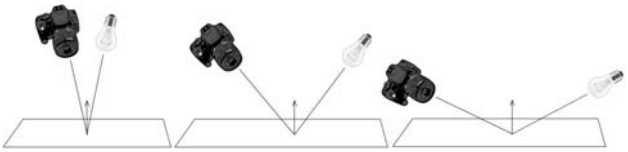


Рисунок 1: Схема расположения камер и источников относительно плоского объекта.

Для каждого положения камеры делается три снимка с разной выдержкой (Рис.2). Большинство реальных материалов устроено таким образом, что отношение яркости блика к яркости самых темных точек материала очень велико. Чтобы передать весь диапазон яркостей, необходимо использовать изображения широкого динамического диапазона. Мы получаем эти изображения из обычных изображений, сделанных с разной выдержкой, также как это сделано в [9].

Рисунок 2: Примеры фотографий, сделанных с разной выдержкой.

На последнем этапе происходит восстановление положения камеры и источника. Положение камеры вычисляется путем решения PnP проблемы [6]. Фактически рассматриваемые нами образцы являются прямоугольными, т.к. для удобства на любую поверхность, может быть положен лист бумаги с прямоугольным вырезом известного размера. Методами машинного зрения автоматически выделяются углы прямоугольника на фотоизображениях. В результате находятся соответствующие точки на разных фотоизображениях, и решается P4P задача. Затем методами машинного зрения на фотографии находится блик и производится расчет направления на источник света с использованием закона Снелла. Класс рассматриваемых нами материалов несколько сужается, поскольку априори считается, что материал отражает большую часть энергии в направлении зеркального отражения, определяемого согласно закону Снелла.

Данные о положении источников и камер, получаемые вышеописанными способами, обладают приемлемой точностью для последующего восстановления материалов.

4. Предлагаемая модель освещения

В качестве модели освещения используется ДФО в специальном формате. По построению нашего процесса геометрия объекта является плоскостью, что упрощает процесс вычисления ДФО.

Пусть $\vec{l}$ - вектор на источник света, v - вектор на камеру, n - нормаль к площадке. Тогда ДФО определяется формулой:

$$ДФО = \frac{L_0}{L_i \cdot \cos\phi_{in} \cdot dw}, \quad (1)$$

где L_0 - количество энергии, которое отражается от поверхности материала в направлении $\vec{v}$, L_i - количество энергии, которое приходит от источника в направлении $\vec{l}$, $\cos\varphi_{in} = -(\vec{l} \cdot n)$ - косинус угла между нормалью $\vec{n}$ и вектором на источник, dw - дифференциальный телесный угол, порожденный направлением $\vec{l}$.

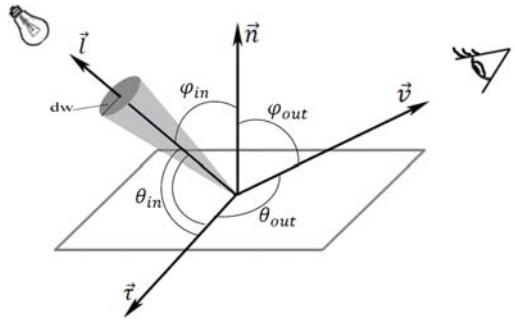


Рисунок 3: Определение ДФО

ДФО определяется двумя векторами $\vec{l}$, $\vec{v}$ и длиной волны падающего света. Зависимость от длины волны мы учитываем, рассматривая три значения ДФО для красной, зеленой и синей длин волн. Для получения значения ДФО для произвольной длины волны необходимо разложить цвет по трем составляющим R, G и B. Тогда результирующая ДФО будет линейной комбинацией ДФО для R, G и B с теми же коэффициентами.

Измерив ДФО для всевозможных пар направлений и длин волн, можно посчитать освещенность материала произвольной геометрии при произвольном освещении. Обобщенное освещение в данной точке объекта с учетом ДФО задается следующей формулой:

$$L_0(\vec{v}) = \int_{\Omega} ДФО(\vec{l}, \vec{v}) \cdot L_i(\vec{l}) \cdot \cos(\varphi_{in}) dw, \quad (2)$$

где Ω - полусфера всевозможных направлений вектора $\vec{l}$.

С вычислительной точки зрения сложность заключается в вычислении интеграла в формуле (2). Однако для точечных источников света интеграл можно заменить суммой и получим относительно простое выражение, которое можно быстро вычислить.

Перед тем, как описать предлагаемый формат хранения ДФО введем следующее определение среза ДФО. Срезом ДФО по направлению на источник называется ДФО с фиксированным направлением на источник света. Любой срез ДФО можно представить графически с помощью специальной поверхности. Поверхность строится так. Для произвольного направления w_o длина вектора, соединяющего точку пересечения w_o с поверхностью и точку падения света, равна значению функции среза в направлении w_o.

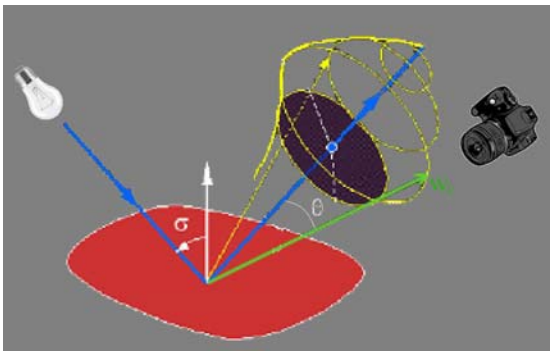


Рисунок 4. Поверхность ДФО

Основное предположение относительно восстанавливаемой функции – любой ее срез должен представлять фигуру вращения вокруг вектора зеркального отражения (Рис. 4).

Реконструкция модели состоит из следующих этапов:

- Получение облака точек ДФО из фотоизображений;
- Аппроксимация облака точек фигурой вращения при помощи алгоритма Левенберга-Марквардта.

Для восстановления фигуры вращения достаточно восстановить ее образующую кривую. В силу наших предположений о виде срезов ДФО, образующая является функцией одного угла θ (Рис.4). Эта функция восстанавливается по облаку точек при помощи нелинейной аппроксимации следующей параметрической кривой:

$$f(A_1, A_2, h, x_0) = A_2 + \frac{A_1 - A_2}{1 + e^{\left(\frac{\theta - x_0}{h}\right)}}, \quad (3)$$

где A_1, A_2, h, x_0 - параметры, θ - угол между вектором на камеру и вектором зеркального отражения.

Пример восстановленной образующей представлен на Рис.5

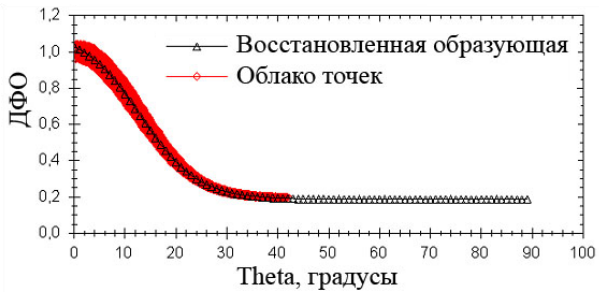


Рисунок 5. Пример восстановленной образующей

В результате строится набор образующих для нескольких фиксированных направлений на источник, данный набор и представляет формат для хранения ДФО.

5.Схема интерактивной визуализации

При визуализации материала с использованием графического акселератора набор восстановленных образующих передается на GPU в виде текстуры. Ее размер прямо пропорционален числу построенных срезов.

На практике число срезов совпадает с числом исходных фотографий. Обычно 5-6 срезов достаточно для корректного восстановления материала.

На графическом процессоре расчет освещения материала происходит следующим образом. Для каждой точки материала вычисляется направление на камеру и на источник света. По нормали к поверхности (поверхность может быть произвольной) и вектору на источник определяется угол падения света. По углу падения света находятся два ближайших среза ДФО, после чего происходит интерполяция между ними. Интерполяция производится при помощи сплайнов для согласования первых производных в местах стыков. Иначе, при обычной линейной интерполяции в местах стыков срезов наблюдались артефакты.

В результате, мы получаем срез для данного угла падения света. По определению среза, значение ДФО для данной пары векторов на камеру и на источник совпадут со значениями среза для данного угла падения. После вычисления ДФО производится стандартный расчет освещенности точки.

6. Анализ необходимых данных

Размер текстуры ДФО линейно зависит от количества использованных фотографий. Основная задача проведенного анализа - понять, сколько фотографий необходимо для полного восстановления модели материала.

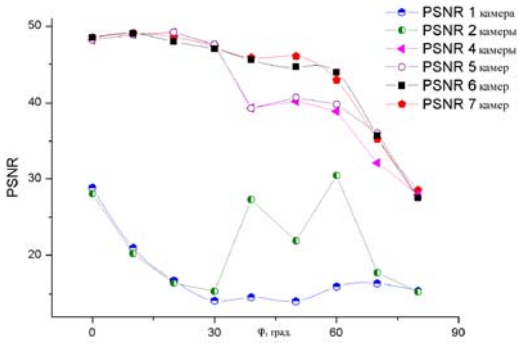


Рисунок 6. Анализ необходимых входных данных

Результаты анализа входных данных приведен на Рис.6: кривая «PSNR 1» соответствует ситуации восстановления модели материала всего по одной фотографии, «PSNR 2» по двум и т.д. Каждая точка графика соответствует одному сравнению фотографии реальной сцены с синтезированной фотографией восстановленного материала при помощи метрики PSNR. Угол φ, откладываемый по абсциссе, характеризует ориентацию образца относительно камеры и источника. Например, для случая восстановления по 5 камерам для угла $\varphi = 0$ реальная, синтезированная фотографии и результат их сравнения при помощи PSNR будут выглядеть как на Рис.7. По результатам анализа было обнаружено, что пяти-шести фотографий вполне достаточно для приемлемого восстановления модели материала.

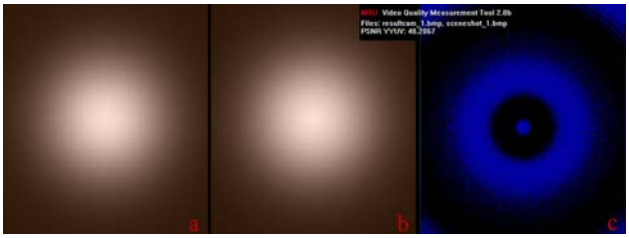


Рисунок 7. Пример реальной фотографии(a), синтезированной (b) и их сравнение при помощи PSNR (c).

7. Результаты

По результатам проведенных исследований был разработан формат представления ДФО, позволяющий восстанавливать модель материала по небольшому числу фотографий (5-6). Разработанные алгоритмы восстановления модели освещения позволяют реконструировать модель абсолютно автоматически. Проведенное сравнение при помощи метрики PSNR показало высокий уровень качества синтезируемых изображений (PSNR порядка 50). Также был разработан алгоритм интерактивной визуализации реконструированных моделей материалов. Проведенные тесты показали, результат приблизительно в 100 кадров в сек. на видеокарте Nvidia GeForce 6600, при визуализации объектов состоящих из 50K треугольников.

8. Литература

1. P.Sikachev, A.Ilyin, A.Ignatenko, *User-Assisted Acquisition, Processing and Rendering of Materials from Images.* Proc. of Graphicon'2007, pp. 131-134, Moscow, Russia, June 2007.

2. Chris Wynn. *An Introduction to BRDF-Based Lighting*, NVIDIA Corporation. http://developer.nvidia.com/attach/6568.

3. A. Ilyin, A. Lebedev, V. Sinyavsky, A. Ignatenko, *The System for the Acquisition, Processing and Material Rendering from Images*" Proc. of Graphicon'2008, pp. 134-141, Moscow, Russia, June 2008.

4. Волобой А.Г., Галактионов В.А., Ершов С.В., Летунов А.А., Потемин И.С. Аппаратно-программный комплекс для измерения светорассеивающих свойств поверхностей "Информационные технологии и вычислительные системы", № 4, 2006.

5. H. Lensch, M. Gösele, Y. -Y. Chuang, T. Hawkins, S. Marschner, W. Matusik, *Realistic Materials in Computer Graphics*. SIGGRAPH 2005 Tutorials, 2005.

6. Кравцов А., Вежневец В. *Решения PnP проблемы. Компьютерная графика и мультимедиа*. Выпуск №2(2)/2003 http://www.cgm.computergraphics.ru/content/view/42

7. Hertzmann A., Seitz S.M. *Example-Based Photometric Stereo: Shape Reconstruction with General, Varying BRDFs*. Pattern Analysis and Machine Intelligence, IEEE Transactions on Volume 27, Issue 8, pp.1254-1264, Aug. 2005.

8. G.J. Ward. *Measuring and modeling anisotropic reflection*. Computer Graphics, 26(2), pp. 265-272, July 1992.

9. Paul E. Debevec and Jitendra Malik. *Recovering High Dynamic Range Radiance Maps from Photographs*, Proceedings of SIGGRAPH 97, pp. 369-378.

Упорядочивание изображений для построения по ним 3d моделей

Д.Б. Волегов*, Д.В. Юрин**
*Московский физико-технический институт (государственный университет)
**Московский государственный университет им. М.В.Ломоносова
dvolegov@rambler.ru yurin_d@inbox.ru

Аннотация

В работе рассматривается задача последовательного построения 3d модели по фотографиям. Сначала обрабатывается небольшое число фотографий и строится первичная модель, затем обрабатываются следующие фотографии, модель уточняется и дополняется. Во-первых, рассматривается задача выбора последовательности изображений. Во-вторых, вводится понятие определенности фундаментальной матрицы, которое позволяет судить о точности реконструкции по паре кадров без выполнение таковой. Приводятся результаты применения предлагаемого метода к реальным и синтезированным последовательностям изображений.

Keywords: *упорядочивание изображений, построение 3d моделей, фундаментальная матрица.*

1. ВВЕДЕНИЕ

На текущий момент разработано множество методов автоматического построения 3d моделей объектов по изображениям. Методы реконструкции могут быть разделены на два типа: пакетная реконструкция и последовательная реконструкция. Методы первой группы обрабатывают все изображения сразу. К ним относятся методы на основе факторизации матриц и его вариации. Методы второй группы основаны на последовательной обработке изображений - модель сначала строится по небольшому числу кадров и постепенно дополняется и уточняется с подключением остальных кадров. К таким методам относятся [1], [2], [3], [4].

В настоящей работе рассматривается задача упорядочивания изображений для методов последовательной реконструкции. В [3], [4] порядок изображений известен, так как изображения образуют последовательность. В [2] проблема построения последовательности решается явно. А именно подход [2] состоит сначала в выборе пары изображений с максимальным количеством соответствий. А следующим обрабатывается изображение, для которого число парных соответствий с предыдущим максимально. В [1] изображения упорядочиваются таким образом, чтобы число уже реконструированных точек, видимых на добавляемом изображении, было максимальным.

Названные подходы к упорядочиванию изображений могут таить в себе опасность. Например, пусть сцена снимается из трех точек, первые две из которых расположены друг к другу гораздо ближе, чем к третьей. Скорее всего, для пары изображений (1,2) найдется больше соответствий, чем для пар (1,3) или (2,3). Но, в силу малого расстояния между точками 1,2 взаимное положение камер в этих точках плохо определено, и подход, используемый в [2] может привести к ошибочному результату. Это же относится и к подходу, используемому в [1], так как большое количество общих точек между обработанными и новым кадром не гарантирует большой базы съемки.

Часто для реконструкции координат точек и камер используются методы [5] на основе нелинейной минимизации некоторой ошибки. В этом случае чем ближе начальная структура сцены соответствует минимуму, тем меньше вероятность попасть в ложный локальный минимум. Следовательно, несмотря на малое число соответствий, более удачным является вычисление положений камер в точках 1,3 (или 2,3) и последующем вычислении положения в точке 2 (или 1).

Удачный выбор последовательности реконструкции важен по следующим причинам. Во-первых, если для реконструкции используются методы на основе нелинейной минимизации [5], то уменьшается вероятность попадания в ложные локальные минимумы и нахождения ложного решения, т.е. увеличивается стабильность алгоритмов реконструкции. Во-вторых, может уменьшиться время работы алгоритмов минимизации, так как уменьшается число итераций.

В настоящей работе построение 3d модели (облака точек) состоит из следующих основных этапов.

- На всех изображениях ищутся особенности. В настоящей работе применяется многомасштабный детектор особенностей типа blob, в качестве функции отклика детектора используется нормированный Лапласиан [6].
- Вычисляются дескрипторы особенностей, найденных на всех изображениях. В настоящей работе используется дескриптор SIFT [7].
- Строится индекс дескрипторов, который позволяет для заданного дескриптора быстро найти несколько ближайших дескрипторов для особенностей этого же и других изображений. Время поиска составляет $O(N^{\alpha})$, где N - общее число особенностей на всех изображениях, $\alpha \in [0,1]$. Параметр α контролирует степень компромисса между качеством и скоростью поиска. Основная идея в построении индекса состоит в том, чтобы сократить размерность дескриптора со 128 до примерно 30-40 методом анализа главных компонент (principal component analysis), а затем использовать деревья для ускорения поиска. В ходе экспериментов на реальных данных было выяснено, что приемлемым значением является $\alpha \in [0.3, 0.4]$. Детальное описание процесса построения индекса выходит за рамки настоящей работы и будет опубликовано в ближайшее время
- Строится таблица парных соответствий. Для каждого дескриптора $\mathcal{D}$, используя индекс

дескрипторов, ищется ближайший дескриптор $\mathcal{D}_1$ (изображения, соответствующие $\mathcal{D}$ и $\mathcal{D}_1$ различны). Для $\mathcal{D}_1$ также ищется ближайший дескриптор $\mathcal{D}_2$. Если дескрипторы $\mathcal{D}$ и $\mathcal{D}_2$ совпадают, то соответствие $\{\mathcal{D}, \mathcal{D}_1\}$ заносится в таблицу. Вместо поиска одного ближайшего дескриптора возможен поиск взаимных пар среди k ближайших соседей. Это увеличивает время работы, но и число парных соответствий.

- Строится список пар изображений, между которыми есть достаточное ($\sim$ 30) число соответствий, используя таблицу парных соответствий. Для каждой такой пары вычисляется фундаментальная матрица робастным (устойчивым к ложным соотетствиям) методом [8] (p. 290) на основе RANSAC [9].

- Строится граф, вершины которого соответствуют изображениям. Пару вершин соединяет ребро, если между соответствующими изображениями вычислена фундаментальная матрица. Граф разбивается на связные компоненты, каждая из которых обрабатывается независимо.

- Ребрам графа назначаются веса. В настоящей работе вводится мера определенности фундаментальной матрицы (см. 2.1), которая является весом ребра. Введенная мера обосновывается в разделе 2.2 Используя алгоритм раздела 3., строится последовательность, в соответствии с которой обрабатываются изображения. Сначала вычисляются матрицы первых двух камер и структура сцены, затем вычисляется матрица следующей камеры, структура уточняется и дополняется и т.д.

Для вычисления определенности фундаментальной матрицы необходимы лишь координаты соответствующих точек. В [10] рассматривается вычисление матрицы ковариации элементов фундаментальной матрицы и демонстрируются несколько применений: оценка ширины полосы вдоль эпиполярной линии, ошибок координат реконструированных точек и ошибки параметров калибровки камер. Целью предлагаемой ОФМ является простой способ оценки достижимой точности реконструкции на раннем этапе обработки и выбор подходящей последовательности реконструкции.

2. ОПРЕДЕЛЕННОСТЬ ФУНДАМЕНТАЛЬНОЙ МАТРИЦЫ

2.1 Определение

Для однородных координат соответствующих точек для пары изображений выполняется соотношение [8]:

$$\vec{y}_i^T \mathbf{F} \vec{x}_i = 0, \forall \quad i = 1 \dots N \tag{1}$$

Через $\vec{x}_i$ обозначены векторы однородных координат точки i на первом изображении, через $\vec{y}_i$ - соответствующей ей точки на втором изображении, $\mathbf{F}$ - фундаментальная матрица между парой изображений, $\vec{x}_{i,3} = \vec{y}_{i,3} = 1$, N - число соответствий между парой кадров.

Используя обозначения,

$$\begin{aligned} \vec{f} &= (F_{11}F_{21}F_{31}F_{12}F_{22}F_{32}F_{13}F_{23}F_{33})^T & (2)\\ Z_{ij} &= x_{i,k}y_{i,l}, \quad j = 3(k-1)+l & (3)\\ k,l &= 1 \dots 3, \quad j = 1 \dots 9 \end{aligned}$$

соотношения (1) переписываются в матричной форме:

$$\mathbf{Z}\vec{f} = 0 \tag{4}$$

Сингулярное разложение матрицы $\mathbf{Z}$ записывается в виде:

$$\mathbf{Z} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{5}$$

Вводится мера определенности фундаментальной матрицы:

$$q(\mathbf{F}) = \frac{\sigma_2 - \sigma_1}{\sqrt{<\delta\sigma_1^2> + <\delta\sigma_2^2>}} \tag{6}$$

В (6) σ_1, σ_2 - пара наименьших сингулярных чисел матрицы $\mathbf{Z}$, $\sigma_1 < \sigma_2$, а $<\delta\sigma_1^2>$, $<\delta\sigma_2^2>$ - средние квадраты отклонений сингулярных чисел σ_1, σ_2, связанные с погрешностью координат соответствующих точек на изображениях и вычисляются по (27), (29).

Определенность фундаментальной матрицы обладает следующими свойствами. Во-первых, если одно изображение может быть получено из другого проективным преобразованием, то определенность фундаментальной матрицы равна нулю. Названная ситуация имеет место, например, когда центры камер совпадают или сцена плоская и реконструкция невозможна. Во-вторых, в среднем определенность фундаментальной матрицы растет с увеличением базы съемки. Таким образом, введенная мера является полуколичественной оценкой того, насколько точно можно вычислить структуру сцены по паре кадров без непосредственного ее вычисления.

2.2 Обоснование

Фундаментальная матрица связана с геометрией съемки и матрицами внутренней калибровки камер следующим образом [8] (p. 246):

$$\mathbf{F} = \mathbf{K}_2^{-T}[\vec{t}]_\times \mathbf{R}\mathbf{K}_1^{-1} \tag{7}$$

В (7), $\mathbf{K}_1$, $\mathbf{K}_2$ - матрицы 3×3 внутренних параметров камер, $\mathbf{R}$ - ориентация второй камеры относительно первой, $\vec{t}$ - сдвиг от фокуса первой камеры к фокусу второй. Матрица $[\vec{t}]_\times$ имеет вид:

$$[\vec{t}]_\times = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix} \tag{8}$$

Теорема 1 *1. Если координаты точек связаны проективным преобразованием, то размерность нуль-пространства матрицы* $\mathbf{Z}$ *не менее трех. 2. Если центры камер совпадают, то элементам нуль-пространства соответствует множество фундаментальных матриц, у которых вектор* $\vec{t}$ *в факторизации (7) может быть произвольным.*

Согласно условию теоремы координаты соответствующих точек связаны проективным преобразованием:

$$\vec{y}_i = \mathbf{H}\vec{x}_i \tag{9}$$

Подставив (9) в (1):

$$\vec{x}_i^T \mathbf{H}^T \mathbf{F} \vec{x}_i = 0, \forall \quad i = 1 \dots N \tag{10}$$

Очевидно, что если

$$\mathbf{H}^T \mathbf{F} = \mathbf{A} \tag{11}$$

где $\mathbf{A}$ ***произвольная*** *антисимметричная матрица, то (10) выполняется для произвольных* $\vec{x}_i$.

1. Построим три взаимно ортогональных вектора $\vec{f}_k$, $k = 1, 2, 3$, *лежащих в нуль-пространстве матрицы* $\mathbf{Z}$. *Для этого рассмотрим матрицу* $\mathbf{H}^{-1}\mathbf{H}^{-T}$. *Так как она симметрична, то может быть представлена в виде:*

$$\mathbf{H}^{-1}\mathbf{H}^{-T} = \mathbf{Q}\mathbf{S}\mathbf{Q}^T \tag{12}$$

где собственные векторы матрицы $\mathbf{H}^{-1}\mathbf{H}^{-T}$ *являются столбцами матрицы* $\mathbf{Q}$ *и обозначаются через* $\vec{q}_k$, *а ее собственные значения сосредоточены на диагонали* $\mathbf{S}$. *Определим матрицы* $\mathbf{A}_k$, $\mathbf{F}_k$:

$$\mathbf{A}_k = [\vec{q}_k]_\times \tag{13}$$

$$\mathbf{F}_k = \mathbf{H}^{-T}\mathbf{A}_k \tag{14}$$

Положим векторы $\vec{f}_k$ *соответствующими матрицам* $\mathbf{F}_k$ *из (14) согласно (2). Векторы* $\vec{f}_k$ *лежат в нуль-пространстве матрицы* $\mathbf{Z}$, *так как (10) выполняется для* $\mathbf{F}$, *равных* $\mathbf{F}_k$ *из (14). Покажем, что* $\vec{f}_k$ *взаимно ортогональны.*

Из (2), используя (14), следует:

$$\begin{aligned}(\vec{f}_k, \vec{f}_l) &= tr\left(\mathbf{F}_k^T \mathbf{F}_l\right) = -tr\left(\mathbf{A}_k \mathbf{H}^{-1}\mathbf{H}^{-T}\mathbf{A}_l\right) = \\ &= -tr\left(\mathbf{A}_l \mathbf{A}_k \mathbf{H}^{-1}\mathbf{H}^{-T}\right), \quad k, l = 1, 2, 3\end{aligned} \tag{15}$$

Подставив (13), (12) в (15), и, используя ортогональность векторов $\vec{q}_k$, *получаем, например, для* $k = 1, l = 2$

$$\begin{aligned}(\vec{f}_1, \vec{f}_2) &= -tr\left([\vec{q}_2]_\times [\vec{q}_1]_\times \mathbf{Q}\mathbf{S}\mathbf{Q}^T\right) = \\ &= -tr\left([\vec{q}_2]_\times (\vec{0}|\vec{q}_3| - \vec{q}_2)\mathbf{S}\mathbf{Q}^T\right) = \\ &= -tr\left((\vec{0}|\vec{q}_1|\vec{0})\mathbf{S}\mathbf{Q}^T\right) = -tr\left(\mathbf{Q}^T(\vec{0}|\vec{q}_1|\vec{0})\mathbf{S}\right) = \\ &= -tr\left((\vec{0}| \quad 1 \quad 0 \quad 0 \quad {}^T|\vec{0})\mathbf{S}\right) = 0\end{aligned} \tag{16}$$

Аналогично, ортогональны и остальные пары $\vec{f}_k$, $\vec{f}_l$ *для несовпадающих* k, l.

Таким образом, построены три взаимно ортогональных вектора $\vec{f}_k$, *в нуль-пространстве матрицы* $\mathbf{Z}$ *и соответствующие им фундаментальные матрицы* $\mathbf{F}_k$. *Следовательно, размерность нуль-пространства* $\mathbf{Z}$ *не менее трех.*

2. Пусть центры камер совпадают. Обозначим через $\mathcal{N}$ *трехмерное пространство, натянутое на векторы* $\vec{f}_k, k = 1, 2, 3$. *Легко показать, что матрица* $\mathbf{H}$ *имеет вид:*

$$\mathbf{H} = \mathbf{K}_2 \mathbf{R} \mathbf{K}_1^{-1} \tag{17}$$

Согласно (7):

$$[\vec{t}]_\times = \mathbf{K}_2^T \mathbf{F} \mathbf{K}_1 \mathbf{R}^T \tag{18}$$

Выражая $\mathbf{F}$ *из (11) и подставляя в (18), получим:*

$$\begin{aligned}[\vec{t}]_\times &= \mathbf{K}_2^T \mathbf{H}^{-T} \mathbf{A} \mathbf{K}_1 \mathbf{R}^T = \\ &= \mathbf{K}_2^T \mathbf{K}_2^{-T} \mathbf{R} \mathbf{K}_1^T \mathbf{A} \mathbf{K}_1 \mathbf{R}^T = \mathbf{R} \mathbf{K}_1^T \mathbf{A} \mathbf{K}_1 \mathbf{R}^T\end{aligned} \tag{19}$$

В (19) матрица $\mathbf{A}$ *является произвольной антисимметричной матрицей, и это же относится к матрице* $[\vec{t}]_\times$, *так как для произвольной* $[\vec{t}]_\times$ *найдется матрица* $\mathbf{A} = \mathbf{K}_1^{-T}\mathbf{R}^T[\vec{t}]_\times \mathbf{R}\mathbf{K}_1^{-1}$, *согласно (19). Таким образом, вектор* $\vec{t}$ *в факторизации (7) фундаментальной матрицы может быть любым* ■

Результат 1 *Если центры камер совпадают или точки сцены расположены на плоскости, то определенность фундаментальной матрицы равна нулю.*

Из условий теоремы следует, что точки связаны проективным преобразованием, тогда согласно теореме 1 размерность нуль-пространства матрицы $\mathbf{Z}$ *не менее трех:*

$$\sigma_1 = \sigma_2 = \sigma_3 = 0 \tag{20}$$

и согласно (6) определенность фундаментальной матрицы равна нулю ■

Теорема 2 *Погрешность фундаментальной матрицы обратно пропорциональна разности* $\sigma_2 - \sigma_1$.

Используя результат [11] о том, что возмущение угла поворота сингулярного вектора обратно пропорционально разности между соответствующим сингулярным числом и ближайшим к нему сингулярным числом (sinθ-теорема), получаем:

$$\frac{\| \delta\mathbf{F} \|}{\| \mathbf{F} \|} \sim \frac{\| \delta\mathbf{Z} \|}{\sigma_2 - \sigma_1} \tag{21}$$

где $\| \delta\mathbf{F} \|$ *- возмущение фундаментальной матрицы, связанное с возмущением компонент матрицы* $\mathbf{Z}$, *т.е. погрешностями координат соответствующих точек на изображениях. Полагая норму фундаментальной матрицы равной единицы, получаем:*

$$\frac{\| \delta\mathbf{F} \|}{\| \mathbf{F} \|} \sim \| \delta\mathbf{Z} \| \tag{22}$$

Используя (6) получаем, что погрешность фундаментальной матрицы обратно пропорциональна введенной мере ее определенности ■

Результат 2 *Погрешность реконструкции обратно пропорциональна определенности фундаментальной матрицы с коэффициентом, зависящим от сцены и положения камер:*

$$\epsilon \sim \frac{C}{q(\mathbf{F})} \tag{23}$$

В (23) ϵ *- относительная ошибка реконструкции по паре кадров,* $q(\mathbf{F})$ *- определенность фундаментальной*

матрицы, константа C зависит от сцены и расположения камер.

Используя результаты (26), (10) работы [10] заключаем, что погрешность реконструированных точек прямо пропорциональна погрешности фундаментальной матрицы. И, используя, результат теоремы 2 заключаем, что погрешность реконструированных по паре кадров точек обратно пропорциональна степени определенности фундаментальной матрицы матрицы между этой парой кадров. Численные значение коэффициента C обсуждаются в конце подраздела 2.3 ■

Остаток настоящего раздела посвящен выводу формул для вычисления $< \delta\sigma_n^2 >$, используемых в (6).

Из-за конечной точности детектора особенностей их координаты подвержены шуму. Тогда элементы матрицы $\mathbf{Z}$ также подвержены возмущениям δZ_{ij}. Отклонение $\delta\sigma_n$ сингулярного числа σ_n записывается в виде:

$$\delta\sigma_n = \sum_{i,j=(1,1)}^{i,j=(N,9)} \frac{\partial\sigma_n}{\partial Z_{ij}} \delta Z_{ij} \tag{24}$$

Средний квадрат отклонения сингулярных чисел $< \delta\sigma_n^2 >$ записывается в виде:

$$< \delta\sigma_n^2 > = \left\langle \sum_{i,j=1,1}^{i,j=N,9} \frac{\partial\sigma_n}{\partial Z_{ij}} \delta Z_{ij} \sum_{k,l=1,1}^{N,9} \frac{\partial\sigma_n}{\partial Z_{kl}} \delta Z_{kl} \right\rangle =$$
$$= \sum_{i,j,k,l=1,1,1,1}^{N,9,N,9} \frac{\partial\sigma_n}{\partial Z_{ij}} \frac{\partial\sigma_n}{\partial Z_{kl}} < \delta Z_{ij} \delta Z_{kl} > \tag{25}$$

В [12] показывается, что производные сингулярных чисел произвольной матрицы по элементам этой матрицы записываются в виде:

$$\frac{\partial\sigma_k}{\partial Z_{ij}} = U_{ik} V_{jk}, \quad k = 1 \ldots 9 \tag{26}$$

где U_{ik}, V_{jk} - элементы матриц $\mathbf{U}$, $\mathbf{V}$ в сингулярном разложении (5). Используя (26) и независимость отклонения координат различных точек:

$$< \delta\sigma_n^2 > = \sum_{i,j,k,l=1,1,1,1}^{N,9,N,9} U_{in} V_{jn} U_{kn} V_{ln} < \delta Z_{ij} \delta Z_{kl} > =$$
$$= \sum_{i=1}^{N} U_{in}^2 \vec{v}_n^T \mathbf{W}_i \vec{v}_n \tag{27}$$

В (27) вектор $\vec{v}_n$ - n-ый столбец матрицы $\mathbf{V}$. Матрица $\mathbf{W}_i$ имеет размеры 9×9 и определяется моделью ошибки координат i-ой пары соответствующих точек, а ее элементы имеют вид:

$$W_{i,kl} \equiv < \delta Z_{ik} \delta Z_{il} > \tag{28}$$

Если положить, что координаты всех точек подвержены независимому нормальному шуму с дисперсией s^2, то из (3) матрицы $\mathbf{W}_i$ симметричны и имеют вид (29)

2.3 Численные эксперименты

В настоящем разделе рассматриваются численные эксперименты, проведенные для исследования определенности фундаментальной матрицы. Эксперименты проводились для различных конфигураций точек в пространстве и траекторий движения камеры.

Конфигурации точек

1. Точки равномерно распределены внутри сферы с центром в начале координат и радиусом 1.
2. Точки равномерно распределены внутри эллипсоида с центром в начале координат и полуосями (1,1,0.25).
3. Точки равномерно распределены внутри эллипсоида с центром в начале координат и полуосями (0.25,1,1).

Траектории движения камеры. Перемещения камеры происходят в плоскости xz.

1. Боковое движение. Камера движется от точки (0,0,-4) до точки (3,0,-4) вдоль прямой; оптическая ось остается параллельной.
2. Круговое движение. Камера движется от точки (0,0,-4) к точке (4,0,0) вдоль окружности, с центром в начале координат и проходящей через начальную и конечные точки; оптическая ось камеры проходит через начало координат.
3. Движение к сцене. Камера движется от точки (0,0,-4) до точки (0,0,-1) вдоль прямой; оптическая ось остается параллельной.

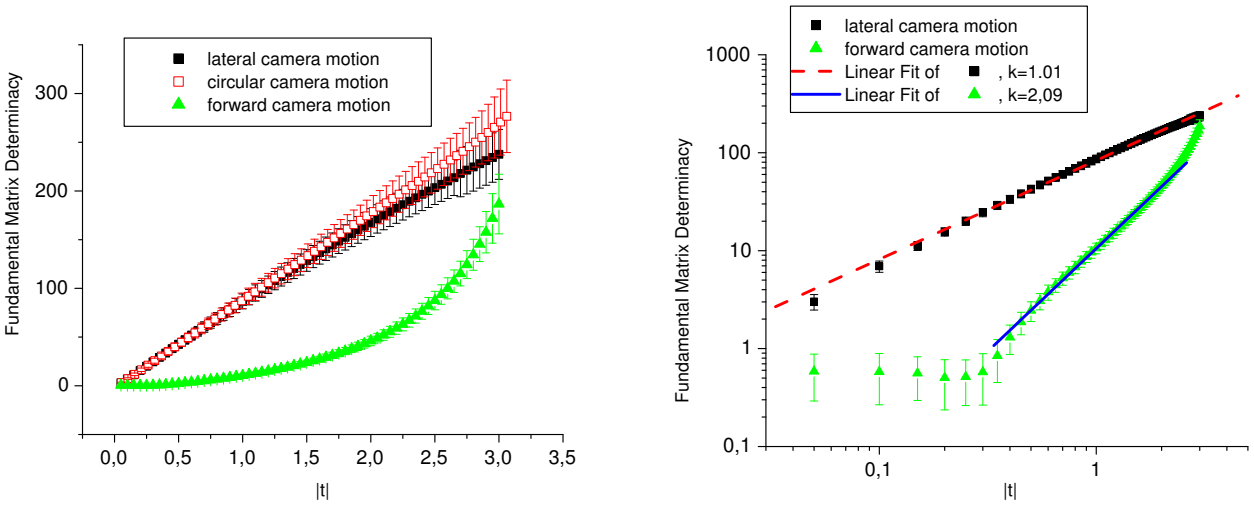
Рассматриваются именно названные траектории движения камер, так как они моделируют часто встречающиеся в реальности последовательности фотографий, полуаемых при приближении к объекту или при его круговом обходе.

На Рис.1 представлены зависимости ОФМ для различных траекторий движения камеры и конфигурации точек 1. Для всех представленных результатов число точек равно 100. Результаты экспериментов усреднялись по 100 случайным конфигурациям точек. Пределы ошибки на рисунках соответствуют стандартному отклонению при усреднении. Матрицы внутренних параметров $\mathbf{K}_1$, $\mathbf{K}_2$ обеих камер полагались единичными. К проекциям точек добавлялся случайный шум со стандартным отклонением 10^{-3}.

На Рис.1a видно, что зависимости для траекторий 1 и 2 очень похожи, поэтому в дальнейшем рассматриваются лишь траектории 1 и 3. На Рис.1b зависимости представлены в двойном логарифмическом масштабе. Следует отметить, что в широких пределах изменения ОФМ ($\gtrsim 3$) выдерживается степенная зависимость: для траектории 1 - линейная, для траектории 3 - квадратичная.

На Рис.2 представлены зависимости средней относительной погрешности реконструированных точек от ОФМ. Каждая точка соответствует одному эксперименту. Для эксперимента задается положение камер и генерируется облако точек. Вычисляются проекции точек, к ним добавляется шум. Из фундаментальной матрицы вычисляются структура сцены [13] и матрицы камер [14]. Далее ищется преобразование, совмещающее реконструированные

$$\mathbf{W}_i = s^2 \begin{pmatrix} x_{i,1}^2+y_{i,1}^2 & y_{i,1}y_{i,2} & y_{i,1}y_{i,3} & x_{i,1}x_{i,2} & 0 & 0 & x_{i,1}x_{i,3} & 0 & 0 \\ - & x_{i,1}^2+y_{i,2}^2 & y_{i,2}y_{i,3} & 0 & x_{i,1}x_{i,2} & 0 & 0 & x_{i,1}x_{i,3} & 0 \\ - & - & y_{i,3}^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ - & - & - & x_{i,2}^2+y_{i,1}^2 & y_{i,1}y_{i,2} & y_{i,1}y_{i,3} & x_{i,2}x_{i,3} & 0 & 0 \\ - & - & - & - & x_{i,2}^2+y_{i,2}^2 & y_{i,2}y_{i,3} & 0 & x_{i,2}x_{i,3} & 0 \\ - & - & - & - & - & y_{i,3}^2 & 0 & 0 & 0 \\ - & - & - & - & - & - & x_{i,3}^2 & 0 & 0 \\ - & - & - & - & - & - & - & x_{i,3}^2 & 0 \\ - & - & - & - & - & - & - & - & 0 \end{pmatrix} \quad (29)$$



(a) Линейный маштаб

(b) Двойной логарифмический масштаб. Для бокового движения зависимость примерно линейная, для движения вперед - примерная квадратичная

Figure 1: Зависимость ОФМ от модуля сдвига камеры для различных конфигурация точек и траекторий движения камеры

точки с исходными, и вычисляется средняя относительная погрешность реконструированных точек.

Результаты численных экспериментов показывают, что:

1. Ошибка реконструкции не бывает маленькой при маленькой ОФМ, и большое значение ОФМ не гарантирует маленькой ошибки. Это значит, что большое значение ОФМ является необходимым, но недостаточным условием малой ошибки.
2. Выдерживается грубая обратная зависимость между ОФМ и относительной ошибкой реконструкции в соответствии с результатом 2.
3. Для большинства сцен коэффициент пропорциональности C лежит в диапазоне $C \in [0.1, 1]$

3. УПОРЯДОЧИВАНИЕ ИЗОБРАЖЕНИЙ

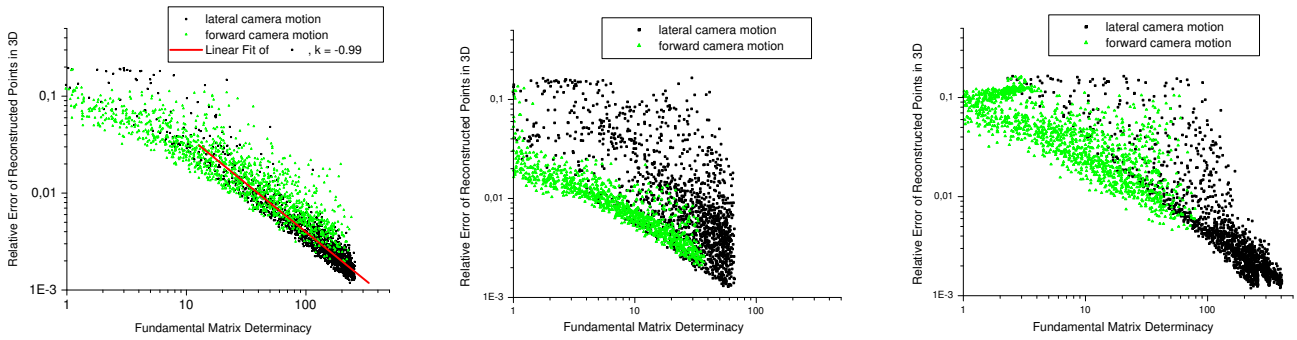
Упорядочивание изображений состоит в определении последовательности, согласно которой вычисляются структура сцены и матрицы камер. Для этого строится взвешенный граф $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, вершины $\{v_i\}$ которого соответствуют изображениям. Пару вершин соединяет ребро, если между соответствующими вершинам изображениями вычислена фундаментальная матрица. Вес ребра $w(v_i, v_j)$ между парой вершин v_i, v_j полагается равным определенности соответствующей фундаментальной матрицы. Сначала обрабатывается пара изображений, соответствующих вершинам v_1, v_2, потом добавляется v_3 и т.д. Назначение номеров вершинам выполняется следующим образом.

Алгоритм упорядочивания изображений

- Выбрать пару вершин, между которыми вес ребра максимален. Назначить этим вершинам номера 1 и 2.
- Следующий номер назначается той вершине, для которой сумма весов ребер из этой вершины к уже выбранным максимальна среди непронумерованных вершин:

$$\sum_{k=1}^{i-1} w(v_k, v_i) \geq \sum_{k=1}^{i-1} w(v_k, v), \quad \forall v \in \{\mathcal{V} \backslash \{v_1, \dots v_i\}\}$$

Отметим, что в [2] выбирается изображение, для которого число соответствий между ним и предыдущим



(a) Конфигурация точек 1. Наклон прямой, проведенной через точки в двойном логарифмическом масштабе, означает примерно обратную зависимость точности ре конструкции от ОФМ

(b) Конфигурация точек 2

(c) Конфигурация точек 3

Figure 2: Зависимость относительной ошибки реконструкции координат точек в пространстве от ОФМ для различных конфигураций точек и траекторий движения камеры

максимально. Предлагаемый алгоритм упорядочивания похож на [2], но отличается способом назначения весов ребрам графа.

4. РЕЗУЛЬТАТЫ

4.1 Синтезированные данные

Генерируется облако из $N = 100$ точек внутри единичной сферы с центром в начале координат. Генерируются положения 30 камер внутри сферы радиусом 4 и вне единичной сферы с центрами в начале координат. Оптические оси камер проходят через начало координат. Количество соответствий между парой кадров i, j задается формулой:

$$N_{ij} = N(1 - \frac{\alpha_{ij}}{\alpha_0})(1 - \frac{|d_i - d_j|}{d_i + d_j}) \qquad (30)$$

В (30) α_{ij} - угол между оптическими осями камер i и j, а d_i, d_j - расстояние от фокусов камер до начала координат, $\alpha_0 = \frac{\pi}{3}$. Зависимость (30) числа соответствий моделирует уменьшение их числа при увеличении угла съемки и масштаба между кадрами. К проекциям точек добавляется шум и, используя (30), вычисляются фундаментальные матрицы между кадрами, их меры определенности и строится последовательность реконструкции.

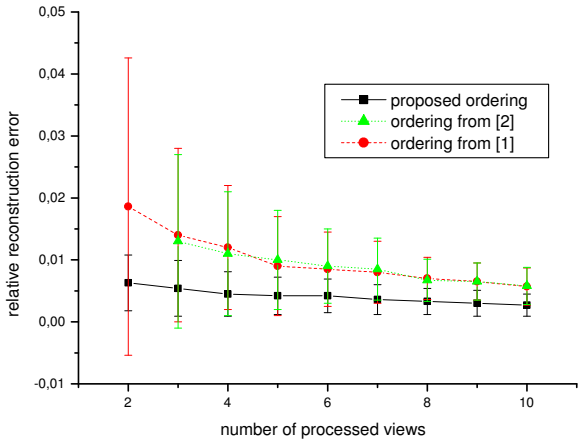
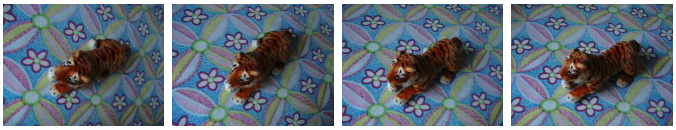


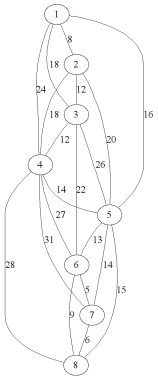
Figure 3: Зависимость относительной ошибки реконструкции координат точек от числа обработанных кадров

На Рис.3 приведена зависимость относительной ошибки реконструкции от числа обработанных кадров для предлагаемого способа построения последовательности и способа из [2]. Погрешность соответствует стандартному отклонению результата при усреднении. По вертикальной оси отложена погрешность реконструкции. По горизонтальной оси отложено число кадров, которое обработано согласно соответствующей последовательности. Повторим, что общее число кадров одинаково для всех экспериментов и всех способов упорядочивания, изменяется лишь число кадров, после обработки которых вычисляется погрешность реконструкции.

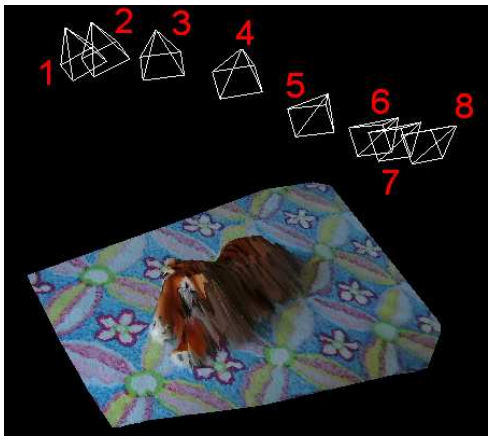
Для предлагаемого способа упорядочивания изображений относительная ошибка и ее разброс меньше, чем в случае упорядочивания, предложенного в [2], [1] особенно в начале процесса реконструкции, когда обработан мало (2-4) кадров. Это связано с тем, что эти алгоритмы выполняют первичную реконструкцию по кадрам, полученным для малой базы. Предлагаемый алгоритм сначала обрабатывает изображения, соответствующие более удаленным камерам, для которых число соответствий меньше. И так как погрешность реконструкции уменьшается с увеличением базы, то предлагаемый подход к построению последовательности реконструкции приводит к более точному результату.



(a) Пример нескольких изображений из набора восьми снимков



(b) Граф, соответствующий последовательности на Рис.4a



(c) восстановленная модель и положения камер. Числа около камер обозначают порядок, в котором изображения были получены. Найденная последовательность реконструкции: $\mathcal{S} = \{4, 7, 8, 5, 6, 3, 1, 2\}$.

Figure 4: Результаты применения предлагаемого алгоритма упорядочивания изображений

4.2 Реальные данные

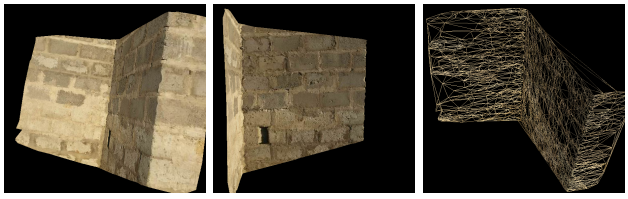
Пример последовательности фотографий, использованных для реконструкции приведен на Рис.4a. На Рис.4b приведен граф изображений. Вершины графа соответствуют изображениям, вес ребра равен определенности фундаментальной матрицы между изображениями. Сначала степень определенности фундаментальной матрицы между изображениями растет с увеличением расстояния между камерами, а потом уменьшается из-за уменьшения числа соответствий.

Сцена сначала вычисляется по паре кадров из середины (4) и конца (7) последовательности. Далее добавляется еще изображение из конца последовательности (8), так как, несмотря на близость камер в положениях 7 и 8, велика база между кадрами между 4 и 8. Далее продолжают добавляться изображения из второй половины последовательности (5,6) и лишь потом обрабатывается начало последовательности (3,2,1).

На Рис.4c представлена построенная модель с наложенной текстурой.

На Рис.5a приведена последовательность из трех фотографий, для которой последовательность реконструкции сильно влияет на погрешность. А именно, база съемки между первыми двумя кадрами значительно меньше, чем между первым (вторым) изображением и третьим. На Рис.5b представлена модель, полученная для предлагаемой последовательности реконструкции. На Рис.5c представлена модель, полученная для последовательности реконструкции, используемой в [2]. Можно видеть, что предлагаемый алгоритм выбора последовательности приводит к более точному

(a) Набор изображений



(b) Трехмерная модель (сеточная и текстурированная) с разных ракурсов, полученная в соответствии с предлагаемым алгоритмом выбора последовательности реконструкции.

(c) Трехмерная модель (сеточная и текстурированная) с разных ракурсов, полученная в соответствии алгоритмом [2] выбора последовательности реконструкции.

Figure 5: Пример набора изображений, для которого разные последовательности реконструкции приводят к существенно разной погрешности погрешности. База съемки между первыми двумя кадрами значительно меньше, чем между первым (вторым) изображением и третьим.

результату, чем в [2].

5. ЗАКЛЮЧЕНИЕ

В работе рассматривается задача последовательного построения 3d модели по фотографиям. Во-первых, вводится мера определенности фундаментальной матрицы. Полученные результаты показывают, что эта мера может служить оценкой достижимой точности реконструкции по паре кадров. Большое значение определенности фундаментальной матрицы является необходимым условием для маленькой погрешности реконструкции. Во-вторых, введенная мера используется для построения графа, вершины которого соответствуют изображениям, а вес ребра равен определенности фундаментальной матрицы между соответствующей парой изображений. Предлагается алгоритм выбора последовательности реконструкции. Полученные результаты численных экспериментов и обработка реальных изображений показывают, что предлагаемый алгоритм упорядочивания изображений приводит к более точным и стабильным результатам. Предложена априорная оценка (23) точности реконструкции по двум кадрам.

6. БЛАГОДАРНОСТИ

Работа выполнена при поддержке грантов РФФИ 09-01-92470-МНКС_а, 09-07-92000-ННС_а.

7. REFERENCES

[1] Richard I. Hartley. Euclidean reconstruction from uncalibrated views. In *Proceedings of the Second Joint European - US Workshop on Applications of Invariance in Computer Vision*, pages 237–256, London, UK, 1994. Springer-Verlag.

[2] Frederik Schaffalitzky and Andrew Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 414–431, London, UK, 2002. Springer-Verlag.

[3] P. Beardsley, A. Zisserman, and D. Murray. Sequential updating of projective and affine structure from motion, 1997.

[4] Andrew W. Fitzgibbon and Andrew Zisserman. Automatic camera recovery for closed or open image sequences. In *ECCV (1)*, pages 311–326, 1998.

[5] Bill Triggs, Philip McLauchlan, Richard Hartley, and Andrew Fitzgibbon. Bundle adjustment – A modern synthesis. pages 298–375, 2000.

[6] Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):77–116, 1998.

[7] David G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the International Conference on Computer Vision ICCV, Corfu*, pages 1150–1157, 1999.

[8] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[9] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[10] Gabriella Csurka, Cyril Zeller, Zhengyou Zhang, and Olivier D. Faugeras. Characterizing the uncertainty of the fundamental matrix. *Computer Vision and Image Understanding: CVIU*, 68(1):18–36, 1997.

[11] Chandler Davis and W. M. Kahan. The rotation of eigenvectors by a perturbation. *SIAM J.*, 7(1):1–46, march 1970.

[12] Théodore Papadopoulo and Manolis I. A. Lourakis. Estimating the jacobian of the singular value decomposition: Theory and applications. In *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part I*, pages 554–570, London, UK, 2000. Springer-Verlag.

[13] A Zisserman P A Beardsley and D W Murray. Navigation using affine structure from motion. In *Proc 3rd European Conf on Computer Vision, Stockholm*, Lecture Notes in Computer Science, pages 85–96. Springer, 1994.

[14] Richard I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In *ECCV '92: Proceedings of the Second European Conference on Computer Vision*, pages 579–587, London, UK, 1992. Springer-Verlag.

ОБ АВТОРАХ

Волегов Дмитрий Борисович - аспирант Московского Физико-Технического Института факультета Общей и Прикладной Физики. dvolegov@rambler.ru.

Юрин Дмитрий Владимирович, к.ф.-м.н., с.н.с. лаборатории Математических методов обработки изображений факультета Вычислительной математики и кибернетики Московского государственного университета им. М.В.Ломоносова. yurin_d@inbox.ru

ORDERING IMAGE SET TO RECONSTRUCT 3D MODELS

Abstract

The problem of reconstruction of 3d models from unordered image set is considered. To begin a small number of images is processed and initial model is built, then other images are processed and the model is refined. Firstly the problem of image ordering is considered. Secondly the original measure "determinacy of fundamental matrix" is introduced. This measure allows easily to estimate the precision of reconstruction without actually doing it. The results for real and synthetic sequences are presented.

Keywords: *image ordering, 3d reconstruction, fundamental matrix .*

ABOUT THE AUTHOR

Dmitriy B. Volegov is a Ph.D student at Moscow institute of Physics and Technology, Department of General and Applied Physics. His contact email is dvolegov@rambler.ru

Dmitry V. Yurin, PhD, is a senior scientist at Laboratory of Mathematical Methods of Image Processing, Chair of Mathematical Physics, Faculty of Computational Mathematics and Cybernetics, Moscow Lomonosov State University. His contact email is yurin_d@inbox.ru

НЕКООПЕРАТИВНАЯ БИОМЕТРИЧЕСКАЯ ИДЕНТИФИКАЦИЯ ПО 3D-МОДЕЛЯМ ЛИЦА С ИСПОЛЬЗОВАНИЕМ ВИДЕОКАМЕР ВЫСОКОГО РАЗРЕШЕНИЯ

А.И. Манолов, А.Ю. Соколов, О.В. Степаненко, А.С. Тумачек, А.В.Тяхт, А. К. Цискаридзе, Д.Н. Заварикин, А.А. Кадейшвили,
Компания Vocord

Аннотация

Получены результаты по распознаванию лиц, основанные на 3D реконструкции без использования какой-либо структурированной подсветки. 3D реконструкция основана на использовании камер высокого разрешения. Вероятность распознавания составляет 92-98%.

Ключевые слова: 3D реконструкция, 3D распознавание лиц.

1. ВВЕДЕНИЕ

Системам распознавания лиц, основанным на двумерных изображениях, присущи определенные недостатки. Такие системы чувствительны к изменениям яркости. Свет, собранный с лица, является функцией геометрии лица, отражательной способности лица, свойствами источника света и свойствами камеры. С учетом этого, сложно создать модель лица, которая учитывает одновременно все вариации этих факторов. В трехмерном распознавании изменения в освещенности влияют лишь на текстуру на лице, в то время как реконструкция поверхности сохраняет свои свойства [1].

Другим отличительным фактором между 2D и 3D распознаванием является эффект изменения позы лица (ракурса камеры). Чтобы компенсировать этот эффект, в двумерном распознавании использовалась трансформация изображения к каноническому положению [2]. Тем не менее, эффективность данного подхода зависит от точности расположения антропометрических точек на лице, и плохо работает при затенении части лица. Более того, данная задача в двумерном виде не может быть решена теоретически строго из-за свойства перспективной проекции двумерных изображений. Чтобы обойти эти проблемы, было предложено сохранять в базе данных различные виды лица под разным ракурсом [3]. Такой подход, тем не менее, требует сохранения большого количества двумерных изображений и снижает производительность системы распознавания. Были предложены статистические модели, учитывающие изменения позы лица [4, 6], а а также обобщенные модели лица [5]. С использованием 3D изображений, интерполяция положения может быть сведена к повороту восстановленной 3D модели лица к новой позе. Было предложено использовать 3D модель, реконструированную из двумерных нефронтальных изображений, и воссоздать фронтальный вид лица для 2D распознавания [7].

Другой проблемой 2D распознавания, связанной с вариацией положения лица, является то, что физические размеры лица нельзя полностью восстановить из 2D изображений. Размер лица на двумерном изображении существенно зависит от расстояния между лицом и камерой. В 3D распознавании физические размеры лица известны и могут использоваться явно для распознавания.

Традиционные системы 2D распознавания используют участки изображения с высоким контрастом, таких как глаза, рот, нос, границы лица и плохо используют информацию в областях низкого контраста – на щеках, лбу, подбородке и др. В противоположность 2D, 3D распознавание включает для анализа и информацию на данных участках. Более того, информация о форме поверхности, сосредоточенная на лобных костях и других слабо деформируемых участках, мало подвержена изменениям при различных выражениях лица (как например, улыбка и др.).

3D распознавание, тем не менее, не является полностью идеальным. Например, освещение не является проблемой на этапе 3D распознавания, но оно может отрицательно повлиять на результат 3D реконструкции формы лица. В зависимости от алгоритма реконструкции, некоторые части лица (например, засвеченные участки, либо области с очень низким контрастом) могут внести пробелы в реконструкцию, или артефакты (выбросы на поверхности).

Другим недостатком 3D распознавания также является высокая стоимость оборудования. Кроме этого, 3D распознавание требует больших вычислительных ресурсов, чем 2D.

Наконец, до настоящего времени, по-видимому, недостаточное внедрение 3D систем распознавания было связано и с отсутствием на рынке видеосенсоров высокого разрешения. Разработанные компанией Vocord доступные по стоимости видеокамеры NetCam и проводимые исследования в области 3D-распознавания с использованием этих камер могут существенно повлиять на внедрение 3D распознавания в России.

Среди различных подходов 3D распознавания можно выделить три основных направления: анализ формы 3D поверхности лица, статистические подходы, и использование параметрической модели лица.

Методы, основанные на анализе формы, используют непосредственно геометрию поверхности, которая описывает лицо. Эти подходы можно классифицировать на те, которые используют либо локальные, либо глобальные свойства поверхности (например, кривизну), на подходы, основанные на профили линий, либо на методы, которые используют метрику расстояний между двумя поверхностями.

В работе [8] было предложено использовать кривизну поверхности для сегментации поверхности лица на признаки, которые могут быть использованы для сравнения поверхностей. Другой подход основан на 3D дескрипторах поверхности лица в терминах средней и гауссовой кривизны [9], либо в терминах расстояний и отношении углов между характерными точками поверхностей [10].

Другим локально-ориентированным методом является подход, использующий точки-сигнатуры [11]. Идея метода заключается в формировании представления-описания точки по соседним точкам вокруг заданной точки поверхности. Эти сигнатуры точек используются для сравнения поверхностей.

Было также предложено удалять из рассмотрения те части поверхности, которые подвержены изменениям в результате изменения мимики на лице. Только жесткие части лица являются информацией для распознавания. Такой подход применен в работе [12]. Кроме того, в этой же работе дополнительно с 3D информацией 2D использовалась текстурная информация на участках лица.

Существуют также и гибридные методы, основанные на объединении локальной информации о поверхности в виде локальных моментов, с глобальной трехмерной сеткой, описывающей все лицо [13].

Глобальными методами являются такие методы, которые задействуют все 3D лицо как информацию на входе системы распознавания. В одной из ранних систем лицо выравнивалось на основе его зеркальной симметрии [14], после чего выделялись и сравнивались профили лица вдоль плоскости выравнивания. В работе [15] было предложено сравнение лиц на основе максимального и минимального значений и направления кривизны профилей.

Другой подход основан на методе сравнения расстояний между поверхностями для распознавания. В работах [16] было предложено использовать метрику наименьших расстояний для вычисления расстояния между поверхностями. Было также предложено измерять расстояние не только между поверхностями, но и текстурой на поверхности [17]. Существенным ограничением данных методов является то, что лицо не может деформироваться и его поверхность является жесткой.

Другим подходом является извлечение и анализ трехмерных профилей и контуров, выделенных на лице [18].

Статистические методы, например, метод главных компонент (principal component analysis - PCA) широко использовались ранее в 2D распознавании. Недавно метод PCA был реализован для 3D распознавания [19, 20, 21]. PCA был расширен одновременно на комбинацию карты глубины и цвета [22]. Альтернативным для PCA является метод линейного дискриминантного анализа [23]. В отличие от PCA, в линейном дискриминантом анализе один объект (заданный человек) задается не одним лицом, а набором моделей (3D лиц).

До сих пор все описанные статистические методы не учитывали эффект изменения формы поверхности лица из-за его различной мимики. Для минимизации этого эффекта были разработаны подходы, основанные на инвариантных изоморфных преобразованиях. Такими преобразованиями являются те, которые не изменяют расстояния между двумя заданными точками на лице под действием мимических изменений формы лица. В работе [24, 25] было предложено для этого использовать преобразование формы лица к каноническому виду. Подобные методы использовали PCA на заключительной стадии распознавания, который применялся к канонической форме лица.

Существуют также способы распознавания, основанные на параметрических моделях лица. Ключевая идея распознавания по моделям основана на так называемых параметрических 3D моделях. В данном подходе форма лица контролируется набором параметров модели. Эти коэффициенты описывают 3D форму лица и также могут задавать цвет (текстуру) на ее поверхности. Подобная модель затем проецируется на двумерные изображения, из которых определяются параметры коэффициентов модели для данного изображения [26]. Недостатком метода является высокая вычислительная сложность и чувствительность к начальной инициализации параметров модели. Для преодоления этих сложностей были придуманы модели, состоящие из независимых участков [27].

Было также предложено использовать трехмерную поверхность среднего лица, которая с использованием анатомических антропометрических точек на лице деформируется до заданной трехмерной поверхности [28]. Параметры деформации использовались в качестве отличительных признаков для данного лица.

К сожалению, как отмечается в работе [17] прямое сравнение различных 3D алгоритмов распознавания по их эффективности крайне затруднено. Связано это с отсутствием общей стандартизированной базы данных трехмерных лиц и различием точности моделей 3D изображений. Тип представления и пространственное разрешение 3D данных существенно различается среди публикаций. По-видимому, стандарты на 3D изображения и стандартизированные базы изображений появятся в самое ближайшее время.

Среди наиболее современных методов, согласно обзору в [29], выделяются глобальные методы (вероятность распознавания 90-96%) и статистические методы (93-100%). Параметрические модели лица характеризуются вероятностью порядка 88-96%.

Среди рассматриваемых моделей, которые предложены до настоящего времени для компенсации мимики на лице, изоморфические модели (приведение к каноническим формам лица) являются, видимо, наиболее перспективными [24,25].

Исходя из анализа современного состояния проблемы компанией Vocord было принято решение использовать и разработать систему, основанную именно на 3D реконструкции лица с последующим его распознаванием, обеспечивающей достаточно высокие показатели, на базе выпускаемых камер Netcam высокого разрешения.

2. 3D РЕКОНСТРУКЦИЯ ПОВЕРХНОСТИ ЛИЦА

Реконструкция выполняется с использованием камер высокого разрешения собственной разработки Vocord NetCam. Данные камеры обеспечивают разрешение до 2048 x 1576 пикселов, частота кадров до 12 кадров/сек.

Ри

с.1 Камеры Vocord NetCam

Особенностью алгоритма является отсутствие какого либо лазерного или другого дополнительного света, направленного на лицо человека (структурированной подсветки). Реконструкция проводится в условиях минимальной освещенности (400-600 люкс) на расстоянии до 2.5 м от камер. Использование камер разрешением 5 Мегапикселов, выпуск которых намечен в скором плане компании, позволит осуществить реконструкцию с теми же параметрами на расстоянии 3-4 м до объективов.

Особенностью алгоритма является также возможность использовать реконструкцию по многим камерам. Например, оптимальным является использование трех

синхронизированных во времени камер, разнесенных друг относительно друга на расстояние порядка 20 см между камерами. Такой режим улучшает среднестатистические показатели реконструированной поверхности.

Кроме того, помимо высокого исходного разрешения камер, разработанный алгоритм поиска точек сопряжения изображений на стереопарах использует субпиксельную точность до ¼ пиксела. В результате, суммарная эффективная погрешность, обусловленная дискретностью сенсоров камер, имеет порядок 0.3-0.5 мм, что достаточно для дальнейшего высококачественного распознавания.

3D реконструкция модели лица осуществляется поэтапно:

- построение поля 3D точек;
- предварительная фильтрация точек;
- построение и окончательной фильтрация триангулированной модели лица (рис 2, 3);
- совмещение нескольких поверхностей лица человека, полученных в разные моменты времени (видеоряд), в одну общую объединенную 3D модель (рис.4, 5).

(а)

(б)

Рис. 2 Пример 3D реконструкции лица. (а) – лицо с текстурой, (б) – трангулированная модель.

В настоящее время, 3D реконструкция с использованием 4ех ядерного процессора 2.9 ГГц занимает от 2 до 10 сек, в зависимости от настроек параметров системы. Компания продолжает работы по оптимизации алгоритма 3D реконструкции, в частности, с использованием графических плат акселераторов на платформе NVIDIA CUDA.

3. 3D РАСПОЗНАВАНИЕ

Распознавание выполняется в два этапа:

(1) быстрое сравнение по базе данных с использованием вектора признаков лица небольшой длины;

(2) уточнение результата по отобранным нескольким наиболее похожим изображениям из базы данных.

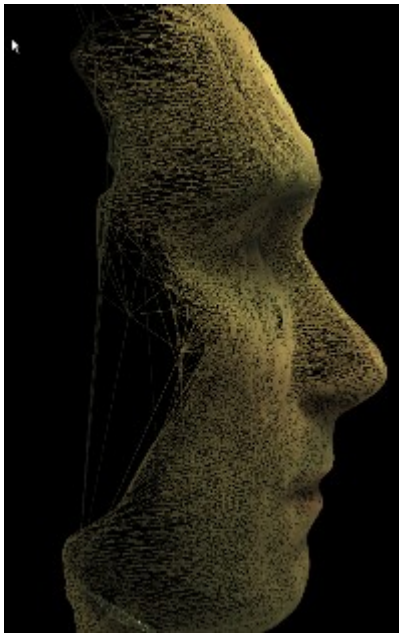
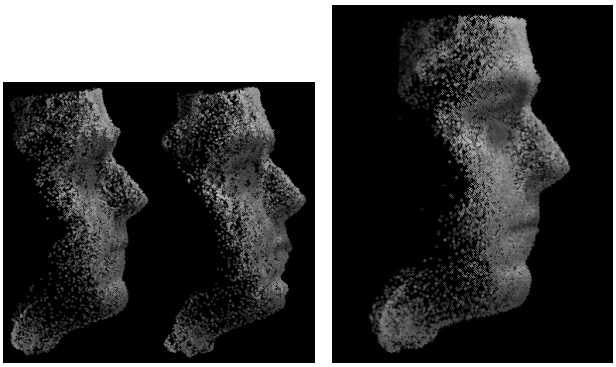
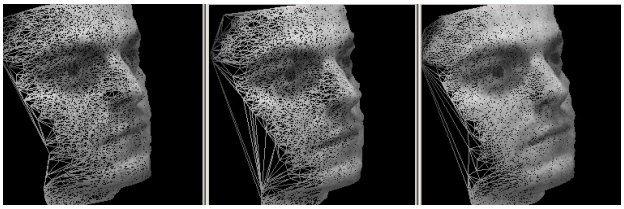


Рис.3 Пример 3D реконструкции лица (вид в профиль поперек к направлению на камеры).



(а) (б)

Рис. 4. Обработка 3D видео. (а) – 3D модели лица в разные моменты времени, (б) – после совмещения 3D моделей в общую модель.



(а) (б) (с)

Рис. 5. Обработка 3D видео. (а), (б) – 3D модели в разные моменты времени, (с) – после совмещения 3D моделей в общую модель.

Система 3D распознавания, разработанная в Vocord, также может использовать приведение 3D поверхности к форме, более устойчивой при мимических искажениях (Рис.6) [24, 25].

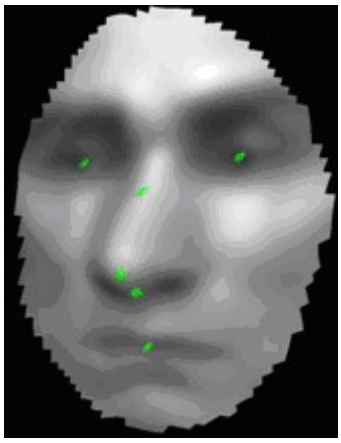


Рис. 6. Использование геодезического преобразования к маске лица

4. РЕЗУЛЬТАТЫ

Тестирование проводилось на базе данных 100 лицах, из которых 50 чел были внесены в базу данных, 50 чел были незнакомыми.

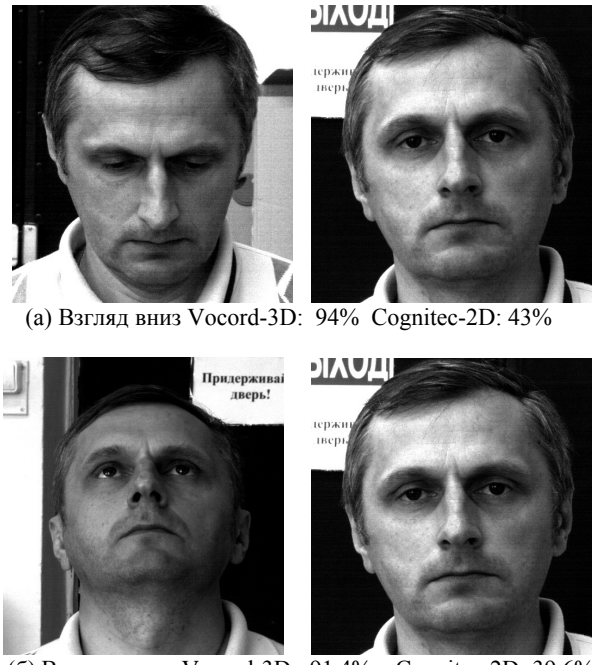
Для неподвижно стоящих людей вероятность распознавания составила P=98% при FAR=FRR=2 %. Для людей, движущихся навстречу камеры со скоростью примерно 3 км/ч, были продемонстрированы результаты FAR=FRR=7%, вероятность распознавания P=93%.

Разработанный алгоритм также устойчив к поворотам лица в пределах +50/-30 градусов вверх-вниз, и примерно +/- 30 градусов влево –вправо, см. Рис.7, 8, таблицу 1. На тестовых изображениях диапазон максимальных углов 3D распознавания выше диапазона 2D распознавания.

Таблица 1. Сравнение Vocord-3D и Cognitec-2D [30]. Человек считается распознанным, если мера сходства более 50%.

Угол наклона головы, град	**Vocord 3D %**	**Cognitec 2D (Ver. 8.1, algorithm B.4) %**
Вниз		
30	94	43.3
40	32	8
Вверх		
45	96	99
50	91.4	39.6

Рис. 7 Мера сходства лица с 3D моделью того же человека, хранимой в базе данных, в зависимости от положения головы.



(а) Взгляд вниз Vocord-3D: 94% Cognitec-2D: 43%

(б) Взгляд вверх Vocord-3D: 91.4%, Cognitec-2D: 39.6%

Рис. 8 Сравнение эффективности с двумерным алгоритмом распознавания Cognitec (Version 8.1, algorithm B4) [30]. Слева – тестируемое изображение, справа – фото из базы

данных. Человек считается распознанным, если мера сходства более 50%.

5. ЗАКЛЮЧЕНИЕ

Компанией Vocord разработан прототип системы 3D распознавания с использованием камер высокого разрешения, без использования какой-либо дополнительной подсветки. Данные преимущества позволяют использовать разработанную систему для некооперативной идентификации людей на основе 3D моделей. Показаны высокие характеристики распознавания с вероятностью 92-98%. На некоторых изображениях (большой угол поворота головы, см. Табл.1) 3D алгоритм имеет преимущество по сравнению с 2D распознаванием.

6. СПИСОК ЛИТЕРАТУРЫ:

[1] Hesher, C., Srivastava, A., and Erlebacher, G. (2003). A novel technique for face recognition using range imaging. *International Symposium on Signal Processing and Its Applications*, стр 201–204.

[2] Kim, T. and Kittler, J. (2005). Locally linear discriminant analysis for multimodally distributed classes for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):318–327.

[3] Li, Y., Gong, S., and Lidell,H. (2000). Support vector regression and classification based multiview face detection and recognition. *International Conference on Face and Gesture Recognition*, стр 300–305.

[4] Lanitis, A., Taylor, C., and Cootes, T. (1995). Automatic face identification systemusing flexible appearance models. *Image and Vision Computing*, 13(5):393–401.

[5] Prince, S. and Elder, J. (2006). Tied factor analysis for face recognition across large pose changes. *British Machine Vision Conference*.

[6] Gross, R., Matthews, I., and Baker, S. (2002). Eigen light-fields and face recognition across pose. *International Conference on Automatic Face and Gesture Recognition*.

[7] Blanz, V., Grother, P., Phillips, J., and Vetter, T. (2005). Face recognition based on frontal views generated from non-frontal images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, стр 454–461.

[8] Gordon, G. (1992). Face recognition based on depth and curvature features. *IEEE Computer Vision and Pattern Recognition (CVPR)*, стр 808–810.

[9] Moreno, A., Sanchez, A., Velez, J., and Diaz, F. (2003). Face recognition using 3D surface extracted descriptors. In *Irish Machine Vision and Image Processing Conference*.

[10]. Lee, Y., Song, H., Yang, U., Shin, H., and Sohn, K. (2005). Local feature based 3D face recognition. *International Conference on Audio- and Video-based Biometric Person Authentication*, стр 909–918.

[11] Chua, C. and Jarvis, R. (1997). Point signatures - a new representation for 3D object recognition. *International Journal of Computer Vision*, 25(1):63–85.

[12]. Wang, Y., Chua, C., and Ho, Y. (2002). Facial feature detection and face recognition from 2D and 3D images. *Pattern Recognition Letters*, 23:1191–1202.

[13]. Xu, C.,Wang, Y., Tan, T., and Quan, L. (2004). Automatic 3D face recognition combining global geometric features with local shape variation information. In *International Conference on Automated Face and Gesture Recognition*, pages 308–313.

[14] Cartoux, J., LaPreste, J., and Richetin, M. (1989). Face authentication or recognition by profile extraction from range images. *Workshop on Interpretation of 3D Scenes*, стр 194–199.

[15] Tanaka, H., Ikeda, M., and Chiaki, H. (1998). Curvature-based face surface recognition using spherical correlation principal directions for curved object recognition. *International Conference on Automated Face and Gesture Recognition*, pages 372–377.

[16] Ackermann, B. and Bunke, H., t. . C. (2000). *15th International Conference on Pattern Recogni- tion*, стр 809–813.

[17] Papatheodorou, T. and Rueckert, D. (2004). Evaluation of automatic 3D face recognition using surface and texture registration. *International Conference on Automated Face and Gesture Recognition*, стр 321–326.

[18]. Nagamine, T., Uemura, T., and Masuda, I. (1992). 3D facial image analysis for human identification. *International Conference on Pattern Recognition*, pages стр 324–327.

[19] Mavridis, N., Tsalakanidou, F., Pantazis, D., Malasiotis, S., and Strintzis, M. (2001). The hiscore face recognition application: Affordable desktop face recognition based on a novel 3D camera. *International Conference on Augmented Virtual Environments and 3D Images*, стр 157–160.

[20] Chang, K., Bowyer, K., and Flynn, P. (2003). Face recognition using 2D and 3D facial data. *Multimodal User Authentication Workshop*, стр 25–32.

[21]. Papatheodorou, T. and Rueckert, D. (2005). Evaluation of 3 D face recognition using registration and PCA. *Audio- and Video-based Biometric Person Authentication*, стр 997–1009.

[22] Tsalakanidou, F., Tzocaras, D., and Strintzis, M. (2003). Use of depth and colour eigenfaces for face recognition. *Pattern Recognition Letters*, 24:1427–1435.

[23]. Gokberk, B., Salah, A., and Akarun, L. (2005). Rank-based decision fusion for 3D shapebased face recognition. In *International Conference on Audio- and Video-based Biometric Person Authentication*, стр 1019–1028.

[24]. Bronstein, A., Bronstein,M., and Kimmel, R. (2003). Expression-invariant 3D face recognition. *International Conference on Audio- and Video-Based Person Authentication*, стр. 62– 70.

[25]. Bronstein, A., Bronstein, M., and Kimmel, R. (2005). Three-dimensional face recognition. *International Journal of Computer Vision*, 64:5–30.

[26] Blanz, V., Romdhani, S., and Vetter, T. (2002). Face identification across different poses and illuminations with a 3D morphable model. In *International Conference on Automatic Face and Gesture Recognition*, стр 202–207.

[27]. Huang, J., Heisele, B., and Blanz, V. (2003). Component-based face recognition with 3D morphable models. In *International Conference on Audio- and Video-Based Person Authentication*.

[28] Passalis, G., Kakadiaris, I., Theoharis, T., Toderici, G., and Murtuza, N. (2005). Evaluation of 3D face recognition in the presence of facial expressions: an annotated deformable model approach. стр. 1022 – 1029.

[29]. Papatheodorou, T. and Rueckert, D. (2007). 3D face recognition, книга Face recognition, Vienna, Austria, 2007, стр 417-423.

[30] http://www.cognitec-systems.de/FaceVACS-SDK.19.0.html

Программно-аппаратный комплекс подготовки и контроля цифровых фотографий для биометрических документов

Software-Hardware System for Digital Face Imagery Acquisition and Testing for Biometric Documents

Сергей Каратеев, Ирина Бекетова, Михаил Ососков, Юрий Визильтер, Александр Бондаренко, Сергей Желтов

Государственный научный центр РФ
Государственный научно – исследовательский институт авиационных систем (ГосНИИАС)

ABSTRACT (ENG)

The software-hardware system for digital face imagery acquisition and testing for requirements of ISO/IEC FCD 19794-5 standard is described. System contains following algorithmic modules: face detector; color and intensity characteristics estimator; opened/closed eyes detector; glasses detector, reflexes, shines and shadows detector; face features detector (nose, brows, mouth); face slope/rotation detector. Following topics are presented in a paper: system structure, functions, hardware, algorithms, user's interface and samples of system operation results.

ABSTRACT (RUS)

Приведено описание программно-аппаратного комплекса подготовки и контроля цифровых фотографий для биометрических документов на соответствие требованиям стандарта ИСО/МЭК 19794-5-2006. Алгоритмическое обеспечение системы включает: детектор лица, модуль оценки яркостных и цветовых характеристик; детектор открытых/закрытых глаз; детектор очков, детектор бликов и теней; детектор основных элементов лица (рот, нос, брови), детектор поворотов и наклонов головы. В статье представлены: общая структура комплекса, функции, аппаратное и алгоритмическое обеспечение, интерфейс пользователя и примеры работы комплекса.

Keywords: *Biometric documents, digital face imagery, face detection, facial features detection, boosting.*

1. ВВЕДЕНИЕ

В настоящее время, в большинстве развитых стран ускоренными темпами идет внедрение персональных проездных документов с биометрической информацией, записанной по единому стандарту. Для обеспечения согласованности национальных стандартов цифровых фотографий международной организацией по стандартизации были выработаны рекомендации ISO/IEC FCD 19794-5**.** В России стандартом, определяющим требования к изображениям лиц для биометрических документов, является ГОСТ ИСО/МЭК 19794-5-2006. Введение стандарта на цифровые фотографии определяет необходимость автоматизации операций контроля качества изображений лиц, как непосредственно в процессе получения этих изображений, так и на любом этапе подготовки паспортных, визовых и иных документов.

Представленный в данной статье программно-аппаратный комплекс предназначен для автоматизации процесса получения цифровых фотографий, удовлетворяющих основным требованиям и рекомендациям ГОСТ ИСО/МЭК 19794-5-2006**.** Комплекс обеспечивает получение цифровых фотографий лица, а также оценку в реальном времени основных характеристик изображения и параметров лица, что позволяет оператору с минимальными усилиями, не превышающими усилия, необходимые для получения обычной качественной фотографии лица, получать цифровые фотографии лиц, гарантированно удовлетворяющие требованиям данного ГОСТ. Кроме того, описанный комплекс может быть использован для контроля параметров фотографий лиц, полученных от других источников изображений – как в цифровом, так и в бумажном виде, предоставляя возможность оценки пригодности фотографий для последующей биометрической обработки.

2. ТРЕБОВАНИЯ К ЦИФРОВЫМ ФОТОГРАФИЯМ ДЛЯ БИОМЕТРИЧЕСКИХ ДОКУМЕНТОВ

ГОСТ ИСО/МЭК 19794-5-2006 определяет следующие основные требования к цифровым изображениям лица и форматам сохранения данных:

- Минимальный размер фотографии 525*420 пикселей.
- Изображение лица на фотографии должно быть фронтальным и не иметь отклонения относительно основных осей более чем на 5 градусов.
- Соотношение ширины головы к ширине фотографии должно быть не менее 7:5 и не более чем 2:1.
- Расстояние от нижней границы фотографии до горизонтальной линии, проходящей через центры глаз . должно составлять от 50 до 70% от высоты полного изображения
- Площадь лица на фотографии должна составлять от 70% до 80% от площади фотографии.
- Цвет и яркость фона должны обеспечивать надежное определение контура головы.
- На изображениях лиц не должно быть световых бликов и сильного затенения.

- На фоне не должно быть теней от головы или каких-либо предметов.
- На изображениях лица не должно быть закрытых глаз, предметов, закрывающих глаза и лицо или искажающих черты лица.

Таким образом, проверка цифровых изображений лица на соответствие требованиям ГОСТ является достаточно нетривиальной задачей. Здесь недостаточно визуального анализа фотографии для принятия решения об её пригодности для использования в документах, удостоверяющих личность. Поэтому при создании комплекса биометрической регистрации изображений лиц было разработано специализированное программное обеспечение, автоматизирующее процесс создания и контроля фотографий, удовлетворяющих требованиям ГОСТ.

3. АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ КОМПЛЕКСА

На основании измеренных и рассчитанных характеристик изображения лица алгоритмическое обеспечение осуществляет диагностику наличия и причин отклонений от требований ГОСТа и вывод сообщений об этих отклонениях и их возможных причинах.

Для получения оценок основных параметров изображения лица в автоматическом режиме решаются следующие задачи:

- автоматическое обнаружение лица на изображении;
- автоматическое определение контура и оценка параметров лица на изображении;
- автоматическое обнаружение глаз на изображении и оценка координат центров зрачков;
- обнаружение бликов и областей сильной затенённости на изображении лица;
- формирование фронтальных и условно-фронтальных цветных и монохромных изображений заданного размера для печати фотографий;
- формирование изображений для систем обмена биометрическими данными.

Алгоритм автоматического обнаружения области лица базируется на известном алгоритме машинного обучения Adaboost [1,2]. В данном алгоритме слабые (week) классификаторы построены на основе фильтров Хаара, однако отклик формируется с использованием аппроксимации распределения вероятностей амплитуды откликов. Аппроксимация распределения вероятностей откликов представляется в виде гистограммы, которая строится по взвешенным примерам, в результате чего, обучение слабых классификаторов проводится на подвыборках одного и того же фиксированного размера, но имеющих различные распределения обучающих изображений. В процессе обучения, при формировании каскадного классификатора для каждого последующего классификатора признаковое пространство сокращается за счёт устранения признаков, на которых построены предыдущие классификаторы. Классификатор представляет собой линейную комбинацию слабых классификаторов, количество которых варьируется от 5 до 75. В свою очередь, каскадный классификатор представляет собой последовательность из 8 классификаторов. Математическое моделирование показало [3], что при работе по случайно выбранной совокупности тестовых изображений алгоритм автоматического обнаружения изображений лица обеспечивает вероятность правильного обнаружения не менее 0.95 при вероятности ложного обнаружения лица не более 0.01.

Алгоритм автоматического определения контура лица на изображении построен на использовании информации об оттенках кожи человека. Областью интереса алгоритма является фрагмент изображения, классифицированный как лицо каскадным классификатором AdaBoost. Каждому пикселю цветного RGB-фрагмента изображения ставится в соответствие вектор параметров цвета (H,S,V) в цветовом пространстве HSV. Распределение оттенков кожи представлено бинарной картой кожи, хранящейся в структуре данных типа LUT-таблицы поиска. Сегментация фрагмента изображения выполняется путём проверки принадлежности параметров цвета пикселей к кластеру модели оттенков кожи с помощью операции поиска по LUT-таблице. Область изображения кожи формируется из пикселей, векторы параметров которых вошли в один из кластеров. Для формирования одной однородной области пикселей по цвету соответствующих оттенкам кожи и удаления мелких областей, линий и отдельных пикселей к изображению применяются такие операции математической морфологии, как дилатация и эрозия. Линии контура лица формируются с помощью применения алгоритма сплайн-интерполяции к координатам граничных пикселей области кожи лица. Размеры прямоугольника, в который вписаны линии контура лица, являются линейными размерами изображения лица.

Задача обнаружения изображений глаз решается как задача поиска и распознавания на цифровом изображении лица локальных областей, обладающих специфическими характерными для изображений глаз параметрами. Областью интереса алгоритма является часть изображения лица, представляющая собой область ожидаемого расположения глаз. Исходя из статистической модели проводится предварительная оценка ожидаемых положения и размеров глаз, благодаря чему существенно возрастает вычислительная эффективность алгоритма. Алгоритм автоматического обнаружения области глаз помощью каскадного классификатора Adaboost определяет координаты глаз в пределах радужной оболочки. Для точной локализации изображений глаз производится поиск координат центров зрачков с использованием операций морфологической фильтрации. Морфологический фильтр выделяет изображение зрачка и радужной оболочки глаз, устраняя при этом шумовые помехи и артефакты изображения, например блики. Координаты центров зрачков определяются путём свёртки изображения с круговым фильтром, подчёркивающим форму зрачка. Проведённое математическое моделирование показало, что при работе по случайно выбранной совокупности тестовых изображений разработанный алгоритм автоматического обнаружения и локализации изображений глаз обеспечивает вероятность правильного обнаружения и локализации не менее 0.95 при вероятности ложного обнаружения глаз не более 0.01. Пример работы алгоритмов обнаружения изображения лица, выделения контура лица и обнаружения глаз представлен на Рис.3.

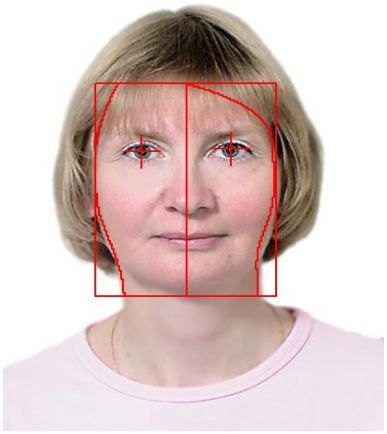


Рис. 3. Пример работы алгоритмов обнаружения лица и обнаружения глаз.

После проведения основных операций над изображением лица выполняется анализ полученных данных на соответствие ГОСТ. Для этого производится расчёт оценок характеристик изображения, производных от геометрических параметров лица, и проверка наличия артефактов на самом изображении. Выполняются следующие операции:

– определение оси симметрии лица;
– определение центровки изображения лица;
– определение угла поворота лица;
– определение угла наклона лица;
– обнаружение очков.

На заключительном этапе интерпретации результатов проводится проверка оцененных параметров изображения на соответствие требованиям стандарта. В случае несоответствия вычисленных параметров изображения требованиям стандарта выдаются рекомендации по изменению условий съемки. В случае формирования условно-фронтальных изображений система выполняет необходимые повороты и перемасштабирование изображения.

4. КОНСТРУКТИВНЫЕ ОСОБЕННОСТИ ПРОГРАММНО- АППАРАТНОГО КОМПЛЕКСА

Одним из основных приоритетов при выборе конструкции комплекса было требование создания наиболее простой, максимально дешёвой конструкции, состоящей из широко доступных готовых покупных компонент.

Созданный программно-аппаратный комплекс включает:

- Персональный компьютер;
- Цифровой фотоаппарат;
- Источник освещения;
- Специальный штатив для крепления фотоаппарата и источника освещения;
- Планшетный сканер;
- Специализированное программное обеспечение.

Комплекс обеспечивает выполнение следующих основных функций:

- Захват (оцифровка) и отображение на мониторе последовательности изображений лица, получаемых от цифрового фотоаппарата в реальном времени;
- Сохранение изображений на жестком диске компьютера;
- Загрузка и отображение изображений с жесткого диска компьютера;
- Обнаружение изображений лиц, близких к фронтальному положению;
- Обнаружение глаз, определение контура лица, вычисление осей симметрии;
- Определение центровки изображения лица;
- Определение размеров изображения головы;
- Определение углов наклона и поворота головы;
- Обнаружение очков на изображении;
- Оценка качества изображения – наличие теней, бликов, оценка цвета, яркости и текстуры фона;
- Сравнение измеренных и вычисленных параметров изображения лица с требованиями стандартов;
- Индикация результатов сравнения в виде пиктограмм и текстовых сообщений;
- Выбор изображения, удовлетворяющего требованиям стандартов (автоматически или вручную);
- Вывод изображения на печать в заданном формате.

При установке системы предлагается выбор используемого устройства видеоввода. В качестве таких устройств могут использоваться цифровой фотоаппарат, имеющий программный интерфейс с компьютером, или сканер. Кроме этого, в качестве источника данных может использоваться любой внешний носитель информации, содержащий массивы цифровых фотографий в форматах BMP или JPEG. На Рис.4 представлен общий вид комплекса.

Рис.4. Программно-аппаратный комплекс в сборе.

Программное обеспечение работает под управлением ОС Windows 2000/XP. Интерфейс программы представляет собой диалоговое окно, в котором помимо изображения текущей фотографии также отображаются результаты проверки требований к изображению лица в виде пиктограмм и текстовой информации. Если полученное изображение имеет отклонения от норм ГОСТ, оператор получает визуальное и звуковое оповещение. При этом изображения на пиктограммах и соответствующие текстовые сообщения подсказывают ему причину ошибки. Каждая из пиктограмм, имеющихся в окне программы, соответствует одному из приведенных выше требований ГОСТ по характеристикам изображения лица и фотографии. Результаты проверки отображаются в виде текстовых сообщений в специальном окне. Кроме этого, для каждой обработанной фотографии

программа сохраняет результаты всех проверок, выполненных в процессе обработки.

5. ЗАКЛЮЧЕНИЕ

В статье представлен программно-аппаратный комплекс, предназначенный для автоматизации процесса получения цифровых фотографий, удовлетворяющих основным требованиям и рекомендациям ГОСТ ИСО/МЭК 19794-5-2006**.** Комплекс обеспечивает получение цифровых фотографий лица как полно фронтальных так и условно фронтальных, а также оценку в реальном времени основных характеристик изображения и параметров лица, подготовку изображений к печати с заданными размерами и разрешением, формирования изображений в биометрическом формате обмена данными, сохранения изображений в различных графических форматах. Кроме того, описанный комплекс может быть использован для оценки и контроля параметров фотографий лиц, полученных от других источников изображений как в цифровом, так и в бумажном виде.

Комплекс обеспечивает качество изображения, близкое к профессиональному, естественную цветопередачу, малый вес, мобильность, низкую стоимость, использование стандартных покупных компонентов, простоту и быстроту сборки/разборки, простоту и удобство использования.

Специализированное программное обеспечение позволяет оператору в реальном времени управлять процессом съемки для получения фотографии, удовлетворяющей требованиям ГОСТа. Наблюдая за процессом съёмки при помощи наглядного графического интерфейса системы, оператор в режиме реального времени видит изображение будущей фотографии с указанием возникающих ошибок, что позволяет оперативно корректировать условия съемки и предотвращать отклонения от стандарта.

Программно-алгоритмическое обеспечение системы построено по модульно-иерархическому принципу. Основными этапами анализа изображения являются: первичный анализ изображения (определение основных цвето-геометрических параметров фотографии лица) и определение соответствия фотографии требованиям ГОСТа. Используемая в системе процедура обнаружения лица построена на основе последовательного использования алгоритма цветовой сегментации и алгоритма классификации на основе метода адаптивного усиления слабых классификаторов Adaboost. Разработанный для данного комплекса модифицированный алгоритм Adaboost отличается от классического варианта использованием аппроксимации распределения вероятностей откликов слабых классификаторов вместо пороговой решающей функции. Данный модифицированный алгоритм Adaboost используется также для обнаружения глаз.

6. ЛИТЕРАТУРА

[1] Freund, Y., Schapire, R. A Short Introduction to Boosting. Journal of Japanese Society for Artificial Intelligence, 14(5): 771-780, September, 1999.

[2] Viola, P., Jones, M. Robust Real Time Object Detection. IEEE ICCV Workshop Statistical and Computational Theories of Vision, July 2001.

[3] Бекетова И.В., Каратеев С.Л., Визильтер Ю.В., Бондаренко А.В., Желтов С.Ю. Автоматическое обнаружение лиц на цифровых изображениях на основе метода адаптивной классификации AdaBoost // Вестник компьютерных и информационных технологий. 2007. №8. С.2-6.

Об авторах

Каратеев Сергей Львович, начальник сектора лаборатории компьютерного машинного зрения ГосНИИАС. e mail goga@gosniias.ru

Бекетова Ирина Валентиновна, ведущий инженер лаборатории компьютерного машинного зрения ГосНИИАС. e mail irus@gosniias.ru.

Ососков Михаил Владимирович начальник сектора лаборатории компьютерного машинного зрения ГосНИИАС. e mail mvo@gosniias.ru

Визильтер Юрий Валентинович, кандидат технических наук, начальник лаборатории компьютерного машинного зрения ГосНИИАС e mail viz@gosniias.ru

Бондаренко Александр Викторович, кандидат физико-математических наук, директор филиала ФГУП ГосНИИ Авиационных систем ЦОД.

Желтов Сергей Юрьевич, доктор технических наук, профессор, член-корреспондент РАН, Генеральный директор ФГУП ГосНИИ Авиационных систем (ГосНИИАС).

Алгоритм распознавания людей в видеопоследовательности на основе случайных патчей

Вадим Конушин, Глеб Кривовязь, Антон Конушин
Лаборатория Компьютерной Графики и Мультимедиа, МГУ им. М.В.Ломоносова, Москва, Россия
{vadim, gleb.krivovyaz, ktosh}@ graphics.cs.msu.ru

Аннотация

Распознавание людей по видео является широко востребованной задачей. Из-за того, что алгоритмы распознавания людей по лицу до сих пор не дают приемлемого качества распознавания в случае низкого разрешения изображения/видео, а также произвольного освещения, в последнее время всё большее распространение находит распознавание людей, учитывающее одежду человека.

В данной статье предложен новый алгоритм распознавания человека по видео, основанный на извлечении из видео большого количества случайных патчей. В отличие от большинства современных алгоритмов, предложенный подход не опирается на маску, полученную с помощью методов вычитания фона, из-за чего он способен работать на видео с произвольным сложным фоном.

Ключевые слова: *Распознавание человека по видео, машинное обучение*

1. ВВЕДЕНИЕ

Распознавание людей является одной из самых бурно развивающихся областей компьютерного зрения. Во многом это связано с тем, что данная область имеет большое количетсво применений на практике: в охранных системах, для рекламы и пр.

Одними из самых надеждых считаются методы распознавания по отпечаткам пальцев или по радужке глаза. Однако эти методы являются инвазивными – т.е. для распознавания человека требуется его «сотрудничество» - например, положить палец в устройство по считыванию отпечатков. Поэтому, представляет интерес разработка алгоритма, способного распознавать людей лишь по их фотографии/видео.

Наиболее исследованными на данный момент являются алгоритмы распознавания по лицу [11]. Но, несмотря на весь прогресс в этой области, для качественного распознавания по лицу по-прежнему требуется большое разрешение изображения, а также контролируемое освещение. Вдобавок, человек зачастую может просто не посмотреть в камеру.

Из-за всех этих недостатков, в последнее время всё чаще используется дополнительная информация: об одежде, о цвете волос и пр. Эта информация не является инвариантной для человека – он может перекрасить волосы, переодеть одежду. Однако на протяжении небольшого промежутка времени (например, один день), все эти признаки часто остаются неизменными. В данной статье предлагается новый алгоритм распознавания человека по видео, снятому со стационарной камеры.

Рис 1. Пример роликов из созданной тестовой выборки.

Данный алгоритм, мотивированный статьей [5], использует случайный набор патчей, взятых из произвольных кадров видео, и классифицирует их с помощью случайного леса деревьев. Его основным достоинством является то, что он не опирается на маску переднего плана, полученную с помощью алгоритмов вычитания фона, а значит, допускает произвольный, сложный фон, с которым не справляются современные алгоритмы вычитания фона.

В виду отсутствия публично-доступных подходящих баз видеороликов, для тестирования предложенного алгоритма была создана собственная выборка видеороликов. Примеры кадров из этих видеороликов показаны на Рис. 1.

Оставшаяся часть статьи организована следующим образом: в секции 2 приведен обзор существующих методов, в секции 3 более подробно описана собранная база видеороликов, секция 4 описывает предложенный алгоритм, в секции 5 рассказывается о проведенных экспериментах, и, наконец, в секции 6 даётся заключение статьи.

2. СУЩЕСТВУЮЩИЕ ПОДХОДЫ

В основном распознавание людей, учитывающее одежду, используется в 2 задачах: аннотации изобрежений/видео и в системах видеонаблюдения.

В алгоритмах аннотации изображений/видео [1], [2] одной из основных проблем является сегментация человека. Современные алгоритмы сегментации не позволяют автоматически получить точную маску объекта на произвольном фоне. Поэтому стандартной схемой является следующая: вначале на изображении находится лицо человека, после чего в качестве области одежды берется прямоугольник под лицом человека. Координаты и размер прямоугольника задаются эвристически.

Понятно, что такой прямоугольник содержит лишь небольшую часть от всей области одежды. Более того, в некоторых случаях, например, если человек стоит нефронтально по отношению к камере, этот прямоугольник может частично захватить область фона.

В [3] используется более сложный алгоритм сегментации, однако ему тоже для первого приближения необходимо найти

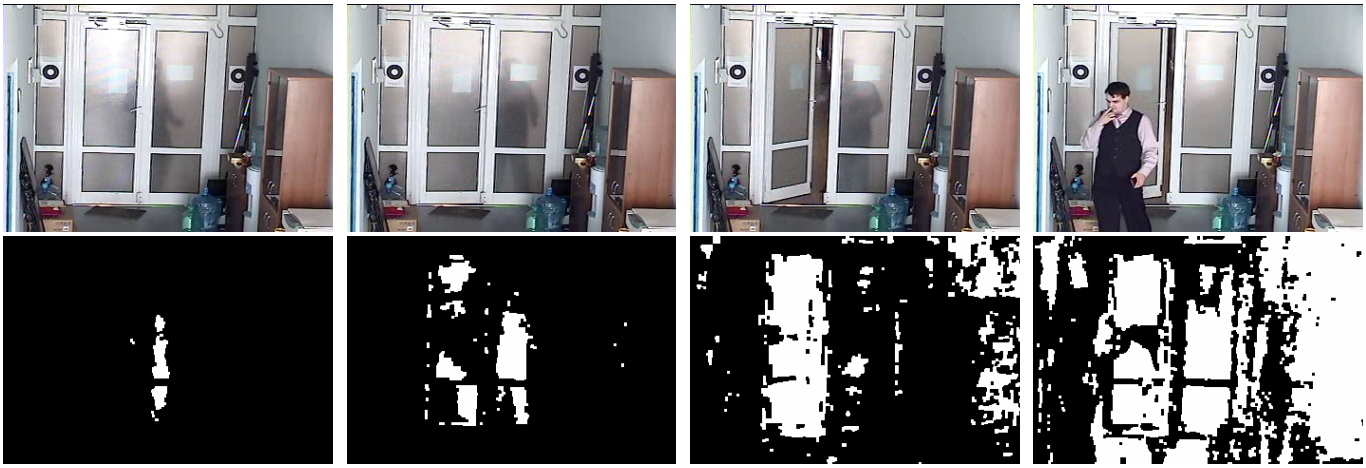


Рис 2. Демонстрация работы алгоритма вычитания фона.

лицо человека. С учетом того, что современные алгоритмы нахождения лица могут относительно надежно находить лишь фронтальные лица, то, в случаях, когда человек ни разу не посмотрит прямо в камеру (или его лицо будет чем-то загорожено), данный подход не сработает.

В [7] предложен метод, который на серии фотографий, используя информацию об одежде, цвете кожи и волос человека, позволяет обнаружить лица на тех фотографиях, на которых алгоритм нахождения лиц ничего не нашел. Однако, для получения цветовой модели одежды, кожи и волос, необходимо, чтобы лицо нашлось хотя бы на части из изображений.

В системах видеонаблюдения [6], [10] маска человека находится с помощью алгоритмов вычитания фона. И основной упор в этих статьях уже делается на том, что лучше всего использовать для описания одежды, и как ее лучше классифицировать.

Однако современные алгоритмы вычитания фона хорошо работают лишь в не очень сложных случаях. Например, чаще всего камера наблюдает за хорошо освещенным коридором. Но, как мы обнаружили на нашей выборке видеороликов, в более сложных случаях, эти алгоритмы могут давать неудовлетворительную маску. Более подробно об этом описывается в секции 3.

Предлагаемый в данной статье метод во многом основывается на статье [5]. Авторы [5] предложили метод для классификации изображений, который, в частности, тестировали на задачах классификации различных домов, разных типов сцен. В предложенном алгоритме, мы применяем похожую схему для классификации людей на видео. Главным отличием является то, что если в [5] классифицируемый объект занимал либо всё, либо большую часть изображения, то в нашем случае, человек занимает лишь маленькую часть от общего пространственно-временного объема видео.

3. ИСПОЛЬЗУЕМАЯ ВЫБОРКА

В рамках данной работы, нами была собрана собственная выборка видеороликов. Эти видеоролики записывались камерой, висящей под потолком, и направленной на входную дверь в лабораторию. Запись проводилась в течении 8 дней, всего было собрано 463 видеоролика. Разрешение видео – 352 на 240, средняя длительность ролика – около 10 секунд. Примеры кадров из собранных видеороликов показаны на Рис. 1. Всего на разных видеороликах присутствует 25 разных людей.

Каждый ролик был вручную аннотирован на предмет присутствующих на нем людей (их меток), а также флагом – входит данный человек в комнату или наоборот выходит из нее.

Во многих случаях было невозможно разбить общее видео на несколько роликов, в каждом из которых присутствовал бы лишь один человек – например, когда двое-трое человек одновременно выходят из комнаты. В данной работе такие ролики были удалены из рассмотрения. Стоит отметить, что существующие методы тоже пока не способны обрабатывать такие случаи (надежно сегментировать людей друг от друга).

С учетом выкидывания таких роликов, в каждый конкретный день людей, появляющихся на двух и более роликах в среднем было порядка 5-8 человек.

Т.к. изначально планировалось использовать стандартную схему по распознаванию людей, нами были протестированы несколько алгоритмов вычитания фона [8],[9] на собранных данных. Оказалось, что используемые видеоролики слишком сложны для испытываемых алгоритмов – входящие/выходящие люди очень нестабильно сегментировались от фона. Один из наиболее плохих примеров работы алгоритмов вычитания фона продемонстрирован на Рис. 2.

Основными причинами таких результатов являются:

- Открывающаяся дверь (и абсолютно произвольный фон за ней)
- Полупрозрачное стекло, за которым видны тени людей, стоящих за стенкой
- Нестабильное освещение.

При подгонке параметров алгоритмов на одних данных, алгоритм вычитания фона давал плохие результаты на других примерах. Общих параметров, подходящих для всех роликов найти не удалось.

В любом случае, даже если путем дополнительных усилий и удалось бы подобрать наилучшие параметры, добавить какие-нибудь эвристические правила (учитывающие

местонахождение двери на кадрах), полученный алгоритм работал бы лишь для данной сцены. Для запуска его на других сценах, пришлось бы заново искать параметры и эвристики. Для автоматического подбора параметров, для каждой новой сцены необходимо ручными (полуавтоматическими) методами отсегментировать большое количество роликов, что занимает существенное время. В нашем же случае, мы ограничиваемся лишь метками (идентификаторами) людей, присутствующих на видео.

Дополнительно, про собранную базу видеороликов стоит отметить, что в большинстве случаев люди меняли одежду каждый день. И даже на протяжении одного одня, один и тот же человек мог один раз появиться как в куртке, так и без нее.

4. ПРЕДЛОЖЕННЫЙ АЛГОРИТМ

4.1 Извлекаемые патчи

Вначале, из тренировочных видеороликов извлекается большое число (N_{Train}) квадратных патчей. Номер кадра, размер и положение каждого кадра выбирается случайным образом. После этого все извлеченные патчи масштабируются до фиксированного размера *rxr* и переводятся в цветовое пространство HSV.

Далее эти патчи вытягиваются в вектор длины $r * r * 3$ и каждому патчу присваивается метка – идентификатор человека с видеоролика, из которого взят данный патч.

Т.к. видеоролики получены с одной и той же стационарной камеры, то, в отличие от [5], становится возможным использовать в качестве признаков патча еще и его пространственное положение – координаты левого-верхнего угла (x, y) и ширину w.

Т.о. мы получаем тренировочную выборку длиной N_{Train} из векторов длиной $r * r * 3 + 3$.

4.2 Обучение и распознавание

Полученную тренировочную выборку можно подать на вход произвольному алгоритму машинному обучения.

С учетом размера входной выборки, а самое главное – с учетом длины вектора признаков каждого элемента выборки, алгоритм машинного обучения должен быть максимально быстрым.

В данном алгоритме используется случайный лес деревьев [4]. Для тренировки каждого отдельного дерева выбирается случайная подвыборка от общей тренировочной выборки. Используемые в узлах дерева функции – сравнение одной из координат входного вектора с порогом. При этом для скорости, выбор координаты и самого порога при обучении происходят абсолютно случайно. Каждое дерево строится до тех пор, пока оно не будет полностью правильно классифицировать свою тренировочную подвыборку. На этапе распознавания из тестового видеоролика также случайным образом выбирается N_{Test} квадратных патчей, после чего они также преобразуются в вектора $r * r * 3 + 3$.

Затем эти патчи подаются для классификации натренированному лесу деревьев. В итоге мы получаем $N_{Test} * K$ разных меток, где K - количество деревьев в лесе.

После этого, в качестве вероятности каждой метки L можно брать относительное число раз, когда патчи из тестового видео были классифицированы как L:

$$p(L) = \frac{\sum_{k=1}^{K} \sum_{x_i \in X} l(T_k(x_i) = L)}{N_{Test} * K},$$

где X - множество из N_{Test} векторов, $T_i(x)$ - результат классификации вектора x i-ым деревом.

Основная проблема заключается в том, что большинство патчей окажутся взятыми из фона, и не будут пересекаться с изображением человека. Поэтому для того, чтобы результат итогового распознавания оказался правильным, необходимо, чтобы извлекаемое число патчей было достаточно большим. В таком случае, патчи, соответствующие фону будут классифицироваться произвольно (и в среднем давать одинаковую вероятность всем меткам), а уже те патчи, которые пересекаются с изображением человека, будут давать бóльшую вероятность правильной метке.

5. ЭКСПЕРИМЕНТЫ

При проведении экспериментов, использовались следующие параметры:

- Число патчей в тренировочной выборке - $N_{Train} = 100000$
- Размер, к которому мастшабируются все патчи - $r = 16$
- Число патчей, извлекаемых из тестового видео - $N_{Test} = 3000$
- Число деревьев $K = 20$

Для тестирования использовалось 2 сценария:

- В первом сценарии используются лишь те ролики, на которых человек входит в комнату. В качестве тренировочных данных использовались первые 2 видеоролика каждого человека, остальные – попадали в тестовую выборку.
- Во втором сценарии уже использовались все видеоролики. Тренировочными были первые 3 ролика каждого человека, все остальные тестовые.

В обоих случаях, человек из тестовой выборки присутствовал в тренировочных данных.

Результаты работы алгоритма представлены на Рис. 3. Демонстрируемые результаты являются суммой результатов за все 8 дней наблюдения.

В качестве метрики качества используется метрика «N лучших» (TopN), которая для каждого конкретного n показывает, для какого процента всех роликов, правильная метка была среди первых n результатов.

Т.к. в один день в тренировочной выборке присутствовало максимум 14 человек, то значение метрики при $n \geq 14$ равно 100%.

Для ориентира на графиках приведен результат распознавания в случае, если бы классификация осуществлялась случайно (подбрасыванием монетки).

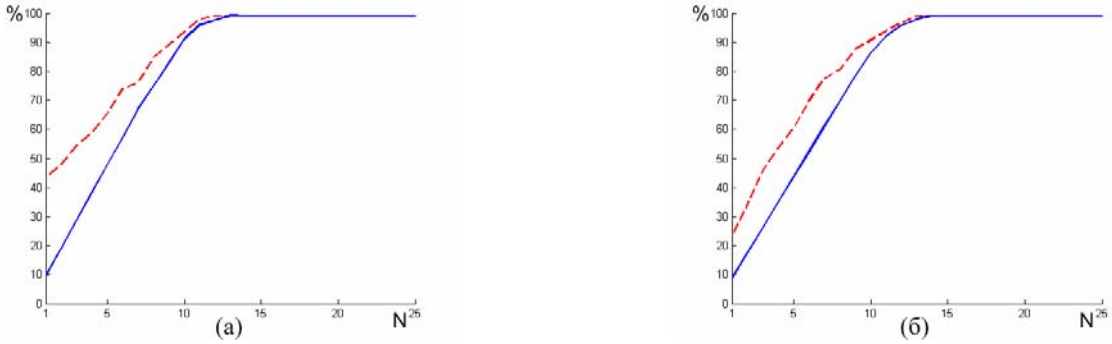


Рис 3. Результаты работы алгоритма. Красным (штриховая линия) отмечен результат работы предложенного алгоритма, синим (сплошная линия) – средний результат случайной классификации.

Из графиков видно, что полученные результаты еще далеки до того, чтобы их можно было надежно использовать в реальной системе. Например, в первом сценарии процент правильно распознанных роликов (т.е. TopN(1)) составляет лишь 45%.

Однако, также стоит учитывать и сложность входных данных. В частности, человек в один и тот же день может появляться на видео как в куртке, так и без нее. А значит, если, скажем, на первых 2 (тренировочных) роликах он будет в куртке, то распознать его на остальных роликах по используемым признакам будет практически невозможно.

Большую проблему составляет освещение. В тех случаях, когда к концу дня, когда уже темно, еще не включили свет в комнате, даже человек зачастую не сразу распознает людей на видео.

Во втором сценарии используется тот факт, что во многих случаях цвет и текстура одежды на спине очень похожа на одежду спереди. А значит, есть надежда, что алгоритм, натренированный на видео, на котором человек входит, сможет распознавать его же, но когда он выходит, и наоборот. Как видно, пока, к сожалению, при данном сценарии алгоритм сработал значительно хуже, чем в первом.

Тем не менее, полученные результаты позволяют надеяться на то, что при должной доработке данного алгоритма, он сможет достичь более хорошего процента правильного распознавания.

6. ЗАКЛЮЧЕНИЕ

В данной статье был предложен новый алгоритм распознавания человека по видео. Основным его преимуществом является то, что он не опирается на маску переднего плана, полученную с помощью алгоритмов вычитания фона, как это делают большинство существующих алгоритмов. Благодаря этому, для его тренировки достаточно предоставить лишь выборку видеороликов с меткой – идентификатором присутствующего на видео человека.

Данный алгоритм был протестирован на собственной выборке видеороликов. Несмотря на то, что полученные результаты еще далеки от идеальных, они позволяют говорить о потенциале данного подхода.

Работа выполнена при поддержке гранта РФФИ 09-01-92474-МНКС_а.

7. ЛИТЕРАТУРА

[1] Anguelov D., Lee K., Göktürk S.B., Sumengen B. *Contextual identity recognition in personal photo albums* Proc. of CVPR, pp. 1-7, 2007

[2] Everingham M. R., Sivic J., Zisserman A. *'hello! my name is... buffy' - automatic naming of characters in tv video* Proc. of BMVC, pp. 889–908, 2006

[3] Gallagher A., Chen T. *Clothing cosegmentation for recognizing people* Proc. of CVPR, No. 1, pp. 1-8, 2008

[4] Geurts P., Ernst D., Wehenkel L. *Extremely randomized trees* Machile Learning Journal, 63(1), 2006

[5] Maree R., Geurts P., Piater J., Wehenkel L., *Random Subwindows for Robust Image Classification* Proc. of CVPR, Vol. 1, pp. 34–40, 2005

[6] Nakajima C., Pontil M., Heisele B., Poggio T, *Full body person recognition system* Pattern Recognition, 2003

[7] Sivic J., Zitnick C. L., Szeliski R. *Finding People in Repeated Shots of the Same Scene* Proc. of BMVC, 2006

[8] Stauffer C., Grimson W.E.L., *Adaptive background mixture modelsfor real-time tracking* Proc. of CVPR, pp. 246-252, 1999

[9] Wren C., Azarbayejani A., Darrell T., Pentland A. *Pfinder:Real-time Tracking of the Human Body* IEEE Trans. on PAMI, Vol. 19, No. 7, pp. 780-785, 1997

[10] Yoon K., Harwood D., Davis L., *Appearance-Based Person Recognition Using Color/Path-Length Profile* Elsevier Real-Time Imaging, 2005

[11] Zhao W., Chellappa R., Phillips P. J., Rosenfeld A. *Face recognition: A literature survey*. ACM Computing Surveys, 2003

ABSTRACT

Video-based human recognition is a highly demanded task. Current face recognition algorithms still do not provide acceptable recognition rates in case of low-resolution video or in case of arbitrary lighting, hence algorithms, which take human clothes into account, are getting more and more attention.

In this paper, a new video-based human recognition algorithm is proposed, which is based on extraction of a large number of random patches. Unlike most modern algorithm, the proposed approach isn't based on a mask, obtained from background subtraction methods, and therefor it is capable of working on video with arbitrary complex background.

Mathematical Models of Fingerprint Image On the Basis of Lines Description

Vladimir Gudkov
Department of Applied Mathematics
Chelyabinsk State University, Chelyabinsk, Russia
diana@sonda.ru

Abstract

This paper presents a mathematical models of fingerprint based on the lines as topological vectors and ridge count. They are stored in the template with the list of minutiae. Templates are used to identify the fingerprint.

Keywords: *Fingerprint, minutiae, topology, ridge count, template, vector.*

1. INTRODUCTION

Fingerprint images (FI) identification is realized on the basis of the templates, containing the pattern features. Their basis is minutiae description in the form of beginnings and endings, junctions and bifurcation of lines [5-7]. They can be detected by gray image, though in the process of template creation they are guided by the lines skeleton [1, 5] (figure 1).

Figure 1. Skeleton and minutiae

Image mathematical model should depend on necessary and sufficient quantity of features [1]. Minutiae and ridge count between minutiae are reputed as informative in dactyloscopy [5]. However they do not exhaust all the great number of mathematical models, used in Biometrics for automatic proof of pattern individuality [3–6]. Each of such models is focused to increase the identification accuracy, but any better, free from defect model is not known [5]. For example, classical ridge count, which should be counted along the straight line due to criminalistics, does not stand up to criticism in the area of considerable curvature of pattern lines and in the area of loops, deltas and whorls [4, 7].

2. IMAGE TEMPLATE

The templates as a set of FI features are varied at different software producers. Some templates formats are limited in minutiae quantity [5]. Some features of templates are irrelevant but it is possible to indicate their common property: the templates have features, being some metrics for minutiae. These are ridge count between minutiae and topological vector for minutiae [5-7]. In paper an image template is synthesized in the form of

$$\Gamma : F_0^{(m)} \to \{L_m, L_l, L_r\}, \qquad (1)$$

where $F_0^{(m)} = \left[f_0^{(m)}(x,y) \right]$ – image skeleton (figure 1); L_m – minutiae list; L_l – topological vectors list for lines; L_r – ridge count vectors list for lines. Let's introduce some definitions.

Definition 1. The skeleton of the line is simple circuit $\langle u, v \rangle$ with nodes u and v in 8- adjacency, which is near geometrical center of the line, at that there are two adjacent nodes p_2 and p_3 for each node $p_1 \in \langle u, v \rangle$, at that the nodes p_2 and p_3 non-adjacent.

Definition 2. Ending is such node p_1 of the skeleton, that there is one adjacent node p_2 for the node p_1.

Definition 3. Bifurcation is such node p_1 of the skeleton, that there are three adjacent nodes p_2, p_3 and p_4 for the node p_1, at that any two nodes from the multitude $\{p_2, p_3, p_4\}$ are non-adjacent in pairs.

Elements of both topological vector and ridge count vector for line are determined with endings and bifurcations, which are read from the skeleton nodes as nodes of graph. These vectors are derivative from minutiae. However all these vectors characterize not the area of separately selected point, but common properties of point's multitude or line segment. In spite of the fact that the lists L_l and L_r characterize FI differently, they are alike in that they represent description for all pattern lines, but not for all the lines points, which can be much more in number. This interesting peculiarity allows synthesizing compact templates.

2.1. Minutiae list

Let M_i – is minutiae which is indexed to number i. The minutiae list L_m is in the following form

$$L_m = \left\{ M_i = \left\{ (x_i, y_i), \alpha_i, t_i, v_i, \theta_i, p_i, h_i \right\} \mid i \in 1..n_1 \right\}, \qquad (2)$$

where $|L_m| = n_1$ – cardinal number; (x_i, y_i), α_i, t_i, v_i, θ_i, p_i, h_i – coordinates, direction and type of minutiae, as well as value and direction of curvature, probability and density of lines about minutiae. Minutiae are detected in the informative area (on the figure 1 FI informative area is darkened, but the skeleton is black) [3, 4]. They are not detected on the edge of informative area.

Coordinates (x_i, y_i) of minutiae M_i are determined with coordinates of skeleton node [1]. Direction α_i like the angle is determined with circuit of skeleton nodes for ending and tree circuits for bifurcation [2, 5]. Type $t_i \in \{0,1\}$ is determined with skeleton nodes valence like the nodes of graph [2], where 0 – bifurcation and 1 – ending. Coordinates (x_i, y_i), direction α_i and type t_i are the basic parameters M_i [5-7].

Value v_i and direction θ_i of the curvature are determined by lines direction difference in the neighborhood ε of minutiae M_i [3, 4]. Probability p_i is calculated as the ratio of the average value of image quality rating in the neighborhood ε to the best quality rating in the FI informative area [3]. Lines density h_i is calculated as the average quantity of lines, located into the neighborhood ε on the straight line traced transversely to lines [3-5]. A value of neighborhood ε is assigned.

2.2. Topological vectors list

Topological vectors list for lines L_l is found on the basis of minutia list L_m, skeleton as the matrix $F_0^{(m)}$ and other auxiliary matrixes, elements of which reflect FI local features. These matrixes are formed in the pyramid, which present data informational layers, distributed among the hierarchy [1].

Topological vectors list for lines is synthesized on the basis of all the nodes of skeleton, excluding minutiae nodes, and written in the form of

$$L_l = \left\{ V_i = \left\{ (e_j, n_j, l_j) \right\} \middle| \, i \in 1..n_2, j \in 1..m_t \right\}, \quad (3)$$

where V_i – topological vector for skeleton nodes cluster; $|L_l| = n_2$ – cardinal number and $n_2 > n_1$; i – index like the number of topological vector; j – number of link in topological vector; e_j – event, and l_j – length of link, formed with minutiae with number n_j; m_t – quantity of links taking into account central line in the form of

$$m_t = 4m + 2 . \quad (4)$$

Let's dwell on the procedure of list synthesis.

In the FI informative area the lines are marked out and an image formalized as skeleton is formed. Two types of minutiae are detected on skeleton: endings and bifurcations (figure 1). Minutiae directions (angle) point to the area of lines number increase [5]. It is parallel to the tangent to papillary line in the small neighborhood of minutiae M_i. Every minutia is numbered and described with coordinates, direction, type, value and direction of curvature, probability and density (2).

Further from each minutia we draw projections to the right and to the left transversely to the direction vector of the minutiae onto adjacent lines and fix the projections. On the figure 2 the projections are shown with dotted lines, and two corresponding nodes of skeleton on the lines 1 and 2 are painted over.

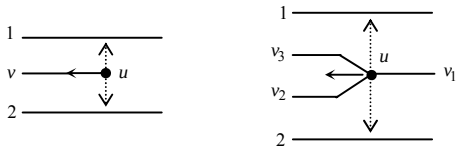


Figure 2. Projections for ending and bifurcation

Let's choose the skeleton node p_i (not the minutiae) and pass across its coordinates (x_i, y_i) the section to the right and to the left at a distance of m lines transversely to the tangent to lines being crossed and enumerate on gyrate the dissected lines (hereinafter 'links'), which turn clockwise. The section passes and traces the lines curvature [3]. The section depth m is varied from one up to eight lines to the right and the same to the left. One line in the section forms two links. The quantity of links in topological vector is calculated according to (4).

Topological vector is determined by the section. To do this we'll follow the move of every link by turns on each link, not leaving it and beginning from the section till meeting another minutiae, located on the link, or a minutiae projection, located on an adjacent line to the right or to the left of the link. At that on the links detected the following possible events, shown on the figure 3 and represented in a binary code:

1101 – an ending projection is detected on the link, which locates to the right of the link in the course of tracing the link, the direction vector of the ending is directed contrariwise to the link course;

1001 – an ending projection is detected on the link, which locates to the right of the link in the course of tracing the link, the direction vector of the ending is directed along the link course;

1110 – an ending projection is detected on the link, which locates to the left of the link in the course of tracing the link, the direction vector of the ending is directed contrariwise to the link course;

1010 - an ending projection is detected on the link, which locates to the left of the link in the course of tracing the link, the direction vector of the ending is directed along the link course;

0101 - a bifurcation projection is detected on the link, which locates to the right of the link in the course of tracing the link, the direction vector of the bifurcation is directed contrariwise to the link course;

0001 - a bifurcation projection is detected on the link, which locates to the right of the link in the course of tracing the link, the direction vector of the bifurcation is directed along the link course;

0110 - a bifurcation projection is detected on the link, which locates to the left of the link in the course of tracing the link, the direction vector of the bifurcation is directed contrariwise to the link course;

0010 - a bifurcation projection is detected on the link, which locates to the left of the link in the course of tracing the link, the direction vector of the bifurcation is directed along the link course;

0011 – a bifurcation is detected on the link, the direction vector of the bifurcation is directed along the link course;

0111 – a bifurcation is detected on the link, formed with the line, the tangent to which forms a minimum angle at the bifurcation direction vector rotating on the link counterclockwise;

1011 – a bifurcation is detected on the link, formed with the line, the tangent to which forms a minimum angle at the bifurcation direction vector rotating on the link clockwise;
1111 – an ending is detected on the link, the direction vector of the ending is directed contrariwise to the link course;
1100 – a minutia or a minutia projection is not detected and the link on the line is close.
0000 – a minutia or a minutia projection is not detected and the link terminates at a papillary pattern or non-informative area edge;

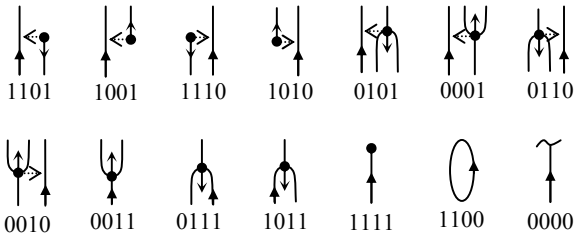


Figure 3. Events

Minutiae' number initiated the event is associated with the event as the number (14 in sum) detected on the link. The event is tied to the link number. For 0000 and 1100 events the numbers of minutiae are absent. Enumerated set of links with formed events and minutiae' numbers is the ***basic topological vector*** (economical). The event and minutiae' number form ordered pair (e_j, n_j). The event is amplified with link length from the section up to the position, in which the event is detected. ***Enlarged topological vector*** is formed in such way. The event, minutiae' number and length of the link form an ordered triplet (e_j, n_j, l_j). For 0000 and 1100 events the links lengths can describe the informative areas without minutiae! The lengths of links, broken on FI edge, are stable in the meaning that they are not shortened in case of fool rolling of the finger [3].
Bit location in the event determines minutiae type, its direction regarding the link course, its location regarding the link etc. Events allow on-the-fly compare the ***basic topological vectors*** and speed up the identification procedure [3].
Topological vectors are built for every node of the skeleton p_i (excepting minutiae). The section divides the lines into the links, numbered on gyrate, turning clockwise. On the figure 4 in the section for the node A of skeleton line, ending in 19, the links are enumerated as 0–17. Topological vector of the node A is shown in the table 1. On the figure 5 in the section for the node B of the skeleton line, which is locked in bifurcation 19, the links are enumerated as 0–17. Topological vector of the node B is shown in the table 2. The sections are shown with dotted line, and the figures represent usual mutation [3] of ending 19 into bifurcation 19 (because of the dirty skin). Per se the nodes A and B of the skeleton are the same.
The start of links numbering in the section for the nodes A and B (link № 0) is insignificant, as since in case of FI turn over the mirror of links numbers in the section is formed, which is easy recognized and taken into account at FI identification. By analogy with the game «Puzzle» assembling is realized by the way of joining of corresponding connectors. At the section depth $m = 4$ 18 links $m_t = 18$ for the line are formed according to (4).
The quantity of topological vectors is enumerable. At the foot of the figure 4 with two-forked dotted arrow is shown the zone, located between minutiae 19 and 25, within the bounds of which for the point A at its displacement on the skeleton the same ***basic topological vector*** is synthesized. The similar zone between minutiae 19 and 25 for the point B is shown at the foot of the figure 5. Topological vectors with equal ***basic topological vectors*** are integrated into one [3]. At that their quantity is reduced by dozens of times from the value $n_2 < 1000$ according to (3) to the value $n_1 < 100$ according to (2). Vector V_i automatically characterizes the line segment, but not the minutiae. The image deformation, especially linear, practically doesn't have any effect upon the content of the ***basic topological vector***. Therefore the vector is named as topological [3, 6, 7].
Here on the list formation (3) is finished.

Table 1. Topological vector for A

Number	Event	Index	Length
0	1110	22	l_0
1	1111	19	l_1
2	1110	19	l_2
3	1111	22	l_3
4	0001	21	l_4
5	1101	19	l_5
6	1010	24	l_6
7	0010	25	l_7
8	0011	21	l_8
9	1111	23	l_9
10	1010	26	l_{10}
11	0011	25	l_{11}
12	0010	21	l_{12}
13	1010	20	l_{13}
14	1111	27	l_{14}
15	0001	25	l_{15}
16	0000	–	–
17	1001	20	l_{17}

Table 2. Topological vector for B

Number	Event	Index	Length
0	1110	22	l_0
1	1011	19	l_1
2	0111	19	l_2
3	1111	22	l_3
4	0001	21	l_4
5	0101	19	l_5
6	0110	19	l_6
7	0010	25	l_7
8	0011	21	l_8
9	1111	23	l_9
10	1010	26	l_{10}
11	0011	25	l_{11}
12	0010	21	l_{12}
13	1010	20	l_{13}
14	1111	27	l_{14}
15	0001	25	l_{15}
16	0000	–	–
17	1001	20	l_{17}

Comparative analysis of [3] and [6] show, that [3] has a series of advantages. Firstly, the section is built along the curved line, which traces curvature direction of the crossed lines. Secondly, at the events calculation the projection of minutiae is used, that result in prevention of the information loss. Thirdly, the links enumerating is turning along the gyrate without links omission.

Fourthly, at integration it is possible to choose topological vector with maximum value of minimal length of link (3). This raises stability and comprehension of mathematical model.

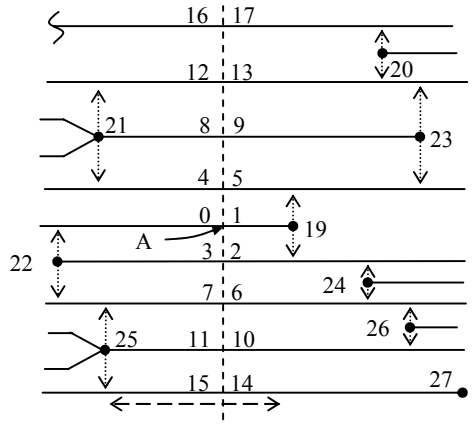


Figure 4. Section for line with ending

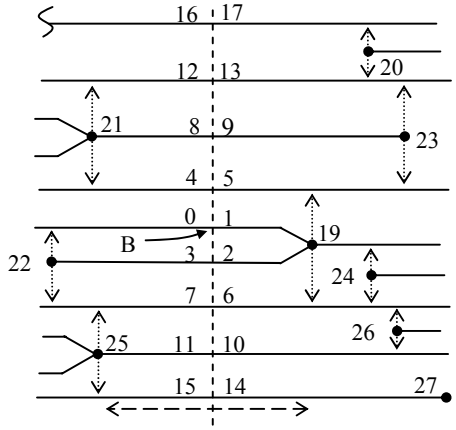


Figure 5. Section for line with bifurcation

2.3. Ridge count vectors list

In dactyloscopy the ridge count is the base for evidence of fingerprint identity in a Court. It should be calculated as the quantity of lines, placed on the straight line between two minutiae. In electron systems for one minutia M_i, as a rule, some similar values are determined.

According to [5] from the list L_m regular minutia M_i is chosen and taken as a center of the scanning ray rotation, the initial location of which coincides with the vector direction $\theta_i \in M_i$ according to (2). The image is scanned, rotating the scanning ray. At the meeting of a ray with $M_k \in L_m$ the ordered pair (r_j, n_j) is formed, where $n_j = k$ – number of minutiae M_k and $k \neq i$; r_j – ridge count between M_i and M_k; j – number of link as the quantity of minutiae met by a ray. The number of such pairs is not more than $n_1 - 1$.

These actions repeat $\forall M_i \in L_m, i \in 1..n_1$. Geometrical characteristics connected with ordered pair and served as derivative from the minutiae and the scanning angle parameters, are redundant for the model and calculated at the time of identification. In the templates the links quantity for M_i, as a rule, is limited, and the links are clustered as quadrants or octants, oriented according to $\theta_i \in M_i$ of the co-ordinates.

There are three main disadvantages of ridge count, mentioned in [4-6]. Firstly, a value r_j is unstable as since in case of mutation of the ending into bifurcation or vice versa the ridge count value is changed. Secondly, in the area of loops, deltas, whirls and considerable curvature of lines, the ridge count is doubtful because of the straight line measuring mechanism. Thirdly, clustering of ridge count values in quadrants or octants results in appearance of boundary effect, which realized in the fact that, minutiae can pass through the limits of plane partition into regions. This increases the identification mistakes a lot.

To solve mentioned defects, it is suggested the list of ridge count vectors for lines, which is synthesize on the basis of all the nodes of skeleton $F_0^{(m)}$, excluding minutiae nodes, of some auxiliary matrixes of the pyramid [1] and minutiae list L_m in the form of

$$L_r = \left\{ R_i = \left\{ (r_j, n_j) \right\} \middle| \, i \in 1..n_3, j \in 1..n_4 \right\}, \qquad (5)$$

where R_i – is a vector of ridge count for the nodes group of the skeleton as ordered by index j set of ordered pairs (r_j, n_j); $|L_r| = n_3$ – cardinal number and $n_3 > n_1$; i – index as a number of vector; j – link number in a vector; n_4 – links quantity in a vector and $n_4 < n_1$; r_j – ridge count value, and n_j – minutiae number as in (2) on a link j.

Let's dwell on the sequence of operations being carried out.

In the informative area of FI we choose lines and form skeleton, using which we detect two types of minutiae: endings and bifurcation [5]. Minutiae direction (angle) point to the area of lines number increase (figure 2). It is parallel to the tangent of papillary line in small area around the minutiae M_i. Each minutia is numbered and described with coordinates, direction and type, as well as curvature value and direction, probability and density of the lines in the neighborhood ε of the minutia (2).

Let's choose the skeleton node p_g (but not the minutiae) and take it for the rotation center of the scanning ray, initial direction of which coincides with the skeleton line direction. The image is scanned, rotating the scanning ray. At the meeting of a ray with $M_k \in L_m$ the ordered pair (r_j, n_j) is formed, where $n_j = k$ – number of minutiae M_k; r_j – ridge count between p_g and M_k; j – number of link as the quantity of minutiae met by a ray. As a result of one rotation of a ray ***ridge count vector*** $R_g = \left\{ (r_j, n_j) \middle| \, j \in 1..n_1 \right\}$ is formed as enumerated by index j set of ordered pairs (r_j, n_j). These operations are represented for all skeleton nodes (excepting minutiae). The links are closed as a ring and the enumeration can be renew with changing of the initial direction of a scanning ray, for example $R_g^s = \left\{ (r_l, n_l) \middle| \, l \in 1..n_1 \right\}$, where the number of link is $l = (j + s) \bmod (n_1 + 1)$. In that case for the node p_g on the basis of vector R_g it is possible taking into account links closing as a ring to synthesize $\left| R_g^s \middle| \, s \in 1..n_1 \right| = n_1$ ***equivalent vectors of ridge count***.

The final set of ridge count vectors is formed for the skeleton. ***Ridge count vectors***, for which it is possible to synthesize the similar ***equivalent vectors of ridge count***, integrate into one [4]. Join means a choice, for example at random one of integrated vectors and placing it into the list of ridge count vectors $L_r = \{ R_i \mid i \in 1..n_3 \}$ according to (5). The quantity of vectors is reduced by one-two order. A vector R_i for the image is unique and automatically characterizes the line segment.
Here on the list building (5) is finished.
There are following advantages. Firstly, ridge count on the one end is closed to the node of skeleton, which is not prone to minutiae mutations, that increases the stability of the ridge count value r_j . Secondly, the lines in the area of loops, deltas, whirls and essential curvature are automatically divided with vectors R_i for short intervals that allow detailing differently the FI conception and reducing mistakes of identification. The quantity of links in the vector can be limited, for example, on the ray length criterion.

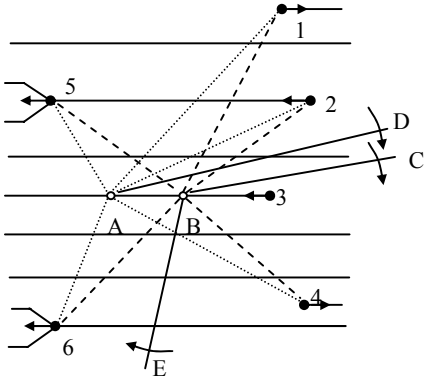


Figure 6. Ridge count of a line

On the figure 6 the node B is chosen as a center of scanning ray BC rotation. At the scanning ray rotating clockwise round the node B with indicated on the figure 6 initial location, the scanning ray will meet with minutiae 3,4,6,5,1,2 in turn. If ridge count determined with quantity of crossed lines along the ways, shown with long-dotted line, than for the sequence of met minutiae ridge count values 0,2,2,1,3,1 are generated. This represents with ordered set of pairs (0,3),(2,4),(2,6),(1,5),(3,1),(1,2), which forms ridge count vector for the node B. At another initial location of the scanning ray, for example ray BE, another set of pairs (2,6),(1,5),(3,1),(1,2),(0,3),(2,4) is determined, which in case of closing into the ring identified with source set of pairs (0,3),(2,4),(2,6),(1,5),(3,1),(1,2). Actually, for the set of ordered pairs of the node B there are 6 equivalent vectors of ridge count.
If the node A is chosen with the help of scanning ray AD, which is rotating clockwise, than the ordered set of pairs (0,3),(2,4),(2,6),(1,5),(3,1),(1,2) is formed. It coincides with one of equivalent vectors of ridge count for the node B (independently of the initial direction of the ray AD). Vectors of ridge count for the node A and nodes B are integrated into one.

3. CONCLUSION

In this paper we suggested mathematical model of FI on the basis of topological vectors for the lines (3), which is stored in the template (1). Topological vectors form linked graph with a high level of redundancy. This allows connecting sub graphs [3] of fragmentary latents of fingerprints. List L_l according to (3) can be represented in economic format (without links lengths). Minutiae mutation don't change the links enumerating and minutiae enumeration queue (tables 1, 2), that increases stability of mathematical model.
Topological vectors stability is additionally increased by integrating ***basic topological vectors*** into one corresponding ***enlarged topological vector***, where the minimum length of link is maximal [3].
We suggested mathematical model of FI on the basis of ***ridge count vectors*** for lines (5), which demonstrate raised stability at calculating of ridge count values. Additionally the lines in the area of loops, deltas, whirls and essential curvature are automatically divided with ridge count vectors in more detail independently from minutiae location that increases stability of mathematical model as well.
An image template according to (1) as a set $\{L_m, L_l, L_r\}$ consists of lists L_l according to (3) and L_r according to (5) which are mutually complementary, for them minutiae list is the determining one according to (2). The lists L_l and L_r are essentially different and don't replace each other, but one of these lists can be excluded from this template. Moreover topological vectors can be stored in enlarged or economic variant (without links lengths). This allows optimizing required memory capacity for the template storage.

4. REFERENCES

1. Gonzales, R. Digital processing of the images / R. Gonzales, R. Woods; translation from English; the editor P. Chochia – M.: Techno sphere, 2006. – 1072 p.
2. Novikov, F. Discrete Mathematics for programmers: manual / F. Novikov. – St. Petersburg: Piter, 2001. – 304 p.
3. Patent 2321057 Russian Federation, Int. Cl. G 06 K 9/52, A 61 B 5/117. The Method of Papillary Pattern Print Coding / V. Gudkov – № 2006142831/09; Field: Dec. 04, 2006; Date of patent: Mar. 27, 2008; Bull. № 9. – 13 p.
4. Favourable decision on application Russian federation, Int. Cl. G 06 K 9/00. The Method of Papillary Pattern Print Coding / V. Gudkov, A. Bokov, A. Mosunov. – № 2007118575/09; Field: May. 18, 2007; Date of patent: Nov. 27. 2008; Bull. № 33. – 13 p.
5. Maltoni, D. Handbook of fingerprint recognition / D. Maltoni, D. Maio, A.K. Jain. – New York: Springer-Verlag, 2003. – 348 p.
6. Pat. 5631971 USA, Int. Cl. G 06 K 9/00. Vector based topological fingerprint matching / M.K. Sparrow (Winchester). – Field: Jul. 15, 1994; Date of patent: May. 20, 1997; U.S.Cl. 382/125. – 17 p.
7. Sparrow, M.K. A topological approach to the matching of single fingerprints: development of algorithms for use on latent finger marks / M.K. Sparrow, P.J. Sparrow // US dep. comer. nat. bur. stand. spec. pub. – 1985. – № 500–126. – 61 p.

About the author

Vladimir Gudkov is a doctor at Chelyabinsk State University, Department of Applied Mathematics. His contact e-mail is diana@sonda.ru.

Алгоритм идентификации человека по ключевым точкам радужной оболочки глаза

Е.А. Павельева, А.С. Крылов

Лаборатория математических методов обработки изображений,

Факультет Вычислительной Математики и Кибернетики МГУ имени М.В. Ломоносова, Москва, Россия

E-mail: paveljeva@yandex.ru, kryl@cs.msu.ru

Аннотация

В работе предложен алгоритм распознавания человека по радужной оболочке глаза на основе преобразования Эрмита, использующего локальные характеристики радужной оболочки. При построении кода радужной оболочки используются только наиболее информативные функции преобразования Эрмита. На основе метода преобразования Эрмита разработан также метод, использующий только ключевые точки радужной оболочки. Этот метод позволяет значительно уменьшать количество хранимых ключевых точек радужной оболочки без сильных потерь в качестве распознавания.

Ключевые слова: распознавание по радужной оболочке глаза, преобразование Эрмита, ключевые точки, биометрия.

1. ВВЕДЕНИЕ

Преобразование Эрмита [1] является известным методом для решения биометрических задач [2,3,4]. В преобразовании Эрмита вычисляются свертки функции интенсивности изображения с функциями преобразования Эрмита (производными функции Гаусса) в каждой точке изображения, т.е. анализируются локальные свойства изображений. Также широко известным методом в обработке изображений является метод моментов Гаусса-Эрмита [2], эквивалентный преобразованию Эрмита [4] с точностью до знаков сверток для нечетных функций преобразования Эрмита. В методе моментов Гаусса-Эрмита вместо сверток ищутся корреляции функции с производными функции Гаусса. Обычно при использовании этих методов в задачах идентификации рассматриваются лишь знаки сверток в каждой точке, а затем производится сравнение бинарных матриц, составленных из знаков сверток, с бинарными матрицами из базы данных.

В данной работе предложен метод нахождения наиболее информативных точек текстуры радужной оболочки на основе преобразования Эрмита. Для этого анализируется сумма модулей сверток функции интенсивности изображения с функциями преобразования Эрмита с наиболее информативными номерами [5] во всех точках области параметризации радужной оболочки. Точки с максимальным значением суммы соответствуют характерным точкам текстуры изображения радужной оболочки. При этом обнаружено, что для предложенного метода уменьшение количества хранимых ключевых точек радужной оболочки не приводит к сильным потерям в качестве распознавания.

В мультибиометрических системах распознавания человека [6,7], размер кода каждой биометрической составляющей и скорость распознавания важны больше, чем 100% точность распознавания по одному признаку. Поэтому предложенный метод ключевых точек для идентификации по радужной оболочке может применяться для распознавания человека одновременно с другими биометрическими показателями, такими как голос, изображения лиц и отпечатки ладоней, пальцев.

2. ПРЕОБРАЗОВАНИЕ ЭРМИТА

2.1 Функции Эрмита

Функции Эрмита определяются следующим образом:

$$\psi_n(x) = \frac{(-1)^n \cdot e^{-x^2/2}}{\sqrt{2^n \cdot n! \cdot \sqrt{\pi}}} \cdot H_n(x),$$

где $H_n(x)$ - полиномы Эрмита:

$$H_0(x) = 1,$$
$$H_1(x) = 2 \cdot x,$$
$$H_n(x) = 2 \cdot x \cdot H_{n-1}(x) - 2 \cdot (n-1) \cdot H_{n-2}(x).$$

Функции Эрмита являются собственными функциями преобразования Фурье, образуют полную ортонормированную систему функций в пространстве $L_2(-\infty, \infty)$ и являются локализованными с вычислительной точки зрения в координатном и частотном пространствах.

2.2 Функции преобразования Эрмита

Функции преобразования Эрмита (Рис. 1) определяются через функции Эрмита:

$$\varphi_n(x) = \psi_0(x) \cdot \psi_n(x) = \frac{(-1)^n \cdot e^{-x^2}}{\sqrt{2^n \cdot n! \cdot \pi}} \cdot H_n(x).$$

Они также являются локализованными с вычислительной точки зрения. Так как ψ_n - ортонормированная система функций, то $\int_{-\infty}^{\infty} \varphi_n(x)\,dx = \int_{-\infty}^{\infty} \psi_0(x) \cdot \psi_n(x)\,dx = 0 \ \forall n > 0$, т.е. функции $\varphi_n(x)$ имеют среднее нулевое значение для номеров $n > 0$. Это свойство является важным в методах, использующих знаки сверток с такими функциями.

Двумерные функции преобразования Эрмита являются произведением одномерных функций:

$$\varphi_{n,m}(x, y) = \varphi_n(x) \cdot \varphi_m(y).$$

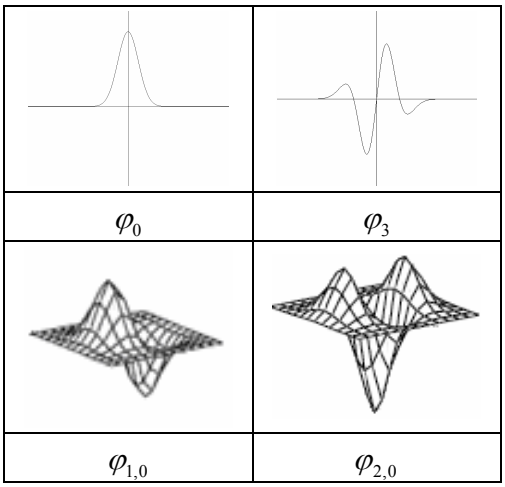


Рис.1: Примеры функций преобразования Эрмита.

2.3 Преобразование Эрмита

Преобразование Эрмита для изображения определяется в каждой точке (x_0, y_0) значениями сверток функции интенсивности изображения $I(x, y)$ с функциями преобразования Эрмита $\varphi_{m,n}(x, y)$ для выбранного конечного набора индексов *(m,n)* [4]:

$$M_{m,n}(x_0, y_0) = \big(I(x, y) * \varphi_{m.n}(x, y)\big)(x_0, y_0) =$$
$$= \iint_G I(x, y) \cdot \varphi_{m,n}(x_0 - x, y_0 - y)dx\,dy,$$

где G – область сосредоточения функции $\varphi_{m,n}$

3. МЕТОД ПАРАМЕТРИЗАЦИИ РАДУЖНОЙ ОБОЛОЧКИ НА ОСНОВЕ АНАЛИЗА ЗНАКОВ ПРЕОБРАЗОВАНИЯ ЭРМИТА

Алгоритм выделения радужной оболочки описан в [8,5] и основывается на поиске максимума скачка средней интенсивности на изображении вдоль круговых контуров. В данной работе произведено уточнение областей интегрирования при поиске внешней границы радужной оболочки, свободных от попадания в них век и ресниц. В каждой из четырех прямоугольных областей, выделенных на Рис. 2, подсчитывается среднее значение интенсивности I_i. Длина дуги окружности, по которой происходит интегрирование в *i*-й области, пропорциональна соответствующему углу

$$\alpha_i = \frac{90^0 \cdot I_i}{\sum_{j=1}^{4} I_j},\ i \in \{1,2,3,4\}.$$

Таким образом, чем больше ресниц попадает в в *i*-ю область, тем меньше дуга окружности интегрирования в этой области.

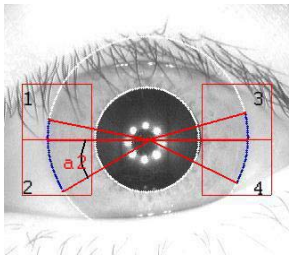


Рис.2: Локализация радужной оболочки.

После выделения радужной оболочки глаза (локализации), радужная оболочка переводится в прямоугольное нормализованное изображение. Для дальнейшей параметризации нами используется только область, включающая правую верхнюю четверть нормализованного изображения, на которую, как правило, не попадают ресницы и века [5] (Рис.3).

Рис.3: Нормализация радужной оболочки.

В предлагаемом методе, называемом в дальнейшем методом знаков, проводится параметризация радужной оболочки на основе анализа знаков сверток преобразования Эрмита в каждой точке (x_0, y_0) области параметризации для выбранного набора индексов *(m,n)*:

$$L_{m,n}(x_0, y_0) = \operatorname{sgn} M_{m,n}(x_0, y_0).$$

В качестве метрики сравнения по данным значениям $L_{m,n}$ берется расстояние Хэмминга между соответствующими прореженными [5] матрицами радужных оболочек $HD(L)$. В работе [5] проведен анализ зависимости расстояний между матрицами изображений радужных оболочек от номеров функций преобразования Эрмита. Показано, что наилучший результат достигается при сравнении и сопоставлении значений

$$HD(L_{1,0}) + HD(L_{2,0}) \text{ и } HD(L_{1,0}) + HD(L_{2,1}).$$

Если минимальные значения для каждого из этих двух выражений достигаются для изображений одного человека из базы данных, то распознавание считается успешным. В противном случае, данная радужная оболочка дополнительно исследуется на наличие попадания нижнего века в область параметризации. В случае детектирования века, данное изображение считается не подлежащим идентификации. В остальных случаях распознавание считается неуспешным.

3.1 Определение наличия века в области параметризации

Для определения наличия века в области параметризации рассматривается вертикальная производная яркости изображения $I'(x, y) = \dfrac{\partial I(x, y)}{\partial y}$ и ищется

$$\max_{y}\sum_{x}\left|I'(x,y)\right|$$

в области $\left[x_p-\frac{r}{2},\, x_p+\frac{r}{2}\right]\times\left[y_p+r,\, y_p+\frac{(R+r)}{2}\right]$,

выделенной на Рис.4. Здесь (x_p, y_p)-центр зрачка, r и R-радиусы границ радужной оболочки. Если это значение больше порогового, то считается, что нижнее веко попало в область параметризации.

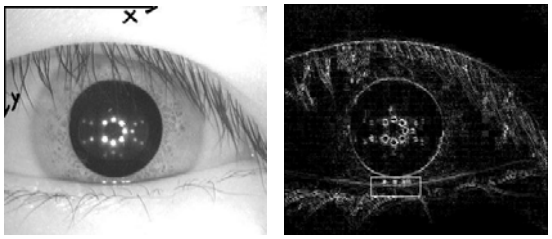


Рис. 4: слева – исходное изображение $I(x,y)$, справа – изображение $I'(x,y)$.

3.2 Результаты работы метода параметризации на основе анализа знаков преобразования Эрмита

В Таблице 1 приведены результаты работы метода на базе данных CASIA-IrisV3 [9] – значения FAR и FRR (False Acceptance/Rejection Rate определяются количеством неверных допусков/отказов). Проведено сравнение предложенных методов с результатами описанных в литературе существующих методов идентификации по радужной оболочке глаза.

МЕТОД	FAR (%)	FRR (%)	БАЗА ДАННЫХ
Метод знаков	0	0.82	CASIA-IrisV3
Метод знаков с учетом определения наличия века	0	0.33	CASIA-IrisV3
Tan [2]	0.001	1.13	CASIA V1.0
Tan [10]	0.001	0.4	CASIA V1.0
Romero-Ramirez [4]	0	9.71	CASIA V1.0
Daugman [11]	0	0.008	NIST (ICE-1)

Таблица 1: Сравнение методов идентификации.

4. МЕТОД ПАРАМЕТРИЗАЦИИ РАДУЖНОЙ ОБОЛОЧКИ НА ОСНОВЕ ВЫДЕЛЕНИЯ КЛЮЧЕВЫХ ТОЧЕК

В работе [5] показано, что наиболее информативными номерами двумерных функций преобразования Эрмита $\varphi_{m,n}(x,y)$ являются номера (1,0), (1,1), (2,0). Эти номера взяты в данной работе для выделения ключевых точек радужной оболочки.

Рассмотрим в каждой точке области параметризации величину $F=\left|M_{1,0}\right|+\left|M_{2,0}\right|+\left|M_{1,1}\right|$ и отсортируем все точки по убыванию величины F. В качестве кода радужной оболочки (ключевых точек) рассматривается матрица 500 точек с максимальными значениями F (пример кода радужной оболочки приведен на Рис. 5). На Рис. 5 пунктиром обозначена граница возможных значений точек кода (граница отстоит от краев области параметризации на полуширину области сосредоточения функции φ_2 - функции преобразования Эрмита с максимальным используемым номером).

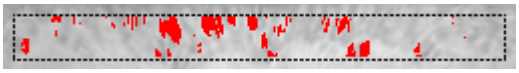


Рис. 5: Область параметризации радужной оболочки с кодом радужной оболочки.

Метод ключевых точек сильно неустойчив к наличию века и бликов в области параметризации, т.к. области века и бликов стягивают к себе все ключевые точки (Рис. 6). Поэтому этот метод следует применять только для изображений без наложения века на область параметризации.

Рис. 6: Пример работы алгоритма выделения ключевых точек на изображении с наложением века на область параметризации.

4.1 Сравнение изображений радужных оболочек

В качестве метрики сравнения кодов радужных оболочек используется расстояние Хэмминга межу матрицами ключевых точек. Чтобы алгоритм был устойчив к поворотам глаза (поворот глаза соответствует циклическому сдвигу всего нормализованного изображения), сравниваются матрицы точек уменьшенного размера. Границы такой урезанной матрицы выделены на Рис. 7 и сдвигаются у исследуемого изображения в обе стороны до границ возможных значений точек кода. В данной работе учитываются углы поворота от -10º до 10º.

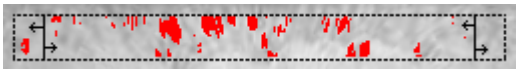


Рис.7: Области сравнения матриц кодов радужных оболочек.

4.2 Результаты алгоритма ключевых точек

Алгоритм параметризации радужной оболочки на основе выделения ключевых точек протестирован на базе данных CASIA-IrisV3, содержащей 2655 изображений глаз. Результаты работы алгоритма приведены в Таблице 2.

Количество неверных ближайших изображений (из 2655)	Неверные из-за наложения века и бликов	Остальные случаи (ошибки FAR+FRR метода)
67 = 54+13	54	13 (0.49%)

Таблица 2: Результаты работы алгоритма ключевых точек.

Если применить метод ключевых точек к тем 0.33% изображений базы данных CASIA-IrisV3 из Таблицы 1, обладатели радужных оболочек которых не были найдены методом знаков с учетом определения наличия века, то он даёт положительный результат на всех этих изображениях. Таким образом, комбинированный метод, включающий последовательно метод анализа знаков преобразования Эрмита и метод ключевых точек, позволяет получить очень надежный алгоритм идентификации человека по радужной оболочке глаза.

Было также обнаружено, что предложенный метод ключевых точек позволяет значительно уменьшать количество хранимых ключевых точек радужной оболочки без сильной потери качества распознавания. В эксперименте рассмотрен выбор не 500 ключевых точек, а 100. При этом дополнительно было поставлено условие, что точки не должны соприкасаться друг с другом (расстояние между точками $\geq$ 2 пикселей). Видно, что 100 точек, взятых таким образом (Рис.8), качественно определяют те же области, что и 500 (Рис.5).

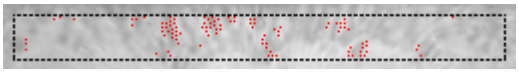


Рис.8: Код радужной оболочки в случае 100 ключевых точек.

Предложенная модификация метода включает также следующее дополнительное уменьшение размера кода радужной оболочки: вместо исходной матрицы 100 ключевых точек радужной оболочки (размер пунктирной области - $Nx \times Ny$) берется укороченная матрица размера $Nx/3 \times Ny/3$ (исходная матрица разбивается на блоки 3×3 и, если в блок попадает точка кода, то соответствующее значение укороченной матрицы равняется 1, иначе 0). Таким образом, размер кода заметно уменьшается и равняется 61 байт. Количество неверно определившихся изображений в базе данных [9] при таком выборе кода в случае отсутствия века получилось равным 40 (1.5 %).

5. ЗАКЛЮЧЕНИЕ

В работе предложен алгоритм идентификации человека, использующий ключевые точки радужной оболочки глаза, найденные локальным методом преобразования Эрмита. Этот алгоритм позволяет значительно уменьшать количество хранимых ключевых точек радужной оболочки без сильной потери качества распознавания. Последовательное использование метода знаков и метода ключевых точек позволяет получить надежный метод распознавания. Метод ключевых точек достаточно перспективен для использования в мультибиометрических системах распознавания людей.

Работа выполнена при поддержке ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009 – 2013 годы.

6. СПИСОК ЛИТЕРАТУРЫ

[1] J.B. Martens. “The Hermite transform-theory”, *IEEE Transactions on Acoustics, Speech, and Signal Processing* vol. 38, no. 9, pp. 1595–1606, 1990.

[2] L. Ma, T. Tan, D. Zhang, Y. Wang. “Local Intensity Variation Analysis for Iris Recognition”, *Pattern Recognition,* vol. 37, no. 6, pp. 1287−1298, 2004.

[3] L. Wang, M. Dai. “Extraction of Singular Points in Fingerprints by the Distribution of Gaussian-Hermite Moment”, *IEEE Proc. 1 Int. Conf. DFMA,* pp. 206-209, 2005.

[4] A. Estudillo-Romero, B. Escalante-Ramirez, “The Hermite Transform: An Alternative Image Representation Model for Iris Recognition”, *LNCS*, no. 5197, pp. 86-93, 2008.

[5] Е.А. Павельева, А.С. Крылов, О.С. Ушмаев. "Развитие информационной технологии идентификации человека по радужной оболочке глаза на основе преобразования Эрмита", *Сист. высокой доступности*, №1, 2009, с. 36-42.

[6] F. Wang, J. Han. “Robust multimodal biometric authentication integrating iris, face and palmprint”, *Inform. techn. and control*, vol.37, no.4, pp. 326-332, 2008.

[7] A.K. Jain, A. Ross. “Multibiometric Systems", *Comm. of the ACM*, vol. 47, no. 1, pp. 34-40, 2004.

[8] A. S. Krylov, E. A. Pavelyeva. “Iris Data Parametrization by Hermite Projection Method”, GraphiCon’2007 Conf. proc., p. 147−149, 2007.

[9] База данных CASIA-IrisV3: http://www.cbsr.ia.ac.cn/IrisDatabase.htm.

[10] L. Ma, T. Tan, Y. Wang, and D. Zhang, “Efficient iris recognition by characterizing key local variations” *IEEE Trans. on Image Processing*, vol. 13, no. 6, p. 739–750, 2004.

[11] J. Daugman, “New Methods in Iris Recognition”, *IEEE Transaction on Systems, Man, Cybernetics-part B*, vol. 37, no. 5, pp. 1167-1175, 2007.

Iris identification algorithm using the most informative iris points

Abstract

The algorithm of human iris identification using local Hermite transform is proposed. The most informative Hermite transform functions are used to form the iris code. The compact algorithm of Hermite transform method which uses only key iris points is also proposed. This method allows to decrease considerably the number of used iris points preserving identification reliability.

Keywords: iris recognition, Hermite transform, iris key points, biometrics.

Об авторах

Павельева Елена Александровна – аспирантка ф-та ВМК МГУ. E-mail: Paveljeva@yandex.ru.

Крылов Андрей Серджевич – к.ф.-м.н., доцент, зав. лаб. математических методов обработки изображений ф-та ВМК МГУ. E-mail: kryl@cs.msu.ru.

About the authors:

Elena A. Pavelyeva is a PhD student of Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University (CMC MSU). E-mail: Paveljeva@yandex.ru.

Dr. Andrey S. Krylov is an associated professor, Head of the Laboratory of Mathematical methods of Image Processing, CMC MSU. E-mail: kryl@cs.msu.ru.

Метод оценки качества биометрической идентификации в операционных условиях на примере дактилоскопической идентификации[1]

Ушмаев Олег Станиславович, oushmaev@ipiran.ru,
Институт проблем информатики РАН,
Россия, Москва, ул. Вавилова, д.44, к.2

Арутюнян Артем Рафаэлевич, artem@ibrae.ac.ru,
Институт проблем безопасного развития атомной энергетики, Россия, Москва, ул. Б. Тульская, д.52

Аннотация

В докладе рассмотрена проблема априорной оценки качества биометрической идентификации в условиях реальной эксплуатации. Предложен подход к оценке, основанный на моделировании отдельных факторов, влияющих на качество распознавание. В качестве основной модели предлагается использовать нормальную аппроксимацию биометрической системы. В качестве основного примера приведено исследование автоматической дактилоскопической идентификации. В частности, показано, что качество идентификации значительно зависит от операционных условий.

Ключевые слова: *Биометрическая идентификация, дактилоскопическая идентификация, искажающие факторы, операционные испытания.*

1. ВВЕДЕНИЕ

В настоящее время биометрические технологии получают широкое распространение в различных областях: от контроля и управления доступом в офисные помещения до электронной коммерции и гражданской идентификации [1-3].

С распространением биометрии и увеличением масштаба биометрических систем, актуальной становится проблема выбора того или иного метода биометрической идентификации. Одним из факторов, определяющим выбор, является качество идентификации, а именно соотношение ошибок 1-го и 2-го рода.

Типовым подходом к оценке качества идентификации является измерение ошибок в стандартном тестировании или на стандартном тестовом массиве [4]. Такой подход позволяет провести обоснованный сравнительный анализ технологий. В то же время он во многих случаях не дает оценку качества идентификации в реальных операционных условиях [4]. Основной причиной является влияние различных искажающих факторов.

Чтобы заведомо избежать проблем с качеством идентификации, используют мультибиометрические системы. Даже в случаях, где применение нескольких биометрических идентификаторов не обосновано операционными условиями. Однако, открытым остается вопрос об обоснованности выбора мультибиометрии. А именно, насколько она лучше одного биометрического идентификатора.

В качестве примера рассмотрим следующую ситуацию: систему идентификации по отпечатку пальца предлагается усилить за счет идентификации по голосу. На рис. 1 приведена оценка ошибок мультибиометрической идентификации [5]. Известным способом улучшения качества дактилоскопической идентификации является использование нескольких приложений (предъявлений) пальца на этапе регистрации. Как видно из рис.1, таким образом можно достичь качества идентификации, схожего с вариантом мультибиометрического усиления за счет голосовой биометрии. Но такой подход не требует ни дополнительного времени на сканирование при идентификации личности, ни модернизации аппаратной части.

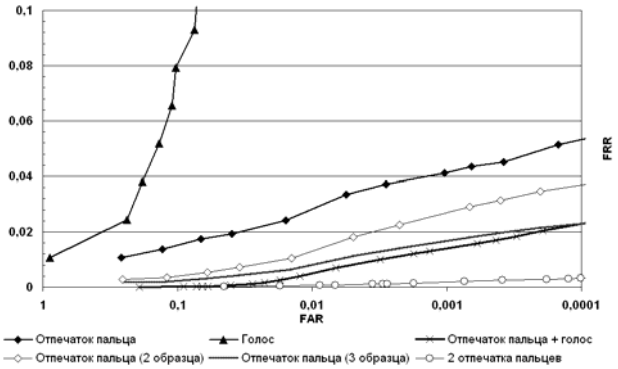


Рисунок 1: Качество идентификации

Вторым примером неправильного оценивания возможностей биометрии является лицевая биометрия. Как показывают результаты тестирований FRVT [6] с 2002 по 2006 года, качество распознавания формально улучшилось в 20 раз. При этом неучтенным остаются различные условия тестирования. В FRVT2006 их можно охарактеризовать, как «лабораторные», а в FRVT2002 как довольно плохие «полевые». В реальной эксплуатации условия будут средними.

Проблемой является отсутствие методики априорной оценки операционных возможностей той или иной биометрии.

В настоящем докладе предложен подход к оценке качества идентификации на основе моделирования искажающих факторов. Основными этапами оценки потенциального качества являются: идентификация искажающих факторов,

[1] Работа поддержана РФФИ (проект 07-07-00031) и Программой ОНИТ РАН «Информационные технологии и методы анализа сложных систем». Работа выполнена в рамках НОЦ ИПИ РАН – ВМК МГУ «Биометрическая информатика».

измерение воздействия фактора, оценка качества с поправкой на искажающий фактор.

Далее доклад организован следующим образом. В настоящем разделе дана постановка задачи. В разделе 2 дан подход к решению поставленной задачи. В разделе 3 приведены результаты применения выработанного подхода к дактилоскопической идентификации. Заключение содержит основные выводы и направления дальнейших исследований.

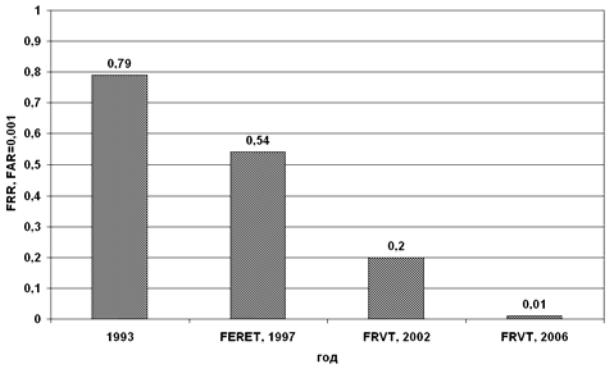


Рисунок 2: Прогресс распознавания по изображению лица.

2. КОЛИЧЕСТВЕННАЯ ОЦЕНКА ВЛИЯНИЯ ИСКАЖАЮЩИХ ФАКТОРОВ

Большинство биометрических систем являются системами распознавания образов [1-3]. Поэтому процесс идентификации состоит из двух процедур: извлечение признаков (создание биометрического шаблона, рис. 3) и сравнение шаблонов.

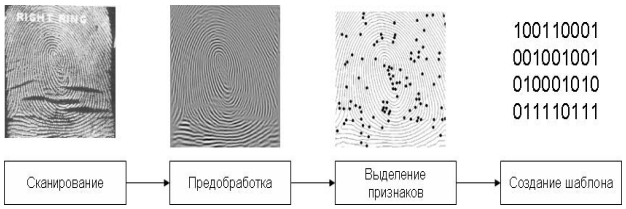


Рисунок 3: Процесс создания биометрического шаблона.

При сравнении шаблонов используются различные, преимущественно пороговые методы принятия решений. Качество идентификации определяется соотношением двух ошибок: ошибка первого рода FRR (результат сравнения двух образцов одного человека, «свои» сравнения, меньше порога) и ошибка второго рода FAR (результат сравнения образцов, принадлежащих разным людям, больше порога).

Под воздействием искажающих факторов происходят смещения в распределениях результатов в «своих» и «чужих» сравнения (рис. 4), что приводит к изменению качества распознавания (рис. 5).

Для количественной оценки влияния искажающего фактора изучим статистически свойства распределений. В качестве примера взята технология BioLink Solutions. Рассмотрим моменты распределений результатов сравнения: математическое ожидание, дисперсию, моменты старших порядков. Чтобы изолировать влияние математического ожидания и дисперсии, заменим моменты старших порядков на производные статистики

$$\gamma_k = \mathbf{E}\left[\left(\frac{s-\mu}{\sigma}\right)^k\right], \qquad (1)$$

где μ – математическое ожидание, σ– среднеквадратическое отклонение.

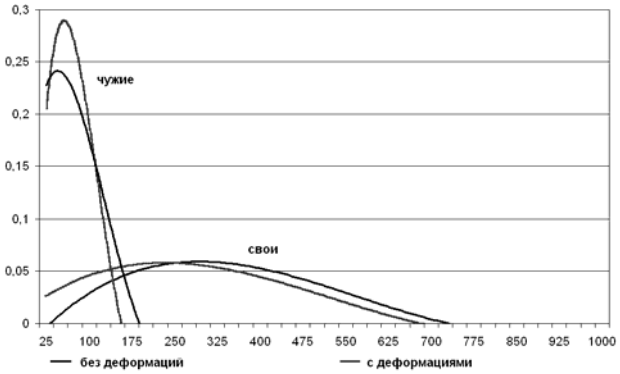


Рисунок 4: Сдвиги в распределениях под воздействием искажающего фактора (деформации отпечатков пальцев).

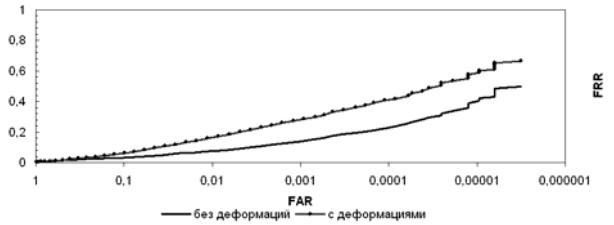


Рисунок 5: Изменение качества идентификации под воздействием деформаций.

Как видно из (1), статистики γ_k инварианты относительно линейной замены аргументов. Данные по изменению статистик под воздействием искажающего фактора приведены в табл. 1.

Таблица 1: Влияния деформаций отпечатков пальцев

	μ	σ	γ_3	γ_4	γ_5	γ_6	γ_7	γ_8
Свои (с компенсацией деформации)	280	143	0,4	2,7	3,3	14,4	35,7	147,5
Свои (без компенсации деформации)	265	156	0,8	3,1	4,9	18,4	49,4	208,3
Чужие (с компенсацией деформации)	48	24	1,1	4,7	15,8	77,2	430,6	2869,3
Чужие (Без компенсации деформации)	46	24	1,1	4,9	16,3	83,0	470,9	3027,3

Анализ данных таблицы 1 показывает, что при грубой оценке данного искажающего фактора можно ограничиться моментами первых двух порядков. Слабая вариация статистик (1) высших порядков указывает на то, что характер распределения не претерпевает принципиальных изменений.
Обозначим первые моменты распределений в своих и чужих сравнениях через (μ_g, σ_g^2) и (μ_i, σ_i^2). Смещения моментов,

сопряженные с воздействием искажающего фактора обозначим через $(\Delta m_g, \Delta s_g^2)$ и $(\Delta m_i, \Delta s_i^2)$ соответственно.

Такая оценка искажений на уровне смещений моментов [7,8] первых двух порядков имеет явное практическое преимущество. Пусть имеется эталонная таблица соответствий ошибок 1-го рода FRR(x) и 2-го рода FAR(x) в зависимости от порога x принятия решения. Тогда скорректированные с учетом искажающего фактора ошибки распознавания определяются следующим образом:

$$\mathrm{FRR}^c(x) = \mathrm{FRR}(\alpha_{\mathrm{FRR}} x + \beta_{\mathrm{FRR}});$$

$$\mathrm{FAR}^c(x) = \mathrm{FAR}(\alpha_{\mathrm{FAR}} x + \beta_{\mathrm{FAR}});$$

где коэффициенты α и β являются коэффициентами линейного преобразования

$$\frac{x - \Delta m - \mu}{\sqrt{\sigma^2 + \Delta s^2}} \to \frac{x - \mu}{\sigma},$$

которые прямо вычисляются через моменты по следующим формулам:

$$\alpha_{\mathrm{FAR}} = \frac{\sigma_i}{\sqrt{\sigma_i^2 + \Delta s_i^2}};\ \beta_{\mathrm{FAR}} = \mu_i - \frac{\sigma_i}{\sqrt{\sigma_i^2 + \Delta s_i^2}}(\mu_i + \Delta m_i); \quad (2)$$

$$\alpha_{\mathrm{FRR}} = \frac{\sigma_g}{\sqrt{\sigma_g^2 + \Delta s_g^2}};\ \beta_{\mathrm{FRR}} = \mu_g - \frac{\sigma_i}{\sqrt{\sigma_g^2 + \Delta s_g^2}}(\mu_g + \Delta m_g); \quad (3)$$

В условиях воздействия нескольких некоррелированных факторов с параметрами $(\Delta m_{jg}, \Delta s_{jg}^2)$, $(\Delta m_{ji}, \Delta s_{ji}^2)$ итоговое воздействие $(\Delta m_g, \Delta s_g^2)$ и $(\Delta m_i, \Delta s_i^2)$ получается суммой отдельных факторов:

$$\Delta m_g = \sum_j \Delta m_{jg};\ \Delta s_g^2 = \sum_j \Delta s_{jg}^2; \quad (4)$$

$$\Delta m_i = \sum_j \Delta m_{ji};\ \Delta s_i^2 = \sum_j \Delta s_{ji}^2. \quad (5)$$

Сдвиги ошибок FRR(x) и FAR(x) вычисляются по всей совокупности факторов. Формально можно ожидать, что некоторые факторы уменьшат дисперсию, и в таком случае величина Δs^2 окажется отрицательной. Примером подобного фактора является, например, уменьшение окна сканирования. Уменьшение дисперсии в таком случае обусловлено общим снижением информативности получаемых биометрических образцов.

3. ПРИМЕНЕНИЕ

В качестве примера использования рассмотрим дактилоскопичечкую идентификацию. Рассмотрим факторы, влияющие на качество распознавания отпечатков пальцев. Выделим следующие [9]:

- шумы;
- внешние условия (температура, влажность);
- временной лаг между регистрацией и идентификацией;
- деформации отпечатков пальцев;
- характеристики типичного пользователя системы;
- различия в способе получения отпечатков пальцев (например, след отпечатка, «живое» сканирование и оцифровка бумажных носителей).

Можно предположить, что перечисленные факторы по своему воздействию независимы. Поэтому можно проводить их оценку по отдельности. В рамках настоящего доклада ограничимся рассмотрением трех факторов: тип сканера (I), деформации (II) и способ сканирования (III). Такой выбор обусловлен возможностью экспериментальной верификации на публично доступных базах отпечатков пальцев. На рис.6 приведены гистограммы распределений.

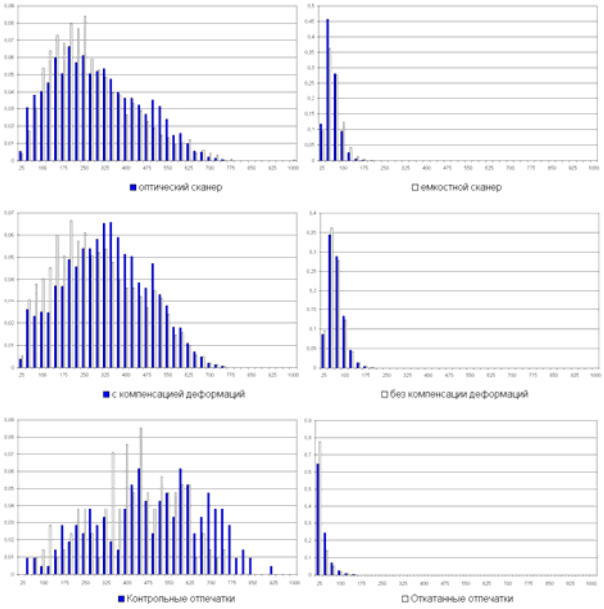


Рисунок 6: Изменение распределений

Численные результаты оценки моментов искажающих факторов и поправочных коэффициентов приведены в таблицах 2 и 3. На рис. 7 приведены графики нормальной аппроксимации искажающих факторов [7,8].

Таблица 2: Оценки моментов

	μ_g	σ_g^2	μ_i	σ_i^2	$\Delta\mu_g$	Δs_g^2	$\Delta\mu_i$	Δs_i^2
I	265	24366	46	580	-16	4039	-2	309
II	265	24366	46	580	15	-3887	2	21
III	265	24366	46	580	-15	-4684	-5	-27

Таблица 3: Поправочные коэффициенты

Эксперименты	α_{FRR}	β_{FRR}	α_{FAR}	β_{FAR}
I	0,926	34,402	0,807	10,460
II	1,091	-40,454	0,982	-1,154
III	1,117	-81,033	1,024	14,252

На рис. 8 приведены оценки качества распознавания в наилучших и наихудших относительно приведенных факторов операционных условиях соответственно. Как видно из рисунка, зависимость качества распознавания от операционных условий является значительной.

4. ЗАКЛЮЧЕНИЕ

В докладе предложен подход к априорному определению качества идентификации в реальных условиях на основе моделирования искажающих факторов. Для случая дактилоскопической идентификации проведено количественное оценивание факторов. Показано, что качество идентификации значительно зависит от операционных условий.

Направлением дальнейших исследований является применение разработанного метода к другим биометрическим технологиям: изображение лица, радужная оболочка глаза и т.д.

5. СПИСОК ЛИТЕРАТУРЫ

[1] I.N. Sinitsyn, O.S. Ushmaev, *Development of metrological and biometric technologies and systems* // Proceedings of 9th International Conference in Pattern Recognition and Image Analysis: new information technologies, PRIA-9-2008, September, 14-20, 2008. – Нижний Новгород, 2008, v.2, p.169-172.

[2] Dessimoz D., Champod C., Richiadi J., Drygajlo A. *Multimodal Biometrics for Identity Documents* // Research Report, PFS 314-08.05. UNIL, June 2006.

[3] Bolle R. M., Connell J. H., Pankanti S., Ratha N. K., Senior A. W. *Guide to Biometrics*. – New-York: Springer-Verlag, 2003.

[4] Wayman J., et al. *Biometric Systems: Technology, Design and Performance Evaluation*. – London: Springer Verlag, 2005.

[5]О.С. Ушмаев, А.В. Босов, *Реализация концепции многофакторной биометрической идентификации в интегрированных аналитических системах* // Системы высокой доступности, 4, 2007, т.3, с.13-23.

[6] *Face Recognition Vendor Test*. http://www.frvt.org.

[7] Ushmaev O. S., Novikov S. O. *Integral criteria for large-scale multiple fingerprint solutions* // In Biometric Technology for Human Identification, edited by Anil K. Jain, Nalini K. Ratha, Proceedings of SPIE. Vol. 5404 – SPIE, Bellingham, WA, 2004. P. 534–543.

[8] О.С. Ушмаев, *Стохастические технологии мультибиометрической идентификации* // Обозрение прикладной и промышленной математики, т. 16, вып.2, с. 393.

[9] *Fingerprint Vendor Technology Evaluation*. http://fpvte.nist.gov.

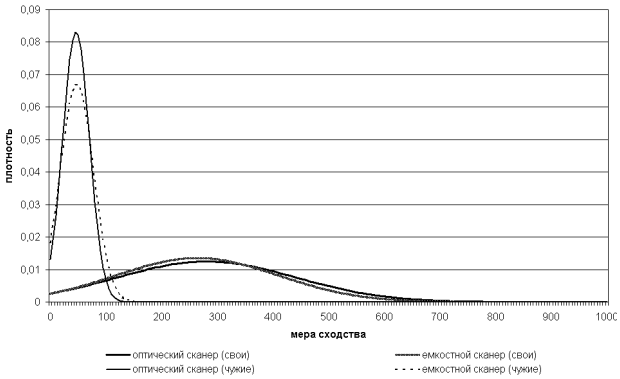
Сведения об авторах

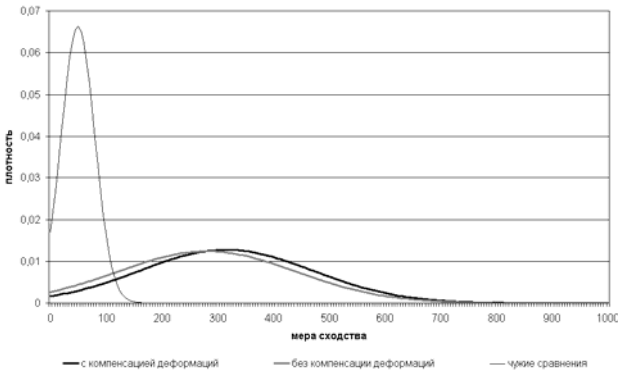
Ушмаев Олег Станиславович, (1981 г.р.), канд. техн. наук, ведущий научный сотрудник Института проблем информатики РАН.

Область научных интересов: информационно-аналитические комплексы и системы, обработка изображений, распознавание образов, биометрические технологии.

Арутюнян Артем Рафаэлевич, научный сотрудник отделения экологической безопасности и радиационного риска Института проблем безопасного развития атомной энергетики (ИБРАЭ РАН).

Область научных интересов: биометрические технологии, анализ и автоматизация бизнес-процессов, создание web-приложений.





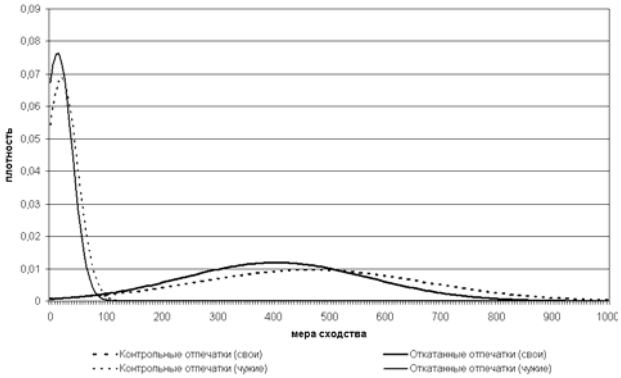


Рисунок 7: Нормальная аппроксимация искажающих факторов

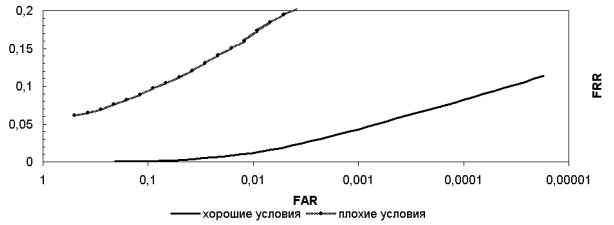


Рисунок 8: Неопределенность в качестве распознавания

Квалиметрия метода эволюционного морфинга

В.И. Протасов
Институт физико-технической информатики, Протвино.
protonus@yandex.ru

Аннотация

Разработана технология измерения точности составления субъективного портрета (фоторобота) с использованием эволюционного морфинга. В ней используется методика измерения способностей свидетелей к распознаванию лиц и составлению субъективного портрета.

Ключевые слова: *Ггенетические алгоритмы, субъективный портрет, эксперт, популяция решений, квалиметрия, точность, способности человека, виртуальный эксперт.*

1. ВВЕДЕНИЕ

Составление субъективного портрета (фоторобота) одиночным свидетелем или группой свидетелей является сложной задачей с непредсказуемыми результатами. Проблема, ввиду ее чрезвычайной важности в криминалистике, имеет давнюю историю и обширную библиографию. В настоящее время в практике составления субъективных портретов широко применяются трехмерные компьютерные модели человеческих лиц [1-2]. Непосредственное использование графических редакторов человеческих лиц, как и полицейских художников, помогающих свидетелям фиксировать образы преступников, не всегда позволяет получать достоверные портреты. Это связано с тем, что люди, как правило, являются плохими художниками. Метод эволюционного морфинга, когда свидетели (свидетель) работают с популяцией 3D образов, используя отбор, скрещивание и мутацию сгенерированных ими образов первого и последующих поколений, дает существенно лучшие результаты [3-6]. Это объясняется тем, что в последнем случае используются отличные способности людей к распознаванию ранее виденных ими людей. Эволюционный морфинг, использующий даже не очень точные первые приближения, помогает «вырастить» популяцию образов, похожих на исходный. Возникает вопрос об измерении степени этого соответствия. Необходимо выработать процедуру, позволяющую сравнивать различные методики составления субъективных портретов.

Известно высказывание, принадлежащее Д.И. Менделееву: «Наука начинается с тех пор, как начинают измерять. Точная наука немыслима без меры». Для измерения способностей человека к какому-либо виду деятельности применяют обычно специальные тесты, либо он сдает квалификационные экзамены. Оба способа обладают субъективизмом, поскольку тесты, составленные разными экспертами, дают различные результаты. Для получения доказательных утверждений в нашем случае необходимо, следовательно, получение достоверных, числовых измерений результатов составления субъективных портретов. Эти измерения могут быть использованы в дальнейшем для улучшения характеристик и эффективности применяемых методов.

2. МОДЕЛЬ ВИРТУАЛЬНОГО СВИДЕТЕЛЯ

В настоящей работе предпринята попытка осуществить квалиметрию процесса составления субъективного портрета коллективом или одиночным свидетелем с использованием технологии эволюционного морфинга. Для достижения этой цели необходимо было разработать модель виртуального свидетеля. Модель должна полностью замещать реального свидетеля таким образом, чтобы результаты деятельности виртуального свидетеля были неотличимы от результатов деятельности реального свидетеля с такими же числовыми характеристиками. По сути, должен быть сформирован искусственный интеллект, заменяющий реального свидетеля для целей тестирования метода и измерений.

Виртуальный свидетель представляет собой программу, содержащую ряд параметров, характеризующих основные свойства реального свидетеля, пытающегося восстановить и зафиксировать портрет ранее виденного им человека. В первом приближении эти свойства можно описать тремя параметрами. К ним относятся величина K_a, характеризующая способности свидетеля, как художника и величины K_0 и α, описанные ниже и характеризующие способности свидетеля к распознаванию лиц. Образ лица, «вспоминаемого» виртуальным свидетелем, моделируется набором n относительных величин G_i, $i=1,2,...n$. Эти величины однозначно определяют трехмерное изображение лица в виде полигональной модели. Так, в применяемой нами программе FaceGenModeller [7], для описания симметричного лица без текстуры $n=50$. Если мы ставим перед виртуальным свидетелем задачу: по хранящемуся у него образу «нарисовать» трехмерный портрет лица, характеризующегося вектором G_i, то программа, моделирующая виртуального свидетеля, восстановит искаженный вектор-образ U_i этого лица по формуле

$$U_i = G_i[1 + K_a\chi(1-2\xi)] \qquad (1),$$

где χ и ξ - случайные числа от 0 до 1. Случайное число χ обеспечивает особенность свидетеля каждый раз рисовать разные портреты, отличающиеся от оригинала, но с постоянным, характеризующим данного свидетеля коэффициентом сходства.

Эксперименты показали, что такая модель достаточно реалистично описывает художественные способности человека. Так, при $K_a = 0$ мы имеем идеального художника, с ростом же этой величины художественные способности падают. На рис. 2 приведены портреты, «нарисованные» виртуальными свидетелями с разными художественными способностями.

Для оценки качества «нарисованного» портрета введем два коэффициента K_R и K_S следующим образом

$$K_R = \frac{1}{n\{G_{\max} - G_{\min}\}} \sum_{i=1}^{n} |U_i - G_i| \qquad (2),$$

здесь K_R - коэффициент различия двух портретов, $G_{\max}$ - максимальное и $G_{\min}$ - минимальное из возможных значений вектора G_i,

$$K_S = 1 - K_R \qquad (3),$$

K_S - коэффициент сходства.

Величины коэффициентов K_R и K_S, как видно из формул (2) и (3), расположены между 0 и 1. Из анализа модельных экспериментов, между величинами K_R и K_a существует прямая пропорциональная зависимость

$$K_R = K_a / b \qquad (4),$$

где $1/b$ - коэффициент пропорциональности, зависящий от диапазона значений компонент вектора G_i.

По аналогии с коэффициентом художественных способностей вводится коэффициент способности свидетеля к распознаванию лиц K_0. Этот коэффициент определяется из эксперимента по распознаванию свидетелем ряда сгенерированных программой лиц, в разной степени похожих на искомое. Свидетель выбирает из этой серии наиболее похожее, по его мнению, лицо. Поскольку программе известно значение коэффициента различия, то после серии таких экспериментов, программа рассчитывает среднее значение этого коэффициента. Это среднее значение и характеризует способность свидетеля, как распознавателя.

Процедура определения коэффициента способности свидетеля к распознаванию лиц K_0 осуществляется следующим образом. Программа генерирует две серии портретов с различными значениями величины K_a, отличающимися в друг от друга в два раза. Допустим K_a=0.1 и 0.2. Если свидетель выбирает из этой серии портрет с K_a=0.1, то в следующей серии ему будет предложены две серии с K_a=0.08 и 0.06. Если же он выберет в качестве наилучшего портрета портрет с 0.1< K_a <0.2, то ему будет предложена серия с Ka =0.14 и 0.18. При выборе K_a=0.2, ему предложат серию с K_a=0.24 и 0.28. Методом последовательных приближений может быть определено более точное значение этой величины и после ее усреднения на различных лицах, выносится заключение о способности свидетеля к распознаванию лиц.

Величина K_0 рассчитывается по формуле.

$$K_o = 1 - K_{acp} / b, \qquad (5)$$

где K_{acp} - усредненное значение K_a.

При длительных экспериментах, когда перед свидетелем проходит большая совокупность распознаваемых лиц (эта ситуация характерна для работы свидетеля при составлении субъективного портрета с использованием метода эволюционного морфинга), свидетель начинает чаще ошибаться и его способность к распознаванию снижается.

Эту способность свидетеля можно охарактеризовать величиной «утомляемости» α, определяемой из выражения:

$$\alpha = -\frac{1}{N} \ln \frac{K_{oN}}{K_o}, \qquad (6)$$

где K_{oN} - значение коэффициента K_0 после предъявления *N* портретов в течение одного сеанса экспериментов.

Выражение (6) получено из формулы

$$K_{oN} = K_o e^{-\alpha N}. \qquad (7)$$

Типичные значения величин коэффициентов K_a, K_0 и α, полученные из экспериментов, проведенных с разными людьми, приведены в таблице 1. Таблица 1

Значения коэффициентов K_a, K_0, α

Коэффициент	Минимальное Значение	Максимальное значение
K_a	0.02	0.3
K_0	0.92	0.99
α	0.00003	0.000009

3. НАСТРОЙКА ПАРАМЕТРОВ ЭВОЛЮЦИОННОГО МОРФИНГА

Наряду с получением аппарата измерений креативных способностей свидетеля к процессу составления субъективного портрета, целью создания концепции виртуального свидетеля является также разработка механизма настройки параметров программы, реализующей эволюционный морфинг. Настройка параметров должна осуществляться для конкретного набора свидетелей, обладающих различными способностями. Эти способности должны быть предварительно измерены. Для оптимизации параметров интеллектуального консилиума реальные свидетели заменяются виртуальными, обладающими теми же характеристиками. Анализ задачи показывает, что оптимизируемая величина – степень соответствия составленного экспертами портрета исходному, в пространстве поиска оптимальных параметров не может быть выражена через них в явной форме. Поэтому для проведения оптимизации были выбраны генетические алгоритмы. Параметрами настройки эволюционного морфинга являются величины KPM – число «прогонов» программы для получения установившихся значений коэффициентов сходства, NC – количество скрещенных портретов, выдаваемых программой виртуальному свидетелю для выбора, NM – количество мутированных портретов из одного выбранного портрета после

скрещивания, PM – параметр мутации, или относительная величина изменения генов выбранного портрета (рис. 1).

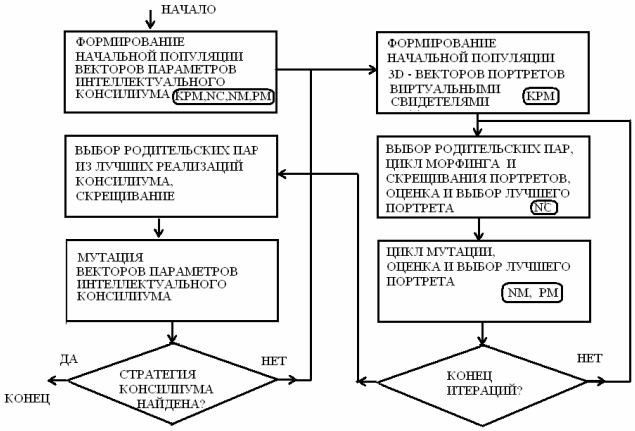


Рис.1. Блок-схема настройки параметров эволюционного морфинга.

Полученная процедура позволяет заменять медленно работающих реальных свидетелей виртуальными для настройки параметров интеллектуального консилиума. Если время составления одного варианта субъективного портрета реальными свидетелями составляет от получаса до полутора часов, то виртуальные эксперты имитируют составление портрета за доли секунды

4. ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

Для проверки модели виртуального свидетеля и настройки параметров были проведены следующие эксперименты.

В первой группе экспериментов была проведена проверка сходимости метода эволюционного морфинга независимо от начальных условий – виртуальным свидетелям были заданы низкие коэффициенты художественных способностей. Несмотря на это, метод сошелся к восстановленному портрету с величиной коэффициента сходства 0.85 (рис.2)

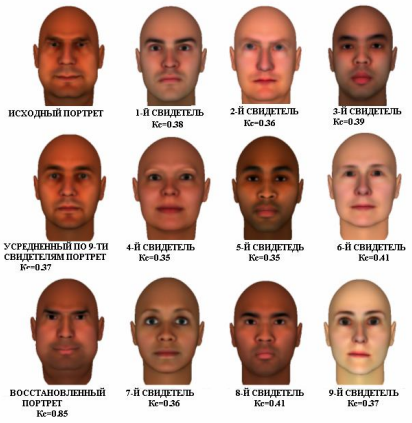


Рис. 2. Результаты эксперимента по проверке сходимости метода эволюционного морфинга независимо от качества начального приближения.

На рис. 3 приведены результаты сходимости метода для группы из девяти виртуальных свидетелей при пяти различных начальных приближениях. Несмотря на то, что коллективы виртуальных свидетелей «стартовали» из различных начальных популяций портретов, они приходили к высоким коэффициентам сходства восстановленного портрета с исходным.

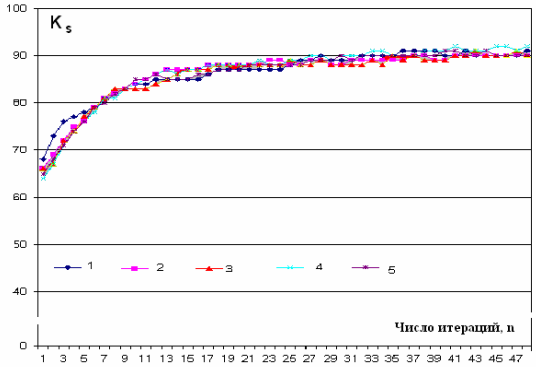


Рис. 3. Сходимость метода эволюционного морфинга

Во второй группе экспериментов с виртуальными свидетелями осуществлялась проверка влияния параметров настройки эволюционного морфинга на качество восстанавливаемого портрета. Для данного коллектива из девяти виртуальных свидетелей с известными параметрами K_a, K_0 и α было испытано 8 различных вариантов составления портретов для различных наборов параметров эволюционного морфинга KPM, NC, NM и PM. Результаты экспериментов приведены на рис.4. Видно, что наблюдается значительное влияние параметров эволюционного морфинга на конечный результат распознавания и, следовательно, является доказанной необходимость предварительной настройки параметров метода для получения лучшего качества составления субъективного портрета.

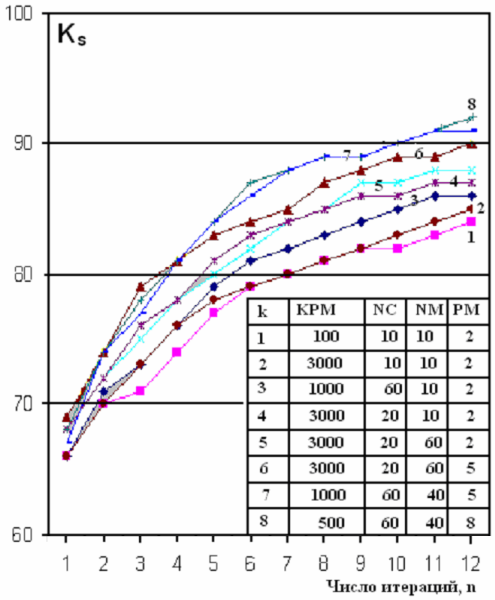


Рис. 4. Влияние параметров настройки эволюционного морфинга на результаты составления субъективного портрета коллективом виртуальных свидетелей.

Проведенные эксперименты позволили сформулировать правила настройки параметров метода составления субъективного портрета в следующем виде.

Свидетели тестируются для определения численных значений их способностей. Им предъявляются тестовые примеры, при работе над которыми измеряются значения их художественных и распознавательных способностей. На этой стадии может происходить отсев свидетелей, обладающих низкими результатами работы по составлению портретов. Далее эти параметры вводятся в программу настройки параметров эволюционного морфинга, и программа работает с виртуальными свидетелями, обладающими свойствами реальных. Для этого коллектива программа настраивает свои параметры, и свидетели могут приступать к работе по составлению искомого портрета. В соответствии с этими правилами был организован эксперимент с участием пяти свидетелей. Вначале они были протестированы по приведенной выше методике. Способности свидетелей характеризовались параметрами $0.03 < K_a < 0.12$, $0.93 < K_0 < 0.98$ и $0.00004 < \alpha < 0.00007$. Затем, после двухчасового перерыва им на непродолжительное время были предъявлены фотографии фас и профиль неизвестного им лица (рис. 5 а). Далее они с использованием методики интеллектуального консилиума составляли субъективный портрет. Усредненный результат по пяти свидетелям после первой итерации приведен на рис. 5 б). На рис. 5 в). приведены результаты составления субъективного портрета этой группой. Для составления субъективного портрета с $K_s = 0.92$ потребовалось всего 6 итераций.

Виртуальным свидетелям с характеристиками реальных свидетелей был предъявлен для «запоминания» и последующего составления портрет, восстановленный реальными свидетелями. Эксперименты показали, что разброс величин K_s от 0.89 до 0.93, полученный при этих экспериментах (их было проведено 200), хорошо коррелирует с величиной K_s, полученной для коллектива реальных свидетелей. Число итераций от 6 до 7, полученных в группе контрольных экспериментов также хорошо коррелирует со случаем реальных экспертов.

а) исходный б) усредненный в) составленный портрет

Рис. 5. Результаты составления субъективного портрета группой из пяти свидетелей

5. ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ

Метод эволюционного морфинга, описанный в [3-4], отличается от предложенного нами в том, что там свидетели работают поодиночке, и в качестве окончательного результата используется усредненный портрет. В нашем случае свидетели работают над портретом синхронно, обмениваясь своими вариантами, вследствие чего возникает синергетический эффект усиления интеллекта и результаты составления субъективного портрета обладают большей точностью. С использованием коллектива виртуальных свидетелей была проведена сравнительная проверка нашего метода и метода, описанного в [4]. Для случая, приведенного выше, величина коэффициента сходства Ks у нас составляла 0.92, для метода, описанного в [4], Ks=0.86. Эксперименты с различными коллективами виртуальных свидетелей и исходными портретами показали, что коэффициенты сходства у метода [4] лучше, как и следовало ожидать, чем усредненные коэффициенты у нас после первой итерации, но всегда хуже, чем в нашем методе после шести итераций.

Итак, можно считать достоинством метода настраиваемого эволюционного морфинга то, что при составлении субъективного портрета мы можем получить квалиметрические характеристики как самого метода, так и его результатов. С использованием концепции виртуальных свидетелей мы можем также сравнивать различные методы составления субъективных портретов по точности.

6. ЗАКЛЮЧЕНИЕ

В результате проделанной работы были сделаны следующие выводы:

- Получены модельные методики для определения способностей свидетелей к рисованию портретов.
- Получены методики для определения способностей свидетелей к распознаванию лиц и коэффициентов утомляемости при длительной работе.
- Разработанная модель виртуального свидетеля может применяться для экспериментов по отладке перспективных схем составления субъективного портрета.
- Достоинством метода настраиваемого эволюционного морфинга является то, что при составлении субъективного портрета мы можем получить квалиметрические характеристики, как самого метода, так и его результатов.

Работа выполнена по гранту РФФИ 08-07-00447а «Разработка и исследование методов оценки достоверности результатов восстановления объемных фотороботов на основе общедоступных установок виртуальной реальности».

7. ЛИТЕРАТУРА

[1] Kovera M.B., Penrod S.D., Pappas C. & Thill D.L. *Identification of computer generated facial composites*. Journal of Applied Psychology, 82, 235-246, 1997.

[2] Mcquiston-Surrett D., Topp L. D., & Malpass R. S. *Use of facial composite systems in U.S. law enforcement agencies*. Psychology, Crime and Law, 12, 505–517, 2006.

[3] Hancock P.J.B. *Evolving faces from principal components. Behavior Research Methods*. Instruments, & Computers,32 (2), 327-332, 2000.

[4] Frowd C.D., Hancock P.J.B., & Carson D. *EvoFIT: A Holistic, Evolutionary Facial Identification Technique for Creating Composites.* Association for Computing Machinery Transactions on Applied Psychology , 1 (1), 1-21, 2004.

[5] В.И. Протасов, Ю.Ю. Здоровеющев, Д.С. Панфилов. *Об одном методе коллективного принятия решений при построении фоторобота.* Труды международной научно-технической конференции «Интеллектуальные многопроцессорные системы ИМС-99», Таганрог, 222-225, 1999.

[6] В.И. Протасов, А.А. Дружинин, Л.В. Михайлов. *Методика восстановления субъективного портрета коллективом свидетелей с использованием 3D-морфинга.* Программные продукты и системы, Тверь, 1(77), 21-24, 2007.

[7] FacegenModeller 3.1.
http://www.facegen.com/modeller.htm

Автор

В.И. Протасов, кандидат физико-математических наук, научный сотрудник Института физико-технической информатики, Протвино, Московская обл.

Email: protonus@yandex.ru

Автоматическая сегментация облаков точек на основе элементов поверхности

A. B. Velizhev, R.V. Shapovalov, D.Potapov, L.Tretiak, A. Konushin
Department of Computational Mathematics and Cybernetics
Moscow State University, Moscow, Russia
avelizhev@graphics.cs.msu.ru, shapovalov@graphics.cs.msu.ru, potapov@graphics.cs.msu.ru, tretiak@graphics.cs.msu.ru, ktosh@graphics.cs.msu.ru

Аннотация

Алгоритмы сегментации результатов лазерного сканирования, работающие с отдельными точками, чувствительны к шуму и требуют значительных вычислительных ресурсов. В последние годы были предложены алгоритмы сегментации, основанные на использовании иерархических деревьев, таких как kd- или окто- деревья. Однако, при построении таких деревьев исходное множество трехмерных точек разбивается с помощью плоскостей, поэтому на сложных поверхностях возникают разрывы. В результате точки принадлежащие непрерывной поверхности попадают в разные листы дерева, что ухудшает результаты сегментации. В данной статье мы предлагаем новую иерархическую структуру данных, названную Seg-Tree, и показываем её эффективность на примере популярного алгоритма разрастающихся регионов. Предлагаемая модификация этого алгоритма более устойчива к шуму и демонстрирует стабильные результаты сегментации при меньшей зависимости от углового порога.

Ключевые слова: *лазерное сканирование, облако точек, структура данных, сегментация, элементы поверхности.*

1. ВВЕДЕНИЕ

Лазерное сканирование широко используется для детального моделирования трёхмерных сцен. Важным этапов в решении задач построения параметрической модели, классификации и анализе формы объектов является сегментация облака точек. Целью сегментации является разбиение неструктурированного облака точек на не пересекающие подмножества точек. Все точки внутри каждого сегмента должны быть близки друг к другу в выбранном пространстве признаков. Типичная задача сегментации - выделение плоскостей в результатах лазерного сканирования, в этом случае все точки внутри каждого сегмента лежат в одной плоскости.

В данной статье мы предлагаем новую иерархическую структуру данных, названную Seg-Tree, и показываем её эффективность на примере задачи сегментации, модифицировав популярный алгоритм разрастающихся регионов [6]. Предлагаемая модификация более устойчива к шуму и демонстрирует стабильные результаты сегментации при меньшей зависимости от углового порога. Seg-Tree создается в процессе специальной процедуры пересегментации и группирует точки в компактные элементы поверхности как для гладких, так и для сильно зашумленных поверхностей. Процедура пересегментации основана на построении R-дерева (R-Tree) [4] в сочетании с алгоритмом k-средних для разбиения узлов дерева.

Статья организована следующим образом. В разделе 2 приведен обзор существующих иерархических структур и методов автоматической сегментации трехмерных точек. Алгоритм построения Seg-Tree рассмотрен в следующем разделе. В разделе 4 предлагается улучшенная версия алгоритма разрастающихся регионов, основанная на Seg-Tree. Результаты и выводы приведены в последнем разделе.

2. ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ

В данном разделе мы рассмотрим основные структуры данных для хранения и обработки точек лазерного сканирования, а также приведем обзор существующих алгоритмов сегментации трехмерных точек.

2.1 Иерархические структуры данных

Для обработки и хранения облаков точек наибольшее распространение получили окто- и kd- деревья [5]. Для таких деревьев были разработаны вычислительно эффективные методы поиска k-ближайших соседних точек [7].

В 2006 были предложены деревья элементов поверхности [2], которые позволили представить неструктурированное облако точек в виде набора элементов поверхности. Одним из главных недостатков этого дерева элементов поверхности является некорректное разбиение для некоторых типов поверхности, которое проявляется при работе со сложными и зашумленными облаками точек (например, для естественных сцен). Однако, идея элементов поверхности является на наш взгляд крайне полезной в контексте различных задач обработки облаков точек, например, задачи сегментации.

Другим подходом к построению пространственного индекса для хранения геометрических примитивов является подход на основе R-дерева, предложенный в работе [4]. Данный подход основан на построении иерархии вложенных параллелепипедов. Важным свойством R-деревьев является возможность взаимного пересечения узлов и листьев дерева. Каждый узел может содержать ограниченное число вложенных узлов дерева, а точки хранятся только в листьях дерева. Таким образом, R-деревья представляют собой вариант пространственного индекса, в отличие от окто-деревьев добавление и удаление точек является быстрой операцией.

В случае, когда число элементов в узле R-дерева превышает заданное, происходит подразбиение узла. Гутман предложил три разных алгоритма разбиения, которые пытаются минимизировать сумму площадей новых узлов (для плоского дерева). Однако, этот критерий не всегда является

удовлетворительным, поэтому в работе [3] было предложено рассматривать задачу разбиения как классическую задачу кластеризации данных. Авторы использовали широко известный алгоритм к-средних для разбиения узлов, назвав полученный пространственный индекс cR-деревом. Применив такое разбиение к облаку точек, мы получили наборы из групп точек близкие к элементам поверхности. Что позволило использовать данную структуру данных для задачи сегментации (подробнее см. раздел 3).

2.2 Сегментация облаков трехмерных точек

Задача сегментации трехмерных облаков точек является популярной областью исследований в последние годы. В данном обзоре мы сконцентрировали свое внимание на полностью автоматических алгоритмах сегментации. Более детальный обзор алгоритмов сегментации можно найти в работе [1].

Современные лазерные сканеры позволяют получить высокую детальность на поверхности объекта съемки. Полученные наборы точек затем сегментируются на основе анализа геометрических свойств точек. Например, в статье [6] в качестве признаков использовались направление нормали и величина отклонения точки от локальной плоскости. Авторы выполняли сегментацию с помощью алгоритма разрастающихся регионов всего с несколькими порогами, задаваемыми вручную. Предложенный метод позволял получить корректную сегментацию сглаженных поверхностей. Применив данный алгоритм, к реальным данным мы столкнулись с проблемой возникновения большого числа сегментов, состоящих всего из нескольких точек. Данный эффект проявлялся как на плоских поверхностях при наличии шума, так и на сложных поверхностях (например, барельеф). На практике часто бывает сложно найти баланс между недо- и пересегментацией за счет ручного изменения порогов. Тем не менее, данный алгоритм позволяет выполнять сегментацию произвольных типов поверхности, что является несомненным достоинством этого алгоритма. Ряд работ по автоматической сегментации облаков точек используют kd- и окто- деревья в качестве основы (Wang et al., 2004, Bucksch et al., 2006). В работе (Biosca et al., 2008) предложили использовать для сегментации облака точек алгоритм нечёткой кластеризации к-средних. Однако, указанная методика применима только для сегментации плоскостей.

Подводя итоги, можно выделить несколько главных особенностей текущих алгоритмов сегментации. Во-первых, большинство из ранее предложенных методов выполняют сегментацию только плоских участков. Во-вторых, часть методов используют для сегментации структуру иерархического дерева, в котором хранятся точки. Безусловным достоинством данного подхода является возможность вычислительно эффективной реализации алгоритма сегментации для больших массивов исходных данных. Однако, неудачное разбиение пространства точек, используемыми структурами данных, автоматически переходят и на результаты сегментации. В-третьих, применение методов кластеризации позволяет получить сегментацию устойчивую к шуму. Однако, применение таких методов ограничивается плоскими поверхностями.

3. SEG-TREE

Исследование структур организации данных мы начали с реализации R-дерева [4] и обнаружили, что оно не удовлетворяет нашим требованиям. Гутман успешно решил задачу построения пространственного индекса (R-деревья активно используются в различных CAD-системах), однако для задачи пересегментации облака точек такой подход не работает и также приводит к разрывам поверхности. У базового алгоритма R-дерева могут быть изменены следующие свойства:

- выбор листа для добавления точки;
- алгоритм разбиения;
- используемая метрика пространства.

Рассмотрим детальнее эти свойства.

3.1 Выбор листа

Главный критерий качества дерева по Гутману является сумма объемов всех листов. Поэтому точка добавляется в тот лист, объем которого увеличится меньше. Иными словами, для добавления точки будет выбран лист с наименьшим значением *minDist* (Рис. 1)

Такой подход неприемлем для облаков точек, по причине того, что точки часто организованы вдоль отдельных полос, что вызвано техническими особенностями процесса сканирования. В результате в листы дерева попадают точки, принадлежащие одной полосе, что противоречит идее элементов поверхности. Однако, объем таких листов оказывается наименьшим. Решением проблемы может стать рандомизированное добавление точек в дерево, что делает процесс добавление точек не последовательным и на практике не всегда устраняет проблему.

Поэтому мы предлагаем наш собственный метод выбора листа для добавления точек. Для получения листов, точки в которых принадлежат элементам поверхности, необходимо выбирать лист, центр масс которого находится ближе всего к данной точке. Проверка всех листов дерева требует больших вычислительных затрат, поэтому мы использовали приближенный алгоритм, который сохранял баланс между качеством формы листа и вычислительной сложностью.

Алгоритм состоял в последовательном спуске от корня дерева к его листам. На каждом уровне дерева выбирались некоторые узлы дерева, затем на следующем уровне иерархии выбирались некоторые потомки выбранных ранее узлов. Наконец, на последнем уровне выбирался тот лист, центр масс которого оказывался ближе всего к добавляемой точке. Критерий выбора узлов дерева на промежуточных уровнях состоял в следующем. Если точка попадала внутрь ограничивающего параллелепипеда, то узел добавлялся в список. Для улучшения качества конечного разбиения в список также добавлялись несколько узлов с наименьшими значениями *Dist* и/или *minDist*. *Dist* задается расстоянием до центра масс узла, который определяется как среднее между центрами параллелепипедов ближайших потомков узла. Выбор количества узлов для добавления с список позволяет находить баланс между качеством конечной пересегментации и производительностью.

В ходе экспериментов мы остановились на следующем эмпирическом подходе. На каждом уровне мы составляли список ближайших потомков всех выбранных на предыдущем уровне узлов. Сначала в новый список добавлялись все узлы,

у которых *minDist* == 0 (точка оказывается внутри узла). Если таких узлов не оказывалось, то в список добавлялся узел с минимальным значением *minDist.* Затем мы вычисляли *pruneDist* – расстояние до ближайшего центра масс отобранных ранее узлов. Наконец, в список добавлялись все узлы, *minDist* которых оказывался меньше, чем *pruneDist*.

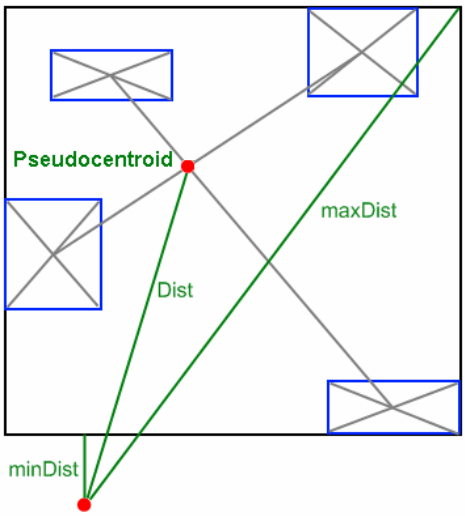


Рисунок 1: Иллюстрация понятий *Dist, MinDist* и *MaxDist*.на примере узла дерева и 4 вложенных потомков.

3.2 Алгоритм разбиения

Гутман предложил три разных алгоритма разбиения, но каждый из них пытался минимизировать сумму объемов новых узлов. Альтернативный подход [3] был основан на рассмотрении задачи разбиения как классической задачи кластеризации данных и использовал для разбиения алгоритм к-средних. Экспериментальные результаты показывают, что использование алгоритма к-средних для подразбиений точек (и даже узлов) позволяет получить листы дерева, являющиеся удачным приближением к элементам поверхности.

3.3 Метрика пространства

Как отмечалось ранее, прямоугольная форма узлов дерева оказывается практичнее, чем сферическая. Однако, использование манхэттенской метрики (расстояния в пространстве L^1) позволяет одновременно сочетать достоинства сферических M-деревьев и прямоугольных R-деревьев.

3.4 Seg-Tree

Наконец, мы решили выбрать следующие параметры R-дерева, соответствующие нашей структуре данных Seg-Tree. Для добавления точек мы используем адаптивный алгоритм выбора листов дерева, описанный в разделе 3.1, который одновременно учитывает *minDist* и *Dist.* Мы использовали алгоритм кластеризации к-средних для подразбиения узлов и листов дерева. Мы допускали от 1 до 64 точек в листах и от 4 до 16 элементов в узлах дерева. Мы использовали манхэттенскую метрику внутри процедур добавления и разбиения. На рисунке 2 представлен результат построения Seg-Tree.

4. СЕГМЕНТАЦИЯ

Рассмотрим развитие алгоритма разрастающихся регионов [6] адаптированного для использования Seg-Tree.. Процедура сегментации выполняется в два этапа. Сначала выполняется грубая сегментация, в которой алгоритм [6] применяется не к отдельным точкам, а к целым листьям Seg-Tree. Для каждого листа независимо оценивается направление нормали только по внутренним точкам листа. Результат сегментации на этом шаге приведен на рисунке 3б.

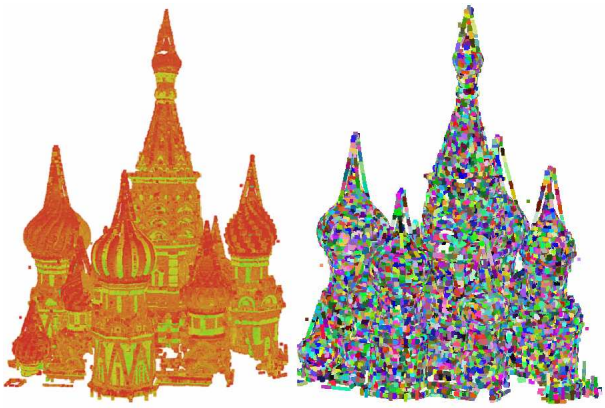


Рисунок 2: Структура Seg-Tree.

Элементы поверхности, расположенные на границах сегментов, могут содержать одновременно точки, принадлежащие разным поверхностям, например, на стыке двух плоскостей. Поэтому на втором этапе происходит уточнение границ сегментов уже на уровне отдельных точек. Для выделения границ выявляются все листы дерева, среди k-ближайших соседей которых есть листы, принадлежащие другому сегменту. Для всех точек, содержащихся в граничных листах, применяется традиционной алгоритм разрастающихся регионов, работающий уже на уровне отдельных точек. Небольшое различие в алгоритме состоит в том, что среди к-ближайших соседних точек граничных листов могут оказаться уже отсегментированные точки, лежащие в негранычных листах. В случае, когда в сегмент добавляется такая отсегментированная точка, мы выполняем слияние сегментов (Рис. 3с).. Такой подход позволяет получить корректную сегментацию на границах сегментов.

5. РЕЗУЛЬТАТЫ

Мы выполнили сравнение предлагаемого алгоритма с вариантом the [6]. Сначала мы выполнили построение Seg-Tree непосредственно из облака точек и получили пересегментированное иерархическое представление в виде элементов поверхности. Затем мы выполнили полностью автоматическую сегментацию с помощью алгоритма, описанного в предыдущем разделе, и провели сравнение результатов. Предлагаемый алгоритм обычно производит аналогичную сегментацию, однако, число небольших сегментов оказывается заметно меньшим. Мы также сравнивали результаты при различных значениях углового порога и получали более стабильные результаты в отличие от существующего алгоритма (Рис. 4.).

Другим способом снижения числа маленьких сегментов является увеличение числа ближайших точек при вычислении направлений нормали, однако, это одновременно приводит к сглаживанию поверхности точек и множество различных сегментов сливаются.

6. ВЫВОДЫ

В данной работе предложена новая схема организации трехмерных точечных данных в специальное иерархическое дерево, листья которого приближают элементарные участки поверхности. В случае некорректной поверхности, например, точки на реальных деревьях, листы дерева представляют собой локально связанные группы точек.

Затем мы применили Seg-Tree для выполнения сегментации с помощью алгоритма разрастающихся регионов. Предлагаемый метод сегментации объединяет в себе свойства части предложенных ранее методов:

- использование структуры организации данных для сегментации;
- применение методов кластеризации;
- использование метода разрастающихся регионов, позволяющего сегментировать разные типы поверхностей.

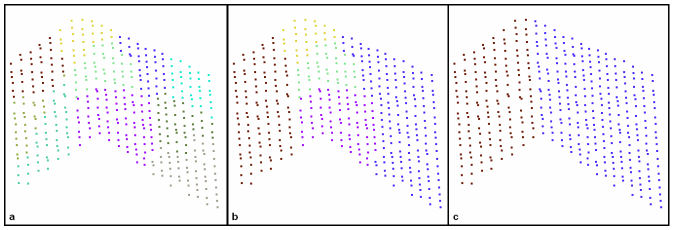


Рисунок 3: Сегментация с помощью листов Seg-Tree. a) Исходные листья Seg-Tree, b) Результат сегментации только по листьям дерева c) Результаты уточнения сегментации на уровне отдельных точек.

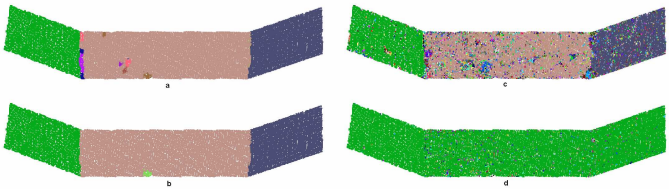


Рисунок 4: : Сравнение результатов сегментации при различных угловых порогах. a), b) – сегментация с помощью листьев Seg-tree, угловой порог 5˚ и 10˚ соответственно. c), d), -сегментация только по отдельным точкам при аналогичных значениях углового порога.

Предложенный подход менее чувствителен к шуму в исходных данных, чем [6], за счет анализа элементов поверхности, а не отдельных точек. Традиционный алгоритм производит большое число отдельных маленьких сегментов, а для сегментации на основе Seg-Tree этот эффект проявляется значительно реже. Наконец, предложенный алгоритм позволяет получить более стабильные результаты при различных значениях углового порога.

7. ЛИТЕРАТУРА

[1] Biosca, J., Lerma, J., 2008. Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods. ISPRS Journal of Photogrammetry and Remote Sensing, Volume 63(1), Pages 84-98

[2] Boubekeur, T., Heidrich, W., Granier, X., Schlick, C., 2006. *Volume-Surface Trees.* Computer Graphics Forum (Proceedings of EUROGRAPHICS 2006), 25(3), pp. 399-406

[3] Brakatsoulas S., Pfoser D., Theodoridis Y., 2002. *Revisiting R-tree construction principles.* Proceedings of the 6th East European Conference on Advances in Databases and Information Systems, pp. 149-162.

[4] Guttman A., 1984. *R-trees: a dynamic index structure for spatial searching*. Proceedings of the 1984 ACM SIGMOD international conference on Management of data, pp. 47-57.

[5] Jackins, C., Tanimoto S., 1983. *Quad-trees, oct-trees, and k-trees. A generalized approach to recursive decomposition of Euclidean space*, IEEE Transactions, Pattern Analysis and Machine Intelligence, vol. PAMI-5, pp. 533-539

[6] Rabbani T., van den Heuvel, F., Vosselmann, G., 2006, *Segmentation of point clouds using smoothness constraint.* International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 36(5), pp. 248-253

[7] Sankaranarayanan, J., Samet, H., Varshney, A., 2007. *A fast all nearest neighbor algorithm for applications involving large point-clouds*. Computers and Graphics, 31(2), pp. 157-174

8. БЛАГОДАРНОСТИ

Работа выполнена при поддержке фонда «Научный потенциал», грант №182.

Robust LIDAR data segmentation using compact point clusters

Abstract

Segmentation algorithms, which work with individual points, are sensitive to measurement noise and are computational-intensive. In recent years several segmentation algorithms based on hierarchical structure, like octree or kd-tree, were proposed. However, such plane-based partitioning procedures cannot produce correct results for various complex scenes with holes and occlusions. Continuous surfaces are often split and neighbor points are inserted into different tree nodes that degrade the segmentation result. In this paper we introduce a new hierarchical tree, named Seg-Tree, and show its effectiveness by applying to the popular segmentation algorithm region grow. Our version of this algorithm is robust to the noise and produce stable segmentation with lesser dependency of angular threshold. Seg-Tree is created using special over-segmentation procedure that produces compact leaves both for smooth and heavily noisy point sample surfaces. The over-segmentation procedure is performed using R-Tree construction with k-means clustering technique.

Задачи динамического поиска на поверхностях в трёхмерном пространстве

Павел Воронин
Факультет Вычислительной Математики и Кибернетики
Московский Государственный Университет им. Ломоносова
pavel.voronin@gmail.com

Аннотация

В статье рассматриваются задачи поиска подвижных объектов на поверхностях в трёхмерном пространстве. Предложен численный метод расчёта информационных множеств для таких задач. Метод основан на представлении множеств при помощи полей расстояний. Выполнение геодезических операций над множествами сводится к нахождению вязкостного решения уравнения Гамильтона-Якоби. При помощи разработанного аппарата решается задача нахождения достаточных поисковых ресурсов, задача построения поисковой траектории и задача нахождения достаточного числа ищущих.

Keywords: *динамический поиск, поля расстояний, геодезические расстояния.*

1. ВВЕДЕНИЕ

Данная работа посвящена геометрическому подходу к решению задач поиска активно уклоняющихся объектов. Базовой идеей методов этого типа является построение переменных во времени информационных множеств, описывающих процесс накопления ищущими и уклоняющимися объектами сведений о положении друг друга. Обширный обзор работ по данной теме дан в работе [1].

Как показала практика, аналитическое построение информационных множеств для нетривиальной структуры области поиска или большого количества ищущих объектов сопряжено со значительными трудностями. В связи с этим в диссертации [2] было предложено перейти к графическим методам решения поисковых задач: рассчитывать информационные множества на компьютере при помощи специальных численных методов с сопутствующей динамической визуализацией.

Для случая поиска на плоскости описано два подхода к построению численных методов расчёта информационных множеств: 1) в работе [2] был предложен метод, основанный на представлении поискового множества набором отрезков, аппроксимирующих его границу; 2) в работе [3] используется неявное задание множеств посредством полей расстояний [4].

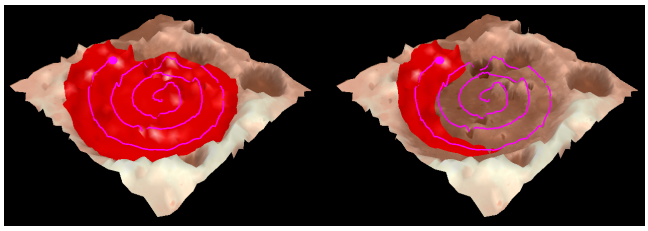


Figure 1: Поиск на ландшафте. Красным выделено множество, гарантированно свободное от искомых объектов (при двух разных соотношениях скоростей ищущего и искомых объектов).

В данной статье описывается подход, позволяющий расширить применимость метода, основанного на использовании полей расстояний, на случай поиска на поверхностях в трёхмерном пространстве (рис. 1). Для вычисления возникающих при этом геодезических расстояний предлагается воспользоваться аппаратом численного решения уравнений Гамильтона-Якоби [5].

На основе описанного подхода были разработаны подходы к решению трёх классов задач: 1) задачи нахождения поисковых траекторий (перебором из заданного класса); 2) задачи отыскания достаточных поисковых ресурсов (скоростей ищущих, дальности видимости); 3) задачи нахождения достаточного количества ищущих.

2. ОСНОВНЫЕ ПОНЯТИЯ. ФОРМУЛИРОВКА ЗАДАЧИ

Рассмотрим задачу поиска объектов на заданном множестве. Воспользовавшись классификацией из статьи [1], введём основные понятия.

Областью поиска называется некоторое множество, на котором допускается нахождение участвующих в поиске объектов.

Объекты поиска (искомые объекты) могут быть статичными или подвижными. Для простоты их можно считать точечными.

Ищущие объекты — это подвижные точки, в задачу которых входит отыскание такого обхода области поиска, при котором будут обнаружены все объекты поиска или же будет доказано, что ни одного искомого объекта в заданной области не было.

В качестве *условия обнаружения* будем рассматривать приближение одного из ищущих к искомому на заданное расстояние.

В зависимости от задачи, поведение искомых объектов может быть разным. Мы будем рассматривать случай, когда искомые объекты активно уклоняются от ищущих, местоположение, скорость и стратегия движения которых в каждый момент времени им известны, — в то время как ищущие знают только максимальную скорость уклоняющихся.Эту же ситуацию с точки зрения ищущих можно интерпретировать иначе: ничего, кроме максимальной скорости противников, ни одной из сторон неизвестно — и, в дополнение к этому, ищущим неизвестно количество искомых объектов. Выбираемая стратегия должна гарантировать нахождение сколь угодно большого конечного числа объектов.

Геометрический подход к решению поисковых задач основан на использовании вспомогательных переменные во

времени *информационных множеств*, соответствующих сведениям, накапливаемым в процессе поиска ищущими и уклоняющимися объектами друг о друге. Наиболее значимым типом информационных множеств является *остаточная область* — множество точек области поиска, в которых уклоняющихся объектов не может быть (иначе они были бы обнаружены ищущими ранее). Целью ищущих является увеличение остаточной области вплоть до момента, когда она охватит всю область поиска.

Информационные множества показали себя как крайне эффективный инструмент при решении многих задач динамического поиска. Вместе с тем, попытки применить их для случая нескольких ищущих объектов или невыпуклых поисковых областей сталкиваются с серьёзными трудностями. Дело в том, что структура информационных множеств в этих случаях значительно усложняется [1] и аналитические методы их построения становятся неэффективными. Поэтому для дальнейшего развития геометрического подхода к исследованию поисковых задач потребовалась разработка численных методов нахождения информационных множеств.

3. ОБЩАЯ СХЕМА ЧИСЛЕННЫХ МЕТОДОВ

Рассмотрим общую схему построения численных методов для расчёта остаточной области [2]. Для простоты ограничимся случаем области поиска без границы, одного ищущего и одного уклоняющегося объекта, движущихся с постоянными скалярными скоростями.

Введём следующие обозначения. Пусть скалярная скорость уклоняющегося объекта равна U, а скорость ищущего объекта — V (будем считать, что $V > U$, то есть ищущий имеет преимущество по скорости). Пусть область обнаружения ищущего — круг радиуса R, а траектория его перемещения $P(t)$. Нам нужно рассчитать изменение переменной остаточной области $A(t)$ на отрезке времени от $t = 0$ до $t = T$.

Дискретизируем задачу. Приблизим заданную траекторию ломаной, проходящей через точки P_0, P_1, … , P_n — положения ищущего в моменты времени $t_0 = 0$, t_1, … , $t_n = T$ ($t_i = i\Delta t$, $\Delta t = T/n$). Будем последовательно искать остаточные области A_0, A_1, … , $A_n = A$, где $A_i = A(t_i)$. Как показано в работе [3], общая формула описания фигур A_k выглядит так:

$$\begin{cases} A_0 = B(P_0; R), \\ A^{old}_{k+1} = Offset(A_k, -\Delta U), \\ A^{new}_{k+1} = A(P_k, P_{k+1}, r, R), \\ A_{k+1} = A^{old}_{k+1} \cup A^{new}_{k+1}. \end{cases} \quad (1)$$

Здесь $B(P_0; R)$ — это круг радиуса R с центром в точке P_0; $Offset(A, -r)$ — сужение множества A на полосу шириной r (r-расширение с отрицательным r); $A(P_k, P_{k+1}, r, R)$ — т.н. фигура A-типа: $A(P_k, P_{k+1}, r, R) = \bigcup_{t\in[t_k,t_{k+1}]} Offset(B(P(t), R), -U(t - t_k))$. На практике вместо точного значения $A(P_k, P_{k+1}, r, R)$ может использоваться некоторое его приближение, например $A(P_k, P_{k+1}, r, R) \approx B(P_{k+1}, R)$ или $A(P_k, P_{k+1}, r, R) \approx \bigcup_{i=0}^{m} B^U(P(t^i)); t \in [t_k, t_{k+1}], \forall i$, где $B^U = Offset(B(P(t^i), R), -U(t^i - t_k))$.

Обобщения вышеприведённых формул на случай нескольких ищущих и областей поиска с границей описаны в статье [3] и опираются на те же самые базовые операции.

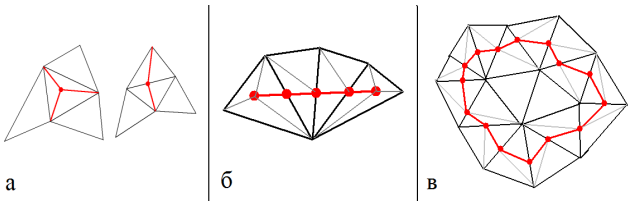


Figure 2: Подразбиение сетки: а) по точке; б) по отрезку; в) по линии уровня.

4. МЕТОДЫ, ОСНОВАННЫЕ НА ИСПОЛЬЗОВАНИИ ПОЛЕЙ РАССТОЯНИЙ

Итак, для широкого класса поисковых задач возможно приближённое рекуррентное вычисление информационных множеств. Заметим, что для реализации численного метода на их основе нам необходимо в каждом случае выбирать такое программное представление множеств, которое позволяло бы эффективно проводить над ними следующие четыре базовые операции: объединение, пересечение, разность и r-расширение.

В работе [3] в качестве такого представления предложено неявное задание множеств, известное как поле расстояний (Distance Field, *DF* [4]): вместо заданного замкнутого геометрического объекта будем рассматривать порождаемое им скалярное поле, значение которого в произвольной точке равно расстоянию от этой точки до границы объекта, взятому со знаком плюс, если точка лежит вне объекта, и со знаком минус в противном случае.

Покажем, как при помощи полей расстояний проводятся вычисления. Возьмём последнее уравнение системы 1 и перепишем его, выразив все действия через операции над полями расстояний. Получим следующую последовательность действий (*DF*-алгоритм) [3]:

$$\begin{array}{l} 1) DF_{old} = DF_{A_k} + \Delta U, \\ 2) DF_{upd} = DF_{A(P_k,P_{k+1},r,R)}, \\ 3) DF_{new} = \min(DF_{old}, DF_{upd}), \\ 4) DF_{A_{k+1}} = DF_{ISO_0 DF_{new}}. \end{array}$$

5. ИНФОРМАЦИОННЫЕ МНОЖЕСТВА НА ПОВЕРХНОСТЯХ

Поисковым задачам на поверхностях в трёхмерном пространстве посвящено значительное количество работ, однако большинство из них посвящено достаточно простым аналитически заданным поверхностям. Например, в работе [6] описаны поисковые траектории шаре, в работе [7] — на цилиндре и торе, а в диссертации [8] — на поверхностях вращения. Целью настоящей работы является разработка метода, который бы позволил решать поисковые задачи на поверхностях достаточно общего вида.

В качестве области обнаружения в задачах на поверхностях обычно рассматривают геодезический круг с центром в точке расположения ищущего. Под r-расширением множества на поверхностях понимается его геодезическое r-расширение, то есть сумма (или разность) Минковского этого множества и всех геодезических кругов радиуса r с центрами на границе множества.

Таким образом, для вычисления информационных множеств по приведённому выше алгоритму нам понадобится аппарат быстрого вычисления геодезических расстоя-

ний и геодезических r-расширений.

5.1 Геодезические расстояния и методы их вычисления

Пусть на некоторой поверхности в трёхмерном пространстве задана область Ω и введена метрика $g : x \in \Omega \mapsto H(x) \in R^{2\times 2}$, $H(x) > 0$. Пусть в области Ω задана некоторая регулярная кривая $\gamma(t)$, $t \in [0, 1]$. Длина этой кривой вычисляется по следующей формуле [9]: $L(\gamma) = L_g(\gamma) = \int_0^1 \sqrt{(\gamma'(t))^T H(\gamma(t))\gamma'(t)}dt$.

В области Ω расстоянием $D(A, B)$ между точками A и B называется наименьшая из длин соединяющих их кривых, целиком лежащих в данной области.

Функционалом длины для множества точек $S = \{x_k\}_{k=1}^N \subset \Omega$ называется функция $U_S(x) = \min_k D(x, x_k)$, $x \in \Omega$. Имеет место следующая теорема [10, 11].

Теорема 1. Для любого множества $S = \{x_k\}_{k=1}^N \subset \Omega$ функционал длины $U_S(x)$ есть единственное вязкостное решение уравнения Гамильтона-Якоби

$$\|\nabla U_S\|_{H^{-1}} = 1 \text{ при условии } U_S(x_k) = 0, \forall k, \qquad (2)$$

где $\|v\|_A = \sqrt{v^T Av}$.

Для вычисления функционала длины разработан целый ряд эффективных численных методов. Большинство из них работают с сеточным приближением задачи: по заданной сетке (и значениям метрики в её узлах) находят для любого подмножества узлов S приближённое значение функционала длины U_S во всех узлах этой сетки. Наиболее распространённым подходом является численное решение уравнения (2) посредством одной из модификаций метода быстрого движения (Fast Marching, *FM* [5]). При использовании подходящих эффективных структур данных его сложность составляет $O(n \log(n))$, где n — число вершин сетки. На практике алгоритм ведёт себя почти линейно. В статье [12] приводится модификация *FM*-алгоритма, в которой при помощи квантования приоритета удаётся добиться снижения теоретической сложности до оптимальной $O(n)$.

При реализации описанных в данной работе алгоритмов мы опирались на вариант *FM*-алгоритма, входящий в состав библиотеки *Geowave*, являющейся частью *Toolbox Fast Marching*[1]. При помощи этого же инструментария мы генерировали равномерные сетки для всех исследуемых поверхностей, следуя алгоритму ретриангуляции, изложенному в статье [13].

5.2 Геодезический поиск

Опишем реализацию *DF*-алгоритма для построения остаточной области на поверхностях в трёхмерном порстранстве.

1. Первый и третий шаги *DF*-алгоритма выполняются за линейное от количества узлов сетки время.

2. В соответствии с первой строкой системы 1, для нахождения фигуры A_0 необходимо вычислить поле расстояний для геодезической окружности с заданным центром и радиусом: $DF_{B(C;R)}(x) = DF_C(x) - R$. Оно выражается через поле расстояний для центра окружности, а поле расстояний для точки есть её функционал длины. Значит, его можно найти при помощи *FM*-алгоритма. Заметим, однако, что в классическом варианте при помощи алгоритма быстрого движения можно найти функционалы длины только для точек, являющихся узлами сетки. Если же точка C попадает внутрь треугольника сетки или на его ребро, понадобится предварительно сделать подразбиение сетки (рис. 2а).

3. На втором шаге *DF*-алгоритма необходимо вычислить поле расстояний для фигуры $A(P_k, P_{k+1}, r, R)$. Сначала рассмотрим случай $r = R$. Тогда $A(P_k, P_{k+1}, R, R) = \bigcup_{t\in[t_k,t_{k+1}]} B(P(t), R)$, то есть $A(P_k, P_{k+1}, R, R)$ — геодезическое R-расширение криволинейного отрезка $[P_k, P_{k+1}]$. Следовательно, $A(P_k, P_{k+1}, R, R) = ISO_R DF_{[P_k,P_{k+1}]}$. Подразбив по $[P_k, P_{k+1}]$ сетку (рис. 2б), получим набор новых точек $S^* = \{s_i^*\}_{i=1}^m$. Имеет место следующее приближение: $A(P_k, P_{k+1}, R, R) \approx ISO_0(\min_i DF_{B(s_i,R)})$. Возвращаясь к общему случаю, по аналогии имеем

$$A(P_k, P_{k+1}, r, R) \approx ISO_0(\min_i DF_{B(s_i,r_i)}),$$

$$r_i = r + (R - r)\frac{L([s_i, P_k])}{L([P_{k+1}, P_k])}.$$

   Таким образом, мы выразили границу фигуры $A(P_k, P_{k+1}, r, R)$ через поверхность уровня некоторого поля.

4. На четвёртом шаге *DF*-алгоритма и в конце п. 3 требуется вычислять поле расстояний для области K, граница которой задана нулевой изолинией некоторого поля, $K = ISO_0 D$, а значит, состоит из отрезков, вершины которых лежат на рёбрах сетки. Как в п.-п. 2-3, подразобьём сетку по этим отрезкам (рис. 2в) и получим набор новых узлов S^*. Легко видеть, что $DF_K = sgn(D) \cdot U_{S^*}$.

5. Заметим также, что *FM*-алгоритм позволяет вычислять функционал длины не на всей сетке, а лишь на тех её узлах, где его значение не привышает некоторой заданной величины. Это позволяет значительно ускорить работу алгоритма: будем вычислять значение поля расстояний лишь в узлах сетки, удалённых от нулевой изолинии менее, чем на $V \cdot \Delta t + e_{max}$ (e_{max} — максимальная длина ребра сетки).

Описанный алгоритм был реализован на языке C++ и позволяет вычислять и визуализировать (с использованием библиотеки OpenGL) остаточные области для траекторий на поверхностях в реальном времени. Например, анимация, кадр из которой приведён на рис. 1, рассчитывалась и отрисовывалась со скоростью 40-50 кадров в секунду на обычном домашнем компьютере (Pentium 4 3 GHz, 1.5 Gb DDR 400 MHz, NVidia GeForce 6800) при размерах сетки в несколько тысяч треугольников.

6. РЕШЕНИЕ ПОИСКОВЫХ ЗАДАЧ

Разработанный аппарат позволяет в интерактивном режиме решать разнообразные поисковые задачи, сводящиеся к вычислению информационных множеств для некоторого набора траекторий и/или ресурсов.

[1] http://www.ceremade.dauphine.fr/~peyre/matlab/

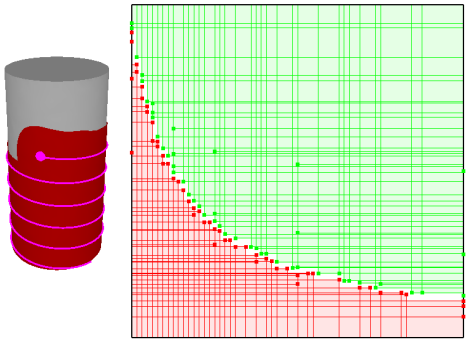


Figure 3: Поиск на цилиндре. На графике зелёным отмечены области фазовой плоскости, которым соответствуют ресурсы, гарантированно достаточные для успешного поиска, красным — гарантированно недостаточные.

Простейшей из таких задач является задача определения, гарантирует ли заданная траектория движения ищущего объекта обнаружение активно уклоняющихся объектов при известных скоростях всех участников и известном радиусе обнаружения. Чтобы ответить на этот вопрос, достаточно построить остаточную область и проверить, совпадает ли она по окончанию движения со всей областью поиска. Так же решается и задача для нескольких ищущих.

Несколько более трудными являются задача нахождения достаточных ресурсов, задача построения поисковой траектории и задача нахождения достаточного числа ищущих объектов: для их решения необходим перебор наборов ресурсов, траекторий и ищущих соответственно. Опишем методы решения каждой из этих задач, рассматривая для простоты изложения поверхности тел простой структуры (те же самые подходы применимы и для поверхностей общего вида, однако соответствующие траектории могут быть устроены гораздо сложнее).

6.1 Нахождение достаточных ресурсов

Рассмотрим задачу гарантированного поиска активно уклоняющихся объектов в некоторой заданной ограниченной области. Пусть траектории ищущих фиксированы. Ищущие обладают определёнными поисковыми ресурсами. Для любого набора ресурсов, пользуясь описанным выше алгоритмом, мы можем построить остаточную область и определить, совпадает ли она с областью поиска или нет, а значит, являются ли ли данные ресурсы достаточными для нахождения уклоняющихся объектов при движении по заданной ломаной. Решая эту задачу для целого ряда наборов, мы можем охарактеризовать ресурсы, достаточные для гарантированного нахождения.

В качестве примера рассмотрим фиксированную винтовую линию на цилиндре. В качестве ресурсов будем рассматривать радиус круга обнаружения R и отношение скорости ищущего объекта к скорости уклоняющихся U. Пусть ресурсы могут изменяться в пределах $R \in [R_{min}, R_{max}]$, $U \in [U_{min}, U_{max}]$.

Заметим, что наша задача обладает свойством монотонности: если какая-то кривая является решением при ресурсах (R^*, U^*), то она является решением и при любых ресурсах (R^{**}, U^{**}) таких, что $R^{**} \geq R^*, U^{**} \geq U^*$ (это следует из вложенности соответствующих остаточных областей). В терминах фазовой плоскости $R - U$ это означает, что достаточность ресурсов (R^*, U^*) влечёт за собой достаточность из квадрата $[R^*, R_{max}] \times [U^*, U_{max}]$. Верно и обратное: из недостаточности ресурсов (R^*, U^*) следует недостаточность ресурсов из квадрата $[R_{min}, R^*] \times [U_{min}, U^*]$. Эти свойства можно выразить и по-другому: достаточные ресурсы на фазовой плоскости лежат в области, ограниченной снизу графиком неубывающей функции, $\{(R, U) \mid U \geq F(R)\}$. Приближённое отыскание этой кривой, в свою очередь, можно свести к набору одномерных задач.

Будем искать приближённые значения $U_i \approx F(R_i)$, $1 \leq i \leq N$, $|R_i - R_{i+1}| < \varepsilon_R$, $|U_i - F(R_i)| < \varepsilon_U$. Припишем каждому значению U_i интервал возможных значений $[U_i^1, U_i^2]$. В начальный момент все интервалы равны, $U_i^1 = U_{min}$, $U_i^2 = U_{max}$. Когда для какого-нибудь значения U_i интервал уточняется, то по свойству монотонности уточняются и интервалы всех остальных значений: $[U_i^1, U_i^2] \to [V^1, V^2] \Rightarrow$ 1) $[U_j^1, U_j^2] \to [\max(U_j^1, V^1), U_j^2]$, $\forall j < i$; 2) $[U_j^1, U_j^2] \to [U_j^1, \min(U_j^2, V^2)]$, $\forall j > i$.

Для нахождения U_i, отстоящего от $F(R_i)$ не более чем на ε_U, нам достаточно заключить его в интервал длины не более $2\varepsilon_U$, $U_i^2 - U_i^1 \leq 2\varepsilon_U$. Будем делать это методом дихотомии. На каждой итерации будем брать ресурсы(R_i, U^*), $U^* = \frac{U_i^1 + U_i^2}{2}$. Если таких ресурсов достаточно для решения задачи, то $[U_i^1, U_i^2] \to [U_i^1, U^*]$, иначе $[U_i^1, U_i^2] \to [U^*, U_i^2]$. Если получившийся интервал имеет длину менее $2\varepsilon_U$, то заканчиваем вычисления. Используя полученный интервал уточняем интервалы для значений $U_j, j \neq i$, а за значение U_i берём середину полученного интервала.

Поскольку уточнение интервала значений одного из U_i влияет на все остальные, то имеет значение порядок их вычиления. Разумным видится следующий порядок: U_1, U_N, $U_{\frac{N}{2}}$, $U_{\frac{N}{4}}$, $U_{\frac{3N}{4}}$, $U_{\frac{N}{8}}$, $U_{\frac{3N}{8}}$, $U_{\frac{5N}{8}}$, $U_{\frac{7N}{8}}$ и т. д. Практика показывает, что в большинстве случаев уточнение интервалов значений при описанном обходе существенно уменьшает количество итераций. Например, для $U, R \in [0, 1]$, $\varepsilon_U = \varepsilon_R = \frac{1}{100}$, $F(R) = 1 - R$ алгоритму требуется 322 шага, тогда как при независимом вычислении U_i потребовалось бы порядка 700 шагов.

Работа алгоритма проиллюстрирована на рис. 3.

6.2 Построение поисковой траектории

Ещё одной задачей, легко сводящейся к нахождению остаточных областей, является выбор из некоторого заданного класса кратчайшей траектории, обеспечивающей нахождение активно уклоняющихся объектов. В качестве примера рассмотрим задачу построения поисковой траектории для поверхности тела, обладающего осевой звёдностью (при заданных скоростях объектов и радиусе обнаружения). Без ограничения общности будем считать, что поверхность задана уравнениями:

$$\{x = R(h, \phi)\cos(\phi); y = h; z = R(h, \phi)\sin(\phi)\},$$

где $h \in [0, 1]$, $\phi \in [0, 2\pi]$, $R : [0, 1] \times [0, 2\pi] \to [R_{min}, R_{max}]$.

Будем искать траекторию в классе обобщённых винтовых линий на этой поверхности:

$$\{x = R(h, \Phi(h))\cos(\Phi(h)); y = h; z = R(h, \Phi(h))\sin(\Phi(h))\},$$

где $h \in [0, 1]$, а $\Phi(h) : [0, 1] \to [0, +\infty]$ — неубывающая

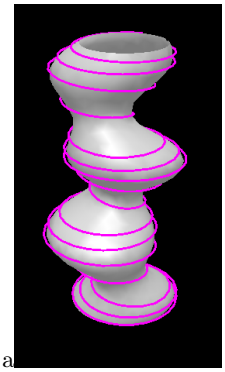
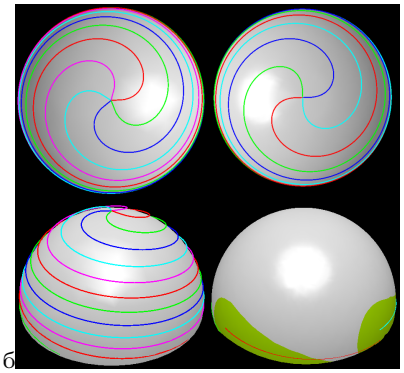
а  б 

Figure 4: а) Поисковая траектория на звёздчатом многограннике; б) Поиск несколькими ищущими.

функция.

Разобьём отрезок $[0,1]$ на N_h равных частей: $h_0 = 0, h_1 = \Delta h = \frac{1}{N_h}, h_2 = 2\Delta h, \ldots, h_{N_h} = 1$. Последовательно на каждом отрезке $[h_i, h_{i+1}]$ будем искать $\Phi(h)$ в виде линейной функции: $\Phi(h) = \Phi(h_i) + \alpha_i (h - h_i)$, где $\alpha_i \in [\alpha_{min}, \alpha_{max}]$. Параметр α_i будем искать методом дихотомии. Приближение $\tilde{\alpha_i} \approx \alpha_i$ будем считать оценкой сверху, если для кривой, заданной $\alpha_0, \alpha_1, \ldots, \alpha_{i-1}, \tilde{\alpha_i}$при $h \in [0, h_{i+1}]$, остаточная область покрывает часть поверхности, заключённую между плоскостями $y = 0$ и $y = h_{i+1}$, — и оценкой снизу в противном случае.

Такой жадный алгоритм с одной стороны гарантирует быстрое нахождение поисковой траектории, а с другой — обеспечивает адаптацию кривой к конкретному виду поверхности: витки винтовой линии будут плотнее упакованы на участках, лежащих близко к оси, и реже — на лежащих далеко от неё.

Результат работы алгоритма для некоторого тела, обладающего осевой звёздностью, приведён на рис. 4а).

6.3 Нахождение достаточного числа ищущих

Если для некоторой области и одного ищущего с заданными ресурсами не удаётся построить поисковую траекторию, которая бы обеспечивала гарантированное нахождение искомых объектов, то встаёт вопрос о числе ищущих, при котором это возможно (и соответствующих кратчайших траекториях). Если для любого числа ищущих мы можем указать достаточно представительный класс поисковых траекторий и порядок их перебора, то такую задачу можно решать тем же методом, что и описанные выше.

В качестве примера рассмотрим полусферу достаточно большого радиуса. Для любого числа ищущих n будем рассматривать поисковые траектории, представляющие из себя спирали, начинающиеся в вершинах вписанного в основание полусферы правильного n-угольника, сходящиеся на полюсе и имеющие одинаковые углы наклона $\alpha \in [\varepsilon_\alpha, \frac{\pi}{2}]$ (см. рис. 4б). Для нахождения минимального числа ищущих достаточно рассматривать траектории с углом наклона ε_α. Кратчайшие траектории из соответствующего класса можно найти, перебирая углы наклона методом дихотомии (см. п.-п. 6.1, 6.2).

Работа алгоритма проиллюстрирована на рис. 4б).

7. REFERENCES

[1] Чхартишвили А. Г. and Шикин Е. В., "Динамический поиск объектов. Геометрический взгляд на проблему," *Фундаментальная и прикладная математика*, vol. 1, no. 4, pp. 827–862, 1995.

[2] Березин С. Б., *Графический подход к исследованию и решению задач динамического поиска*, Ph.D. thesis, МГУ, 2000.

[3] Воронин П., "О новом подходе к расчёту и визуализации информационных множеств в задачах динамического поиска," in *Труды 18-й международной конференции ГрафиКон*. Московский Государственный Университет, 2008.

[4] B. A. Payne and A. W. Toga, "Distance field manipulation of surface models," *Computer Graphics and Applications*, vol. 12, no. 1, pp. 65–71, 1992.

[5] J. A. Sethian, *Level Sets Methods and Fast Marching Methods. 2nd edition*, Cambridge University Press, 1999.

[6] Залгаллер В. А., "Кратчайшие линии осмотра сферы," *деп. в ВИНИТИ 24.09.92, № 2844-В92.*

[7] Губайдуллин С. М. and Шикин Е. В., "Следящие области на цилиндре и на торе," *Вестник московского университета. Сер. 15. Вычислительная математика и кибернетика*, vol. 0, no. 2, pp. 46–50, 1992.

[8] Чхартишвили А. Г., *Метод следящих областей в задачах динамического поиска*, Ph.D. thesis, МГУ, 1994.

[9] Картан Э. Ж., *Риманова геометрия в ортогональном репере*, МГУ, 1960.

[10] Курант Р., *Уравнения с частными производными*, Мир, 1964.

[11] M. G. Crandall, H. Ishii, and P. L. Lions, "User's guide to viscosity solutions of second order partial differential equations," *Bulletin of the American Mathematical Society*, vol. 27, no. 1, pp. 1–67, 1992.

[12] L. Yatziv, A. Bartesaghi, and G. Sapiro, "O(n) implementation of the fast marching algorithm," *Journal of Computational Physics*, vol. 212, no. 2, pp. 393–399, 2006.

[13] G. Peyré and L. D. Cohen, "Geodesic remeshing using front propagation," *International Journal on Computer Vision*, vol. 69, no. 1, pp. 145–156, 2006.

ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ

Расширенная версия данной статьи и дополнительные цветные анимированные иллюстрации, демонстрирующие работу описанных алгоритмов, доступны на сайте http://www.primaler.ru/cg/. Там же доступна для скачивания реализация описанных алгоритмов на C++ (платформо-независимая) и MATLAB. Отдельный раздел сайта посвящён ссылкам на литературу по теме.

О вычислении полей расстояний для многоугольников на плоскости

Павел Воронин, Андрей Адинец
Факультет Вычислительной Математики и Кибернетики
Московский Государственный Университет им. Ломоносова
{pavel.voronin, adinetz}@gmail.com

Аннотация

В статье рассматривается задача вычисления значений поля расстояний для многоугольника в узлах неравномерной сетки на плоскости. Описывается несколько подходов: решение "в лоб", с последующей оптимизацией под векторные и многопоточные вычисления, доступные на большинстве совеременных процессоров; использование иерархических пространственных структур данных, позволяющих оценивать расстояния между группами примитивов; наконец, перенос вычислений на графический процессор (при помощи системы C$). Приводится детальный сравнительный анализ алгоритмов (скорости их работы, ограничений применимости, сильных и слабых сторон). Использование описанных методов демонстрируется на примере задачи построения сетки, приближающей заданную фигуру, и задачи отыскания области, свободной от подвижных нарушителей, при известной траектории движения ищущего.

Keywords: *поля расстояний.*

1. ВВЕДЕНИЕ

Полем расстояний для некоторого объекта называется функция, значение которой в произвольной точке равно расстоянию от этой точки до границы объекта. Дополнительно можно ввести знак поля: точкам внутри объекта соответствуют отрицательные значения, а точкам вне него — положительные.

Поля расстояний получили широкое распространение в самых различных областях машинного зрения, компьютерной графики и математического моделирования. Двумерные дискретные поля расстояний, среди прочего, применяются для сегментации, выделения скелета и сравнения формы фигур [1]. Трёхмерные дискретные поля расстояний используются для обработки и анализа медицинских трёхмерных изображений [2]. Непрерывный вариант трёхмерных полей расстояний применяется в физическом моделировании, определении столкновений, анимации, трассировке лучей и системах автоматизированного проектирования [3]. Непрерывные двумерные поля расстояний нашли своё применение в таких разных областях, как навигация роботов [4], хранение векторных шрифтов [5], построение равномерных сеток с заданной границей [6] и расчёт информационных множеств в задачах динамического поиска [7].

Непрерывные поля расстояний обычно задаются в виде замеров на узлах некоторой сетки, с возможностью интерполяции значения в любой точке. Для вычисления приближённых значений поля разработаны быстрые методы как для равномерных сеток [8, 9], так и для сеток общего вида [10]. С другой стороны, описанные в литературе алгоритмы точного вычисления полей расстояний [11, 12, 13, 14] работают исключительно с равномерными сетками.

Данная статья посвящена вычислению значений поля расстояний в узлах сетки общего вида. В работе отдельно рассматриваются случаи полей расстояний со знаком и без; расчёта поля во всех точках и только в принадлежащих некоторой полосе вокруг границы объекта; разового вычисления и многократного вычисления для одного и того же набора точек, на разных объектов.

Нами было реализовано три подхода: решение "в лоб", полным перебором, но с низкоуровневой оптимизацией кода (SSE, многопоточность); использование иерархических пространственных структур данных; перенос вычислений на графический процессор (при помощи системы C$ [15]). Был проведён подробный сравнительный анализ реализованных алгоритмов для различных сценариев работы. Приводится несколько примеров использования реализованных подходов для решения практических задач.

2. РЕШЕНИЕ ПОЛНЫМ ПЕРЕБОРОМ

Введём обозначения. Пусть даны набор точек $X = \{x_i\}_{i=1}^N$ и многоугольник A со сторонами $\{e_j\}_{j=1}^M$. Требуется найти значение поля расстояний многоугольника DF_A во всех точках из набора X: $DF_A(x_i)$, $1 \le i \le N$. Обозначим расстояние от точки x до отрезка e как $\rho(x, e)$. Тогда, согласно определению, значение поля расстояний DF_Aв точке x равно

$$DF_A(x) = \sigma_A(x) \cdot \rho(x, A), \text{ где}$$

$$\rho(x, A) = \min_{1 \le j \le M} (\rho(x, e_j)),\ \sigma_A(x) = \begin{cases} -1, & x \text{ внутри } A, \\ +1, & \text{иначе.} \end{cases}$$

Рассмотрим отдельно алгоритмы вычисления расстояния $\rho(x, A)$ и знака $\sigma_A(x)$.

2.1 Вычисление расстояний до границы многоугольника

В соответствии с приведённой выше формулой, имеем следующий алгоритм для вычисления расстояний:

Algorithm 1 Расстояния, полный перебор.

```
Для каждого i от 1 до N
    ρ(x_i, A) = +∞
    Для каждого j от 1 до M
        d_ij = ρ(x_i, e_j)
        Если d_ij < ρ(x_i, A), то
            ρ(x_i, A) = d_ij
```

Для повышения скорости работы этого элементарного алгоритма мы использовали следующие приёмы.

Вместо $\rho(x_i, e_j)$ вычисляется его квадрат. Корень берётся один раз, из найденного минимума.

Все необходимые для вычисления $\rho(x_i, e_j)$ данные (координаты точек; координаты концов отрезка, координаты его направляющего вектора, длина) предпросчитываются и хранятся не в виде массива структур, а виде структуры массивов.

Используются параллельные вычисления, доступные на большинстве современных систем. Вершины сетки делятся на столько равных групп, сколько нам доступно потоков — и каждая группа обрабатывается в отдельном потоке (при помощи кросс-платформенной библиотеки Intel TBB[1] [16]).

Используются векторные вычисления, также доступные во всех современных процессорах (например, технология SSE (Streaming SIMD Extensions) на процессорах Intel и AMD).

2.2 Определение принадлежности многоугольнику

Для того, чтобы вычислить значение функции $\sigma_A(x)$, необходимо определить, принадлежит ли точка x многоугольнику A. Как показано в работе [17], в случае многоугольников с большим числом сторон наиболее эффективным является алгоритм, основанный на подсчёте пересечений сторон многоугольника с горизонтальным лучом.

Итак, пусть заданы точка $p = (p_x, p_y)$ и многоугольник A со сторонами $E = e_{i\,i=1}^{K}$, $e_i = (p^{e_i}, q^{e_i})$. Выпишем алгоритм определения принадлежности точки многоугольнику (в качестве испускаемого луча R из соображений эффективности возьмём луч $y = p_y, x \geq p_x$).

Algorithm 2 Знак, полный перебор.

Для каждого i от 1 до N
  Если ЧислоПересечений(x_i, R, E) — нечётное, то
    $\sigma_A(x) = -1$, иначе
    $\sigma_A(x) = +1$.

Функция ЧислоПересечений(x, R, E)
  $n = 0$.
  Для каждого e из E
    Если ЛучПересекает(x, R, e), то
      $n = n + 1$.
  Верни n.

Функция ЛучПересекает(x, R, E)
  Если $(p_y^e - p_y)(q_y^e - p_y) > 0$, то Верни Нет.
  Если $(p_x^e < p_x)$ и $(q_x^e < p_x)$, то Верни Нет.
  Если R не пересекает e, то Верни Нет.
  Верни Да.

Для некоторых классов многоугольников существуют более простые и эффективные алгоритмы определения принадлежности им заданной точки. В частности, для выпуклых, монотонных и звёздных многоугольников легко строятся алгоритмы, основанные на двоичном поиске [17, 18].

3. ИЕРАРХИЧЕСКИЙ ПОДХОД

При расчётах по алгоритму 1 вычисляются расстояния между от всех точек до всех отрезков. Такой подход не учитывает пространственной когерентности входных данных: расстояния от точки до двух отрезков, находящихся на малом расстоянии друг от друга, отличаются слабо; аналогично, слабо отличаются расстояния до отрезка от двух точек, расположенных близко друг к другу. Поэтому для сокращения объёма вычислений мы применили идеи, использованных в статье [19] для ускорения нахождения кратчайшего расстояния между двумя объектами. Метод основан на использовании иерархий ограничивающих тел для оценки расстояний между группами точек и отрезков с последующим исключением из рассмотрения тех пар групп, которые заведомо не могут быть ближайшими друг к другу. В качестве иерархий ограничивающих тел как для точек, так и отрезков мы использовали kd-деревья [20] (в листьях дерева содержится не менее некоторого фиксированного числа точек $(2^5 - 2^8)$ и по одному отрезку соответственно).

Рассмотрим алгоритм нахождения расстояний $\rho(x_i, E)$ от точек $X = x_{i\,i=1}^{N}$ до множества отрезков $E = e_{j\,j=1}^{K}$. Будем обозначать kd-дерево, соответствующее множеству отрезков L как $T(L)$; множество отрезков, соответствующее kd-дереву K — как $S(K)$; поддеревья kd-дерева K — как K_1 и K_2; kd-дерево, соответствующее множеству точек V — как $T(V)$; множество точек, соответствующее kd-дереву K — как $P(K)$.

Algorithm 3 Расстояния от множества точек до множества отрезков, иерархическое отсечение.

$\rho(X, E) = +\infty$
Очередь $Q = \{(T(X), T(E))\}$
Пока Q непуста
  $(t^V, t^S) = Q.pop()$
  Если t^V и t^S листья, то
    Для всех v из $V(t^V)$
      Если $\rho(v, E) < \rho(v, S)$, то $\rho(v, E) = \rho(v, S)$.
    $\rho(V(t^V), E) = \max_{v \in V(t^V)} \rho(v, E)$
  иначе
    $\widetilde{d}_{min} = \widetilde{\rho}_{min}(t^V, t^S)$
    Если $\rho(t^V, E) > \widetilde{d}_{min}$
      $\widetilde{d}_{max} = \widetilde{\rho}_{max}(t^V, t^S)$
      Если $\widetilde{d}_{max} < \rho(t^V, E)$, то $\rho(t^V, E) = \widetilde{d}_{max}$
      Если t^V лист, то
        $Q.push(t^V, t_1^S)$, $Q.push(t^V, t_2^S)$
      иначе
        $\rho(t_1^V, E) = \rho(t_2^V, E) = \rho(t^V, E)$
        Если t_S лист, то
          $Q.push(t_1^V, t^S)$, $Q.push(t_2^V, t^S)$
        иначе
          $Q.push((t_1^V, t_1^S) Q.push(t_2^V, t_1^S)$
          $Q.push(t_1^V, t_2^S)$, $Q.push(t_2^V, t_2^S)$

Алгоритм совершает обход деревьев в ширину и для каждой пары нелистовых поддеревьев вычисляет верхнюю $\widetilde{\rho}_{max}$ и нижнюю $\widetilde{\rho}_{min}$ оценки расстояний между ними. Если текущее значение ρ меньше $\widetilde{\rho}_{min}$, то данные поддеревья не является друг для друга ближайшими, и пара исключается из дальнейшего рассмотрения. В противном случае процедура позднее повторяется для их поддеревьев.

В качестве нижней оценки берётся расстояние между описанными прямоугольниками поддеревьев. В качестве верхней — минимум максимальных расстояний от вершин отрезков, лежащих на границе описанного прямоугольника одного поддерева до описанного прямоуголь-

[1] http://www.threadingbuildingblocks.org/

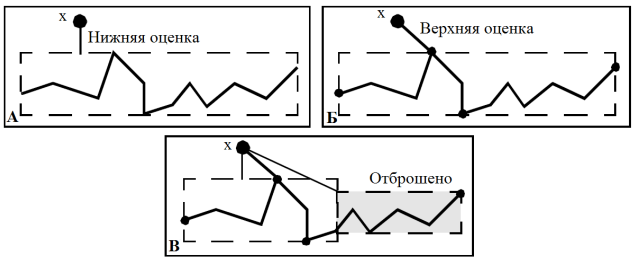


Figure 1: Алгоритм 3: А) Нижняя оценка; Б) Верхняя оценка; В) Отсечение группы отрезков, заведомо не являющихся ближайшими.

ника другого (заметим, что верхняя оценка сложнее нижней, но и вычисляется далеко не всегда).

Работа алгоритма для случая одной точки проиллюстрирована на рис. 1.

Заметим, что если первую строчку алгоритма 3 заменить с $\rho(X,E) = +\infty$ на $\rho(X,E) = d$, то он будет вычислять фукцию $\min(d, \rho(X,E))$.

Тем же образом можно ускорить проверку принадлежности точки многоугольнику (алгоритм 2): опять будем обходить дерево в ширину и будем отбрасывать такие пары поддеревьев, где отрезки лежат левее, выше или ниже точек, проверяя пересечения только с оставшимися после этого отрезками.

4. ВЫЧИСЛЕНИЯ НА ГРАФИЧЕСКОМ ПРОЦЕССОРЕ

Как уже отмечалось в п. 2.1, алгоритм расчёта значений поля расстояний на сетке для многоугольника методом полного перебора легко разделить на большое количество параллельных потоков вычислений: расстояние для каждой точки ищется независимо ото всех других. Это свойство делает данный алгоритм идеальным кандидатом для переноса на современные видеокарты. В качестве инструментария для такого переноса мы выбрали язык C$[2].

C$ является высокоуровневым языком программирования, который позволяет писать короткие и переносимые программы, использующие одинаковые исходные коды для различных архитектур графических процессоров (ГПУ). Следовательно, язык C$ не может предоставить программистам доступ к низкоуровневым особенностям конкретных графических платформ. Как следствие, он должен уметь использовать эти особенности автоматически, чтобы позволять достигать уровня производительности, сравнимого с ручной оптимизацией. Это достигается за счет сочетания различных приемов хранения данных и трансляции и оптимизации кода. Здесь приводится краткий обзор используемых приёмов, более подробную информацию можно найти в статье [15].

Данные, используемые в C$-программах на ГПУ, хранятся в массивах C$. В отличие от других языков, C$ не специфицирует ни формат представления данных, ни места, где они хранятся. Массивы C$ могут храниться в памяти хоста или ГПУ, или в обоих местах сразу, и перемещаться между ними прозрачно для программиста. Когда массивы C$ находятся в памяти ГПУ, формат их представления может меняться от одного исполняемого ядра к другому, чтобы наилучшим образом соответствовать решаемой задаче.

C$ представляет код ГПУ как направленный ациклический граф функциональных операций, или ФДАГ (функциональный ДАГ). Его трансляция в код ГПУ состоит из нескольких этапов. На первом этапе граф разбивается на подграфы, каждый из которых примерно соответствует одному вызову ядра на ГПУ. Далее каждый подграф обрабатывается отдельно. Для каждого из них выбирается наилучшее его отображение на ГПУ. Для этого ставится оптимизационная задача минимизации количества вычислений и обменов данными с памятью. Она является задачей геометрической оптимизации [21]. Проблема решается с использованием релаксации и метода барьеров, что позволяет получить оптимальные размеры развёртывания циклов. Они, в свою очередь, передаются на этап генерации кода, на которой также выполняется сокращение общих индексных выражений.

Оптимизации, используемые в системе C$, позволяют использовать быструю память на чипе. Для ГПУ NVidia это разделяемая статическая память, а для ГПУ AMD — большой регистровый банк. Для ГПУ AMD это также позволяет выполнять автоматическую векторизацию. Для вычислительно ёмких задач использование методов системы C$ позволяет достигать ускорения в несколько раз, а иногда и на порядок, по сравнению с тривиальной версией. При этом получаемый код является относительно небольшим и переносимым между различными архитектурами ГПУ.

Для нашей задачи система показала себя очень хорошо: код получился компактным и заработал на видеокартах трёх разных архитектур. При этом автоматическая оптимизация увеличила и без того немаленькую скорость работы алгоритма в 5-9 раз.

5. СРАВНЕНИЕ

Поскольку скорость работы алгоритмов полного перебора (неоптимизированного, оптимизированного и C$-версии) никоим образом не зависит от конкретных данных, с которыми они работают, не имеет никакого значение, на каких данных их тестировать (когда требовалось N точек, мы брали равномерную сетку $\sqrt{N}$x$\sqrt{N}$; в качестве M-угольника мы брали правильный). Однако совсем не так обстоит дело с иерархическим алгоритмом, и чтобы проверить, какой эффект оказывает структура данных на его работу, мы сгенерировали представительные тестовые наборы нескольких классов многоугольников:

- Набор псевдослучайных простых многоугольников, сгенерированных при помощи библиотеки RPG [3], основанной на эвристических алгоритмах, описанных в статье [22].
- Набор псевдослучайных ортогональных многоугольников, сгенерированных по алгоритму, описанному в работе [23]. Алгоритм основан на сложных эвристиках, позволяющих по заданному количеству вершин генерировать широкий класс сеточных многоугольников, с последующей преобразованием в ортогональные многоугольники общего вида.
- Набор случайных звёздных многоугольников, сгенерированных при помощи библиотеки RPG. Вычис-

[2] http://www.codeplex.com/cbucks

[3] http://www.cosy.sbg.ac.at/~held/projects/rpg/rpg.html

ления основаны на точном алгоритме [22], за полиномиальное время находящем все звёздные многоугольники с заданными вершинами.

В таблице 1 приведена зависимость (усреднённого) времени, которое требуется на вычисление поля расстояний для 50-угольников разных типов, от размеров сетки. Видно, что результаты для звёздных многоугольниках и многоугольниках общего вида практически идентичны, в то время как для ортогональных скорость работы выше в 1,3 раза. Скорее всего, это объясняется выбранной пространственной структурой данных: в основе kd-деревьев лежит разделение прямыми, параллельными осям координат, что отлично отражает структуру ортогональных полигонов. В то же время, сравнительна небольшая разница между ними и многоугольниками общего вида говорит о достаточно хорошем качестве приближения.

В таблице 2 приведено среднее время работы описанных алгоритмов для простых 50-угольников и сеток разных размеров. Для больших размеров сеток иерархический метод оказывается в 3 раза быстрее переборного, а ГПУ-метод — почти в 30 раз быстрее иерархического. Для больших многоугольников первый показатель увеличивается, а второй — падает (оба приблизительно до 10).

Заметим, наконец, что из всех приведённых алгоритмов только иерархический позволяет эффективно вычислять расстояние в некоторой полосе вокруг многоугольника, и в задачах, где этого достаточно, может работать быстрее ГПУ-алгоритма (для 1000-угольника, 2^{20} вершин и полосы толщиной 0.01 в квадрате $[0,1] \times [0,1]$ — в 4 раза).

6. ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ

В качестве примеров использования рассмотренных выше алгоритмов нам хотелось бы привести две задачи, одинаково активно использующие вычисление полей расстояний, но делающие это принципиально различными способами. В первой задаче сетка фиксирована, а многоугольники всё время разные, причём точный результат требуется лишь в узкой полосе, окружающей границу многоугольника. Во второй же задаче фиксирован многоугольник, а сетки всё время меняются, причём точный результат нужен для всех их узлов.

6.1 Расчёт области, свободной от нарушителей

Пусть на некоторой многоугольной плоской области передвигаются ищущие и активно уклоняющиеся объекты. Скорость первых — V, вторых — $U < V$. Пусть условием обнаружения искомого объекта является попадание в прямую видимость ищущего. Для заданной траектории $P(t), t \in [0,T]$ остаточной областью называется переменное во времени множество $A(t)$ такое, что в любой момент времени t^* при любом поведении искомых, они либо были обнаружены ранее, либо находятся гарантированно вне $A(t^*)$. Рассмотрим задачу нахождения нахождения $A_k = A(t_k)$, $t_k = k\Delta t$, $\Delta t = \frac{T}{N}$, $0 \le k \le N$.

В статье [7] предложен итерационный алгоритм приближённого расчёта остаточных областей, использующий для их представления поля расстояний. Для плоской задачи и условия прямой видимости он записывается следующим образом:

0) $DF_{A_0} = DF_{Vis_0}$

1) $DF_{old} = DF_{A_k} + \Delta U$,

2) $DF_{upd} = DF_{Vis_{k+1}}$,

3) $DF_{new} = \min(DF_{old}, DF_{upd})$,

4) $DF_{A_{k+1}} = DF_{ISO_0 DF_{new}}$.

где $Offset$ — операция сужения\расширения, $ISO_0 DF$ — нулевая линия уровня поля, а Vis_k— многоугольник видимости [24] для ищущего в момент времени t_k.

Как показано в работе [7], шаги 1, 3, 4 выполняются за линейное от числа узлов сетки время. Для вычисления многоугольника видимости требуется $O(M \ln M)$ операций, где M — число вершин границы области поиска [24] (очень быстрые алгоритмы этого типа представлены в библиотеке VisiLibity [25]). Таким образом, единственным потенциально медленным моментом остаётся вычисление поля расстояний многоугольника видимости. Заметим, однако, что достаточно вычислять поле расстояний в точках, удалённых от границы многоугольника видимости не более чем на $\Delta U + e_{max}$, где e_{max} — максимальная длина ребра сетки. Это позволяет воспользоваться наиболее быстрым вариантом иерархического алгоритма.

Неоптимизированный алгоритм полного перебора позволяет работать в интерактивном режиме с сетками размерами не более полутора тысяч вершин. Его оптимизированная версия расширяет это предел приблизительно до пяти тысяч. Оптимизированный иерархический алгоритм и C$-алгоритм позволяют работать с сетками размерами вплоть до ста тысяч вершин, причём на больших сетках иерархический алгоритм показывает несколько бóльшую производительность.

6.2 Генерация равномерных сеток

В работе [6] аппарат полей расстояний был использован для написания очень простого, но качественного генератора сеток (mesh generator). На вход алгоритму подаётся функция, вычисляющая значение поля расстояний для заданного тела в любой точке, некоторый прямоугольник, содержащий это тело, и желаемая средняя длина ребра сетки. Алгоритм начинает со случайной триангуляции прямоугольника, а затем постепенно подгоняет форму сетки к форме тела, процируя вершины сетки на границу тела (что легко делать при помощи поля расстояний). На выходе после определённого числа итераций (от десяти для простых тел до нескольких сотен для более сложных) получается достаточно однородная сетка, приближающая заданное полем расстояний тело.

Однако же при росте сложности геометрии исходного тела алгоритм становится малопрактичным: если тело задаётся многоугольником с несколькими тысячами сторон, то даже небольшие сетки могут строиться по несколько минут[4]. Дело в том, что на каждую итерацию алгоритма приходится четыре вычисления поля расстояний для всех узлов сетки. В выложенной автором алгоритма библиотеке поле расстояний для многоугольника рассчитывается неоптимизированным методом полного перебора. Мы заменили его нашими алгоритмами. Поскольку сетки при генерации меняются на каждой итерации, то с иерархическим алгоритмом генератор работает даже медленнее, чем с неоптимизированным полным перебором. Оптимизированный перебор ускоряет работу генератора всего в 2-5 раз. Зато перенос вычислений на ГПУ ускоряет генерацию в 20-30 раз.

[4]Мы пользовались реализацией, выложенной автором метода: http://www-math.mit.edu/~persson/mesh/

	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}	2^{17}	2^{18}	2^{19}	2^{20}
Простые	0.78	0.93	1.25	1.87	2.66	4.36	7.18	13.11	24.18	46.17	94.06
Ортогональные	0.78	0.93	1.25	2.80	2.66	3.74	5.77	10.76	18.41	36.19	72.23
Звёздные	0.93	1.10	1.25	1.87	3.90	4.05	6.71	12.63	22.31	44.93	95.01

Table 1: Время работы иерархического метода для разных размеров сеток (мс).

	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}	2^{17}	2^{18}	2^{19}	2^{20}
Перебор	0.31	0.62	1.40	2.18	4.37	9.05	17.47	36.03	70.98	141.49	287.04
Иерархический	0.78	0.93	1.25	1.87	2.66	4.36	7.18	13.11	24.18	46.17	94.06
ГПУ	0.01	0.01	0.02	0.04	0.07	0.11	0.24	0.47	0.94	1.88	3.75

Table 2: Время работы различных методов (мс).

7. REFERENCES

[1] R. Fabbri, O. M. Bruno, J. C. Torelli, and L. da F. Costa, "2d euclidean distance transform algorithms: A comparative survey," *ACM Computing Surveys*, , no. 40, pp. 2:1–2:44, 2007.

[2] J. Toriwaki and K. Mori, *Digital and Image Geometry: Advanced Lectures*, chapter Distance Transformation and Skeletonization of 3D Pictures and Their Applications to Medical Images, Springer, 2001.

[3] M. W. Jones, J. A. Baerentzen, and M. Sramek, "3d distance fields: A survey of techniques and applications," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12.

[4] D. Ferguson and A. Stentz, "Using interpolation to improve path planning: The field d* algorithm," *Journal of Field Robotics*, vol. 23.

[5] E. J. Jakubiak, R. N. Perry, and S. F. Frisken, "An improved representation for stroke-based fonts," in *Proceedings of ACM SIGGRAPH*, 2006.

[6] P.-O. Persson and G. Strang, "A simple mesh generator in matlab," *SIAM Review*, vol. 46.

[7] П. Воронин, "О новом подходе к расчёту и визуализации информационных множеств в задачах динамического поиска," in *Труды 18-й международной конференции ГрафиКон*, 2008.

[8] M. Akmal Butt and P. Maragos, "Optimum design of chamfer distance transforms," *IEEE Transactions on Image Processing*, vol. 7.

[9] R. Satherley and M. W. Jones, "Vector-city vector distance transform," *Computer Vision and Image Understanding*, vol. 82.

[10] J. A. Sethian, *Level Sets Methods and Fast Marching Methods. 2nd edition*, Cambridge University Press, 1999.

[11] F. Dachille and A. Kaufman, "Incremental triangle voxelization," in *Proceedings of Graphics Interface*, 2000.

[12] S. Mauch, "A fast algorithm for computing the closest point and distance transform," Tech. Rep., Technical Report caltechASCI/2000.077.

[13] C. Sigg, R. Peikert, and M. Gross, "Signed distance transform using graphics hardware," in *Proceedings of IEEE Visualization*, 2003.

[14] A. Sud, M. A. Otaduy, and D. Manocha, "Difi: Fast 3d distance field computation using graphics hardware," *Computer Graphics Forum*, vol. 23.

[15] А. В. Адинец and М. А. Кривов, "Методы оптимизации программ для современных графических процессоров," in *Сборник трудов Всероссийской научной конференции «Научный сервис в сети Интернет-2008: решение больших задач»*, 2008.

[16] J. Reinders, *Intel Threading Building Blocks: Outfitting C++ for Multi-core Processor Parallelism*, O'Reilly Media, Sebastopol, CA, 2007.

[17] E. Haines, *Graphics Gems IV*, chapter Point in Polygon Strategies, Academic Press Professional, Cambridge, MA, 1994.

[18] Ф. Препарата and М. Шаймос, *Вычислительная геометрия: введение*, Мир, 1989.

[19] D. Johnson and E. Cohen, "A framework for efficient minimum distance computation," in *Proceedings of IEEE Conf. On Robotics and Animation*, 1998.

[20] H. Samet, *The Design and Analysis of Spatial Data Structures*, Addison-Wesley, Reading, MA, 1990.

[21] Ф. П. Васильев, *Методы оптимизации*, Факториал Пресс, 2002.

[22] T. Auer and M. Held, "Heuristics for the generation of random polygons," in *Proc. of 8th Canadian Conf. Computational Geometry*, 1996.

[23] A. P. Tomás and A. L. Bajuelos, "Generating random orthogonal polygons," in *Proc. of CAEPIA-TTIA*, 2003.

[24] T. Asano, S. K. Ghosh, and T. C. Shermer, *Handbook of Computational Geometry*, chapter Visibility in the plane, Elsevier, 2000.

[25] K. J. Obermeyer, "The visilibity library," http://www.VisiLibity.org, 2008.

ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ

Расширенная версия статьи и реализация описанных алгоритмов доступны для скачивания по адресу http://www.primaler.ru/cg/.

Интерактивная трассировка лучей и фотонные карты на GPU

Владимир Фролов, Алексей Игнатенко
Факультет Вычислительной Математики и Кибернетики
Московский Государственный Университет имени М.В. Ломоносова, Москва, Россия
{vfrolov, ignatenko}@graphics.cs.msu.ru

Аннотация

Данная работа посвящена ускорению алгоритма фотонных карт с использованием графических процессоров (GPU). Фотонные карты - один из наиболее универсальных и реалистичных методов синтеза изображений, но и один из самых требовательных к вычислительным ресурсам. Мы представляем модифицированный алгоритм, работающий на CUDA-совместимых GPU. В работе также предложен подход организации трассировки лучей, позволяющий в высокой степени утилизировать ресурсы GPU. Наша реализация обратной трассировки лучей такая же быстрая, как и в передовых работах по интерактивной трассировке лучей на GPU.

Ключевые слова: *Интерактивная трассировка лучей, фотонные карты, GPU, глобальное освещение.*

1. ВВЕДЕНИЕ

Фотонные карты [1] – один из наиболее часто используемых алгоритмов расчета глобального освещения. Этот метод позволяет учесть самый широкий спектр эффектов – мягкие тени, вторичное освещение, каустики от отражающих и преломляющих объектов, спектральное разложение света. Благодаря своей универсальности он является стандартным методом расчета освещения в большинстве программ фотореалистичной визуализации. Условно метод фотонных карт можно разбить на три подзадачи – трассировка лучей, построение фотонной карты, сбор освещенности.

1.1 Трассировка лучей на GPU

Задача ускорения трассировки лучей на графических процессорах решалась в работах [2], [3], [4]. Для быстрого поиска пересечений луча и сцены, обычно используют специальные рекурсивные структуры пространственного разбиения – kd и BVH деревья.

В работе [2] применялись алгоритмы поиска в kd дереве, не требующие стека (или алгоритмы с коротким стеком) и трассировка пакетов лучей. В работе [3] также использовались пакеты большого размера, BVH деревья и один стек на весь пакет, реализованный на разделяемой памяти CUDA. В работе [4] наилучшие результаты были получены для алгоритма поиска в kd дереве с коротким стеком и использованием пакетов лучей. Позже мы рассмотрим эти работы подробнее.

Существуют две основные проблемы при реализации трассировки лучей на графических процессорах - отсутствие стека и нехватка регистров GPU. Мы вернемся ко второй проблеме после того как рассмотрим архитектуру GPU и технологию CUDA, а пока что остановимся на первой.

Для ускорения поиска пересечений в трассировке лучей используются иерархические структуры – kd и BVH деревья. Фактически, поиск пересечений луча и треугольника (или какого-то другого примитива) сводится к поиску в kd или BVH дереве. Стандартные алгоритмы поиска в этих деревьях имеют рекурсивную природу. Так как изначально на GPU нет стека, авторы работ [2] и [4] разработали алгоритмы поиска, не требующие стека и алгоритм, использующий короткий стек. Однако на современных GPU с архитектурой G80 и GT200, поддерживающих CUDA, возможна реализация стека на локальной памяти. Мы попробовали алгоритмы, не использующие стек и алгоритм с коротким стеком (на регистрах GPU) и сравнили их нашей реализацией классического алгоритма со стеком на локальной памяти (рис. 4,5). По-видимому, в нашем случае латентность локальной памяти покрывается высокой степенью параллелизма, поэтому алгоритм, использующий стек на локальной памяти, работал быстрее, чем любой из его аналогов, не использующих стек (или использующих короткий стек). Мы вернемся к этому после рассмотрения нашей реализации, где покажем, что проблема отсутствия быстрого стека для современных GPU менее существенна, чем проблема нехватки регистров.

1.2 GPU и CUDA

CUDA (Compute Unified Device Architecture) – это технология, позволяющая создавать программы для графических процессоров на подмножестве языка C++.

В CUDA ядром называется некоторая функция (обычно небольшая), работа которой распараллеливается на GPU. Все потоки выполняют эту функцию, и вначале работы отличаются только идентификатором потока (threadId).

Единица исполнения потоков в CUDA – группа из 32 потоков с названием warp. Можно считать, что GPU – это SIMD машина с шириной команды, равной размеру группы warp (то есть 32). Группы warp исполняются SIMD процессорами, которые также называются мультипроцессорами - Symmetric Multiprocessor (SM).

В SIMD парадигме так организованы вычисления, что если 31 поток пошел по одной ветви кода, но хотя бы один поток пошел по другой, процессору придется выполнить обе ветки. Поэтому расхождения потоков в местах ветвления кода внутри одной группы warp следует избегать. Это означает, что нужно так организовывать вычисления, чтобы минимизировать расхождения потоков по разным веткам кода в каждой группе из 32 потоков с подряд идущими идентификаторами потоков threadId.

Рассмотрим простой пример. Пусть имеется всего 64 потока. Мы хотим, чтобы половина из них выполняла какую-то одну задачу, а другая половина – другую. Для этого мы ставим некоторое условие if и ветку else. Условие “if (threadId %2 == 0)” будет порождать расхождение в группах warp, потому что потоки с четными номерами попадут внутрь тела условия if, а потоки с нечетными номерами - нет. Условие “if (threadId < 32)” не будет вызывать расхождений, так как все потоки первой группы warp с threadId от 0 до 31 попадут внутрь тела

условия, а все потоки следующей группы warp с threadId от 32 до 63 не попадут внутрь него.

В CUDA локальные переменные помещаются по возможности в регистры. Мультипроцессор GPU имеет ограниченное число регистров, причем вычисления организованы таким образом, что эффективность выполнения кода на GPU напрямую зависит от количества регистров, занимаемых ядром. Существует понятие занятости мультипроцессора. Занятость мультипроцессора – это отношение числа активных групп warp на мультипроцессоре к максимально возможному числу активных групп warp, которые могут работать одновременно на этом мультипроцессоре. Чем меньше занимаемое число регистров, тем выше занятость. Например, если ядро занимается в основном арифметическими вычислениями, снижение занимаемых регистров с 32 до 16 увеличит в два раза количество активных групп warp на мультипроцессоре и соответственно скорость возрастет в 2 раза.

Следует обратить особое внимание на работу с текстурами. Если ядро делает много текстурных выборок, латентность которых не скрывается вычислениями, эффективность может очень сильно зависеть от занятости. Так в архитектуре GT200 используется один текстурный кэш на 3 мультипроцессора (рис. 1). Это означает, что если на каждом мультипроцессоре количество активных групп warp возрастет в 2 раза, эффективность использования текстурного кэша может возрасти в 6 раз (при условии, что большинство потоков читают близкие или одни и те же области памяти). Поиск в kd дереве относится именно к такому случаю. Близкие по идентификатору потоки шагают примерно по одним и тем же узлам. Поэтому, несмотря на то, что латентность текстурной памяти не может быть покрыта вычислениями, при поиске в kd дереве, за счет правильного использования текстурного кэша может быть достигнута высокая эффективность.

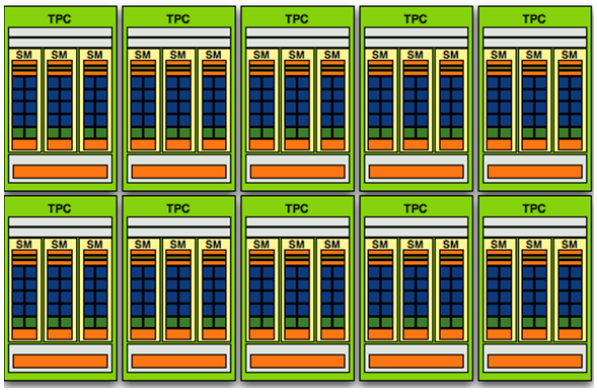


Рис. 1. Архитектура GT200. В каждом TPC на 3 SM один текстурный кэш.

Проблема нехватки регистров не является чем-то специфическим для CUDA, она существует во всех языках программирования для GPU – HLSL, GLSL, Brook, OpenCL, DirectX11 Compute Shaders и прочих.

1.3 Экономия регистров

Сложность реализации трассировки лучей на GPU в том, что алгоритм требует очень много локальных переменных. Причем, все переменные обычно используются. Компилятор старается поместить все переменные в регистры. А те, что не удается поместить в регистры, оказываются в медленной локальной памяти. Обычное пересечение луча и треугольника занимает уже больше 20 регистров (это нетрудно оценить, если выписать алгоритм и посмотреть, от каких переменных компилятор точно не сможет избавиться). То есть пересечение луча и треугольника в принципе не может быть размещено в малом количестве регистров. Но это только самая малая часть алгоритма трассировки лучей. Нужно еще выполнять поиск в ускоряющих структурах (в нашем случае это kd деревья). Также нужно делать затенение, которое может быть очень сложным и занимать произвольное число регистров и локальной памяти.

В случае критической нехватки регистров в качестве стандартного решения используется паттерн uber-kernel (который также называют mega-kernel). Сложный код делится некоторым образом на части. В ядре присутствуют все части, но каждая в своей ветке if. Также имеется флаг, отвечающий за то, какая часть кода должна выполняться. Во время выполнения, процессор может периодически прыгать с одной части кода на другую, сохраняя некоторые важные данные в разделяемой или локальной памяти. Суть такого подхода в том, что он позволяет использовать одни и те же регистры для разных переменных.

Рассмотрим пример того, как мог бы быть реализован паттерн uber-kernel для трассировки лучей. Во время поиска в kd дереве наступает момент, когда луч дошел до листа дерева. Тогда флаг устанавливается в «intersect» и делается переход в начало ядра. Нужные данные могут быть сохранены в разделяемой памяти. Далее считаются пересечения. Если пересечение найдено в пределах узла kd дерева, то данные сохраняются в разделяемой памяти, флаг устанавливается в «shade» и делается переход в начало ядра. После чего ядро начнет выполнять затенение. Если же пересечение не было найдено, флаг устанавливается в «traverse» и выполняется переход в начало ядра. Затем ядро продолжит поиск в kd дереве.

У такого подхода есть ряд серьезных недостатков:

1. Очень усложняется код
2. Самая сложная часть является лимитирующей по регистрам
3. Трудно профилировать код – невозможно понять, какая же именно часть работает медленно.

Наиболее существенный недостаток – второй. Предположим у нас есть три части алгоритма, занимающие разное число регистров. Например, поиск в kd дереве – 16 регистров, подсчет пересечений – 24 а затенение – 32 регистра. Результирующее ядро будет занимать максимум из этих трех чисел. Это означает, что мы очень сильно потеряем в эффективности на поиске в kd дереве из-за того что число активных групп warp уменьшится ровно в 2 раза. При этом, эффективность текстурного кэша может упасть более чем в два раза (в 2-6 раз на архитектуре GT200). А так как поиск в kd дереве – это нагрузка исключительно на текстуры, то и общая скорость может снизиться довольно сильно по сравнению с теоретическим максимумом.

Судя по всему, авторы работ [2] и [3] используют паттерн uber-kernel, так как они утверждают, что реализовали весь алгоритм трассировки лучей в одном ядре. Причем, в работе [3] говорится, что при использовании полного затенения по фонгу и трассировки теневых лучей, занятость мультипроцессоров падает с 63% до 38% по сравнению с простым затенением и трассировкой только первичных лучей. Таким, образом, uber-kernel не позволит сделать код по-настоящему эффективным, если потребуется реализация

сложного затенения, распределенной трассировки лучей или каких-либо других дополнительных эффектов.

Авторы работы [4] использовали совсем другой подход. Они разделили алгоритм поиска на 3 части (на самом деле больше, но мы решили не усложнять изложение) – спуск по kd дереву (kd_down), подъем по kd дереву (kd_up) и подсчет пересечений (intersect). Трассировка ведется блоками. Сначала запускается ядро kd_down, осуществляющее спуск. Как только все лучи блока дошли до листа, ядро спуска останавливается. Запускается ядро подсчета пересечений intersect. Когда все лучи в блоке завершили подсчет пересечений, запускается ядро kd_up. Затем снова ядро спуска kd_down для тех лучей, которые не нашли пересечение. При этом каждый раз происходит остановка и запуск ядра для всего блока, даже для тех лучей, которые уже ударились о поверхность (просто эти потоки сразу же завершают свою работу). Фактически, авторы работы [4] просто разрезали код на части. Ядра kd_down, intersect и kd_up работают последовательно в цикле до тех пор пока все лучи не найдут пересечения или не вылетят за пределы kd дерева.

Это лучше, чем uber-kernel, но проблема такого подхода в том, что на сложных сценах происходит слишком много переключений ядер. Каждое переключение - это дополнительные расходы времени, прежде всего, на работу с памятью. Каждый раз, когда запускается новое ядро, значения всех локальных переменных теряются. Промежуточные данные нужно сохранять в DRAM GPU и загружать из нее, что снижает скорость. В самой статье [4] детали реализации не описаны, поэтому нам пришлось проанализировать исходные коды их трассировщика лучей. Мы полагаем, что в значительной степени производительность программы из работы [4] объясняется именно тем, что авторы разделили код на части и сократили таким способом число занимаемых регистров для каждой отдельной части (ядра).

2. ДЕКОМПОЗИЦИЯ ТРАССИРОВКИ ЛУЧЕЙ

Мы предлагаем новый подход для организации трассировки лучей на CUDA, лишенный недостатков, присущих uber-kernel и множественным переключениям ядер. Основная идея нашего подхода та же самая, что и в работе [4] - нужно разделить алгоритм трассировки лучей на несколько ядер, каждое из которых выполняет только одну узкую задачу. Мы разделили алгоритм на 3 части – поиск в kd дереве, подсчет пересечений и затенение (рис. 2). Отличие от работы [4] заключается в том, что переключение ядер производится всего 3 раза для одного типа лучей (первичных, теневых, отраженных).

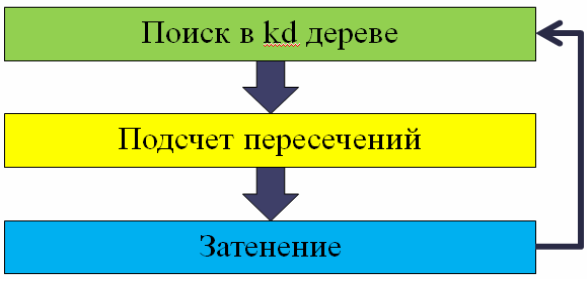


Рис. 2: декомпозиция алгоритма трассировки лучей.

Если отделить затенение от всего остального достаточно просто, то для того чтобы разделить поиск в kd дереве и подсчет пересечений мы изменили сам алгоритм поиска. Идея формулируется следующим образом: если мы выполняем какой-то код (например, поиск в kd дереве), то мы должны стараться выполнять его как можно дольше, не останавливая ядро и не теряя значений локальных переменных. На рис. 3 показан классический код поиска в kd дереве. На рис. 4 и 5 изображена наша модификация алгоритма.

```
boolean kd_tree_traversal(Ray ray, Hit* out_hit):
  (hit,t_near,t_far) = IntersectRayAABB(ray, scene_AABB)
  if (!hit): return

  while (true):
    while (!node.Leaf()):
      t_split  = IntersectRayPlane(ray, node.Plane())
      if(t_split < t_near):
        node = Near(node)
      else if(t_split > t_far):
        node = Far(node)
      else:
        node = Near(node)
        stack.push(t_far, &Far(node))

    (hit,t) = IntersectAllPrimitivesInLeaf(ray, out_hit)
    if (hit && t <= t_far):
      return true

    if (stack.empty()):
      return false
    t_near = t_far
    (t_far, farNodeAddress) = stack.pop()
    node = GetNode(farNodeAddress)
```

Рис. 3: Псевдокод классического алгоритма поиска в kd дереве.

```
void kd_tree_traversal_only(Ray ray, Hit* out_hit)
  (hit,t_near,t_far) = IntersectRayAABB(ray, scene_AABB)
  if (!hit): return

  while (true):
    while (!node.Leaf()):
      t_split  = IntersectRayPlane(ray, node.Plane())
      if(t_split < t_near):
        node = Near(node)
      else if(t_split > t_far):
        node = Far(node)
      else:
        node = Near(node)
        stack.push(t_far, &Far(node))

    STORE_DATA_IN_LEAF_LIST(node, t_far)

    if (stack.empty()):
      return false
    t_near = t_far;
    (t_far, farNodeAddress) = stack.pop()
    node = GetNode(farNodeAddress)
```

Рис. 4: Наша модификация классического алгоритма.

Таким образом, на первом этапе мы проходим kd дерево насквозь, составляя для каждого луча список листьев,

которые он посетил. А на следующем шаге мы проходим по списку листьев и считаем в них пересечения.

```
boolean intersect_in_leafes(Ray ray, Hit* out_hit,
                            LeafList* leaf_list):
  while (!leaf_list.End()):
    (node,t_far) = LOAD_DATA_FROM_LEAF_LIST(leaf_list)
    (hit,t) = IntersectAllPrimitivesInLeaf(ray, out_hit)
    if (hit && t <= t_far):
      return true
  return false
```

Рис. 5: Алгоритм подсчета пересечений в списке листьев.

Стек реализован на локальной памяти. Мы пробовали различные безстековые алгоритмы из работ [2] и [4], в частности алгоритм “kd-tree restart” и короткий стек на регистрах. Но они дали худший результат чем классический вариант. По-видимому, это связано с тем, что благодаря высокой занятости мультипроцессоров, нам удалось скрыть латентность локальной памяти за счет высокой степени параллелизма.

Конечно, с алгоритмической точки зрения, наш подход менее эффективен, чем классический вариант, так как мы все время проходим kd дерево насквозь. Но на практике, за счет того, что мы выделили поиск в kd дереве в отдельное ядро и уместили его в 16 регистров, наша реализация трассировки лучей столь же быстрая, как и в работе [2]. Благодаря тому что мы уложились в 16 регистров для поиска в kd дереве, нам удалось добиться 67% занятости мультипроцессора на архитектуре G80 и 100% занятости на архитектуре GT200. Более того, на сценах размером вплоть до 1.5 миллиона треугольников поиск в kd дереве не является узким местом.

По сравнению с работой [2], наш подход имеет ряд преимуществ:

1. За счет того, что мы провели декомпозицию, нам удалось измерить скорость работы отдельных частей алгоритма трассировки лучей, что не может быть сделано, используя подход из работы [2].
2. Все части системы слабо связаны и легко заменяемы. Например, можно заменить kd деревья на BVH и придется поменять только одно ядро – то что отвечает за поиск в kd дереве. Но весь остальной код останется без изменений. То есть наш подход будет работать и для BVH деревьев. Аналогично,
3. Ядро, отвечающее за затенение может быть сколь угодно сложным и это никак не влияет на производительность остальной части системы.
4. Во время прохода по kd дереву, количество ветвлений в warp-ах значительно меньше, чем в случае с объединенным вариантом, так как никакие лучи не простаивают, пока другие считают пересечения в листьях (такое происходит если какая-то часть лучей из warp-а попала в лист, а другая часть не попала в него). В нашем случае во время поиска в kd дереве пересечения вообще не считаются.

Недостаток нашего подхода в том, что при больших разрешениях тратится довольно много памяти, потому что для каждого пиксела нужно сохранять список листьев, которые прошел луч. Эта проблема, однако, легко решается путем разбиения экрана на несколько более мелких частей и их последовательной либо параллельной (на нескольких GPU) обработки.

Мы также сравнивали наш алгоритм с подходом, реализованным в работе [4]. Для этого мы уменьшали размер списка листьев в n раз и запускали наши ядра последовательно, в цикле из n итераций. На видеокарте GF8800GTX производительность падала в 1.5 раза даже для n=2. Это означает, что для CUDA и текущих архитектур G80 и GT200 подход с простым разрезанием кода, использованный в работе [4] (где использовалась GPU с иной архитектурой), будет работать намного хуже нашей декомпозиции.

2.1 Результаты для обратной трассировки лучей

Мы провели сравнение с другими работами на сцене Conference Room (280 тысяч треугольников). Эта сцена была выбрана по причине того, что она упоминается практически во всех статьях по интерактивной трассировке лучей. Замеры были сделаны в разрешении 1024x1024 на GF8800GTX.

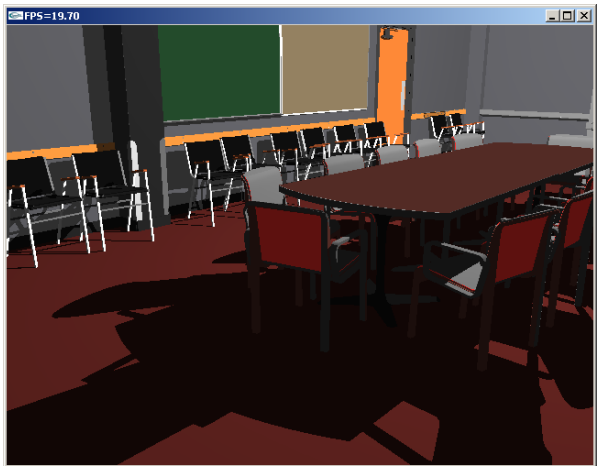


Рис. 6. Conference Room (280 тысяч треугольников). На картинке разрешение 640x480, тени от одного точечного источника, GTX260.

Работа	Ускоряющая структура	Только первичные лучи	Первичные лучи + теневые от одного точечного источника
Наша реализация	kd tree	12 FPS	5.8 FPS
[2]	kd tree	16.7 FPS	6.7 FPS
[3]	BVH	16 FPS	6.1 FPS
[4]	kd tree	15.2* FPS	7* FPS
[5]	BVH	8** FPS	-

Таблица 1: Сравнение скорости обратной трассировки лучей с другими работами. Числа приведены в кадрах в секунду (FPS).

В работах [2], [3] и [4] для первичных лучей использовалась трассировка пакетов, поэтому падение производительности при переходе к вторичным лучам в них больше чем в 2 раза. Мы не использовали пакеты в явном виде (неявно, на уровне групп warp, они конечно присутствуют). Поэтому в нашей

работе меньше падение производительности при переходе к вторичным лучам – теневым и отраженным.

На рис. 8 приведен график зависимости производительности от количества треугольников для сцены, состоящей из стульев (рис. 9). Один стул содержит 1464 треугольника. Мы добавляем стулья в сцену, расставляя их по квадрату, и стараемся сделать так, чтобы все стулья были видимы и занимали экран полностью. На графике каждая следующая отметка содержит в 4 раза больше треугольников, чем предыдущая. Измерялось время только для первичных лучей. Мы откладываем кривые 1/log(N) и 1/(log(N)*log(N)) от первой точки, чтобы оценить характер кривой зависимости производительности от количества треугольников (где N – количество треугольников).

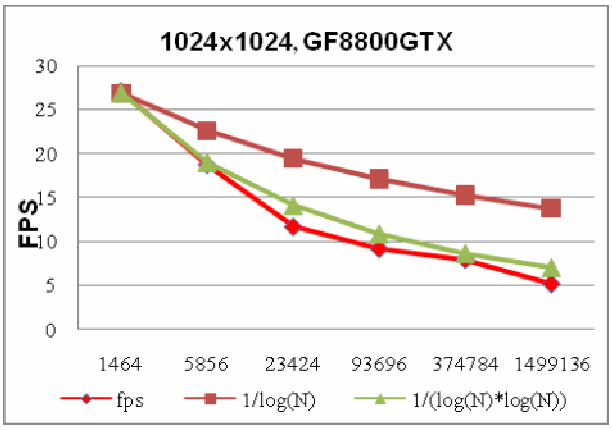


Рис. 7: график зависимости производительности от количества треугольников в тестовой сцене из стульев

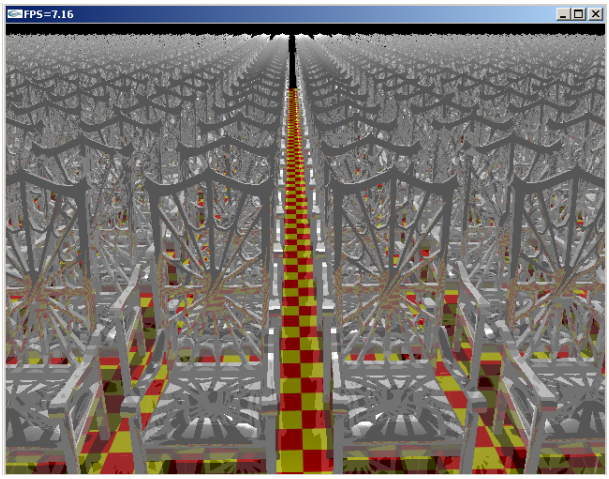


Рис. 8. Сцена из стульев, на которой производились замеры производительности. На картинке разрешение 640x480, один уровень отражения и тени от точечного источника, 1.5 миллиона треугольников, GTX260.

2.2 Измерения скорости отдельных частей алгоритма

На сцене Conference Room (280 тысяч треугольников):

- поиск в kd дереве занимает 15% общего времени
- подсчет пересечений - 74% общего времени
- все остальное (в том числе затенение) - 11%.

Таким образом, мы можем утверждать, что наша реализация поиска в kd дереве крайне эффективна, потому как занимает всего 15% общего времени. Это означает, что на современных GPU (с архитектурами G80 и GT200) не требуются специальные безстековые алгоритмы, о которых говорится в работах [2] и [4].

На сцене из стульев для 1.5 миллиона треугольников:

- поиск в kd дереве занял 46% времени.
- подсчет пересечений – 45% времени.

Это является довольно неплохим результатом, так как на такой сцене kd дерево очень глубокое и равномерное. То есть это худший пример для нашего алгоритма.

2.3 Трассировка фотонов

Особенность трассировки фотонов заключается в том, что в отличие от лучей, идущих из глаза, теневых или даже отраженных лучей, начальные позиции и направления фотонов распределены случайно. Причем, это верно как для первичных фотонов так и для отраженных. Если генерировать случайное направление для каждого фотона, то лучи, представляющие фотоны, будут очень сильно расходиться. Это создает проблемы для GPU так как сильно расходящиеся лучи порождают множество ветвлений в группах warp. В результате производительность может упасть в несколько раз. Вместо того чтобы вычислять случайно направление для каждого фотона, мы поступили следующим образом:

- разделили все фотоны на группы по 32.
- Для каждой группы вычислили случайное направление
- Для каждого фотона в группе ввели небольшое отклонение от заданного направления

Таким образом, мы трассируем фотоны группами, в каждой из которых фотоны расходятся не сильно друг от друга. Для отраженных фотонов мы поступаем аналогичным образом, хотя для них эта техника менее эффективна, так как даже после сортировки по узлам kd дерева, фотоны с близким адресом могут лежать достаточно далеко друг от друга на поверхности.

3. СБОР ОСВЕЩЕННОСТИ

После завершения трассировки фотонов в некоторых точках сцены происходит сбор освещенности с ближайших фотонов. Существует несколько способов выбирать, в каких именно точках собирать освещенность и как визуализировать фотонную карту. Мы выбрали простой, но универсальный способ - трассируем лучи из виртуального глаза для каждого пиксела и собираем освещенность во всех точках, где луч пересек сцену. Мы не производили какую-либо фильтрацию. Этот способ часто называют “визуализацией фотонной карты напрямую”.

Обычно задача сбора освещенности формулируется следующим образом: в каждой точке нужно найти N ближайших фотонов, сложить их энергию, возможно с некоторой зависимостью по расстоянию, и поделить полученную сумму на площадь поверхности сферы, радиус которой равен расстоянию до самого дальнего фотона. Этот алгоритм хорош тем, что для заданного числа N он динамически выбирает радиус сбора в зависимости от того, сколько фотонов лежит вблизи точки сбора. Если фотонов мало, радиус сбора большой, фотонов много – маленький. Качество получаемого изображения зависит только от числа

* Результаты измерены на ATI X1900 XTX с другим ракурсом

** Результаты измерены на GTX280 и масштабированы на 8800 GTX

N. Однако такой алгоритм создает множество проблем. Во-первых, чтобы эффективно искать N ближайших фотонов, каждый раз после трассировки необходимо построить специальное kd дерево, содержащее в своих узлах фотоны. Можно использовать и другую ускоряющую структуру, например регулярную сетку, как в работе [5]. Во-вторых, алгоритм поиска N ближайших фотонов довольно сложен для эффективной реализации на GPU, хотя сделан в работах [5], [6] и [7].

Мы использовали более простой алгоритм - выбрали некоторый фиксированный радиус сбора и взяли для оценки освещения все фотоны, попавшие в сферу с заданным радиусом сбора. Такой алгоритм намного проще реализовать на GPU. Проще в данном случае означает, что можно добиться от него более высокой утилизации ресурсов GPU, чем от алгоритма, выполняющего поиск N ближайших фотонов. Более того, этот алгоритм дал нам возможность не перестраивать ускоряющие структуры для фотонов.

3.1 Построение пространственного индекса

Если сделать предположение, что фотоны могут быть распределены только по поверхности (отказаться от объемных эффектов), то для заданной геометрии можно построить kd-дерево для фотонов всего один раз. Мы использовали SAH kd дерево для геометрии и разбили каждый его лист равномерно до некоторой величины, сравнимой с радиусом сбора.

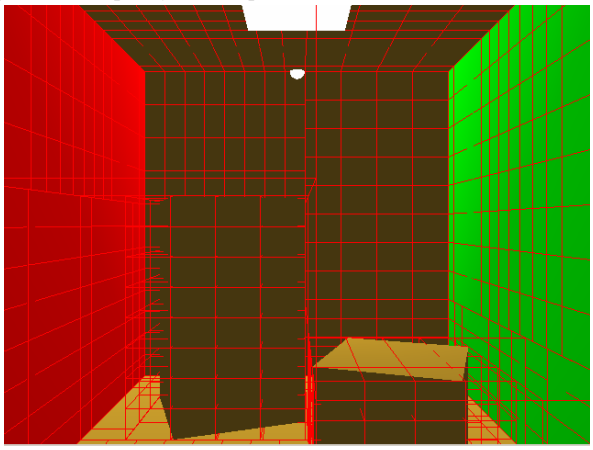


Рис. 9. kd дерево для фотонов.

После окончания трассировки фотонов нужно определить для каждого фотона, в какой узел kd дерева он попал. А затем нужно отсортировать фотоны по узлам kd дерева. И в каждый узел записать смещение для списка фотонов и их число в узле. Мы определяем индекс узла kd дерева для каждого фотона с помощью поиска в kd дереве на GPU. Затем мы сортируем по узлам kd дерева фотоны, используя линейную сортировку индексов фотонов на CPU с помощью быстрых списков. После этого мы загружаем отсортированные индексы на GPU и за одну операцию копирования для каждого массива (мы использовали принцип хранения Structure Of Arrays – SOA, поэтому фотоны храняться в нескольких массивах) перемещаем все остальные данные о фотонах.

4. МНОГОПРОХОДНОЙ АЛГОРИТМ

Первая проблема, с которой мы столкнулись при реализации фотонных карт на GPU – это учет фотонов, отразившихся от поверхности один или более раз. Для того чтобы обрабатывать фотоны параллельно, в CUDA на каждый фотон заводится свой поток. И каждый поток должен записывать информацию об ударах в фотонную карту строго в свою область памяти.

Пусть на начальном этапе имелось n фотонов и на каждый фотон отводится k байт. Пусть глубина трассировки фотонов равна m. Тогда даже если в соответствие с принципом “русской рулетки” на каждом этапе отражается только часть фотонов (например половина), нам все равно потребуется k*n*m байт. Нетрудно посчитать, что на CPU в случае если каждый раз отражается только половина фотонов, нам потребуется не больше чем k*m*2 байт.

Следующая проблема заключается в том, что для достижения хорошего качества нужно больше фотонов, чем способно поместиться в памяти GPU. Обе проблемы усугубляются тем, что для эффективного доступа и соблюдения правила объединения запросов к памяти в CUDA (coalescing), мы были вынуждены потратить 56 байт на фотон (3 массива из элементов float4 и один массив float2). Мы сохраняли позицию фотона, направление, цвет, индекс узла kd дерева, флаг активности фотона и некоторую информацию о поверхности, в которую ударился фотон. Мы могли бы сократить эту информацию до 48 байт, используя битовые операции, но от этого ничего принципиально бы не изменилось. Памяти все равно бы тратилось слишком много.

Мы модифицировали многопроходный алгоритм, предложенный в статье [9] и переложили его на GPU. Сначала мы испускаем некоторое число (например, один миллион) фотонов из источника света. Трассируем их, строим фотонную карту и собираем освещенность. Затем, мы обрабатываем фотоны – часть из них отражается, часть умирает. Все активные фотоны трассируется. Строится фотонная карта, собирается освещение. При этом мы используем те же области памяти, что и для первичных фотонов. Так как с первичных фотонов мы уже собрали освещение, они нам больше не нужны, и мы можем задействовать их память для вторичных фотонов. Аналогично мы поступаем с третичными, четвертичными фотонами и так далее. Таким образом, мы можем обработать любое число переотражений фотонов, имея ограниченное количество памяти. Та же самая идея позволяет нам рассчитать фотонную карту для вообще любого числа фотонов. Мы обрабатываем фотоны последовательно - блоками по N фотонов. Если у нас имеется 70 миллионов фотонов, то мы можем разделить их на блоки, например по 2 миллиона фотонов, и за 35 проходов обработать все 70 миллионов. Таким способом можно легко распараллелить расчет фотонной карты на несколько GPU. Часть проходов рассчитывается на одной GPU, часть на другой. Результирующее освещение в каждой точке просто складывается. Это дает возможность использовать суперкомпьютер Tesla для расчета фотонных карт достаточно простым способом.

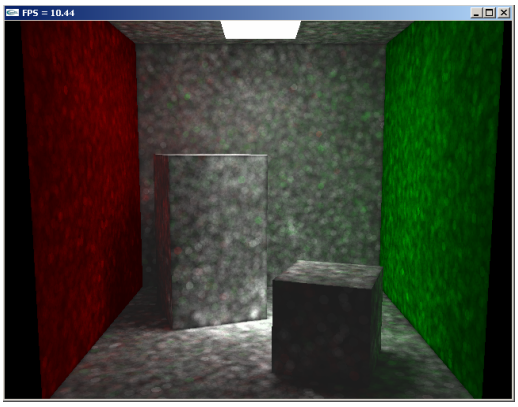


Рис. 9. Cornel Box, 260 тысяч фотонов, 1 проход, 10.4 fps.

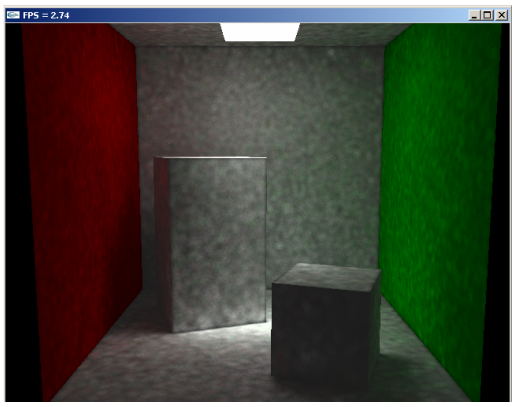


Рис. 10. Cornel Box, 1 миллион фотонов, 1 проход, 2.7 fps.

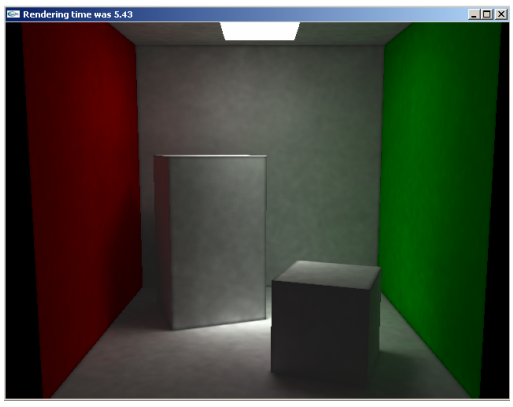


Рис. 11. Cornel Box, 2 миллиона фотонов, 8 проходов, время рендеринга – 5.4 секунд.

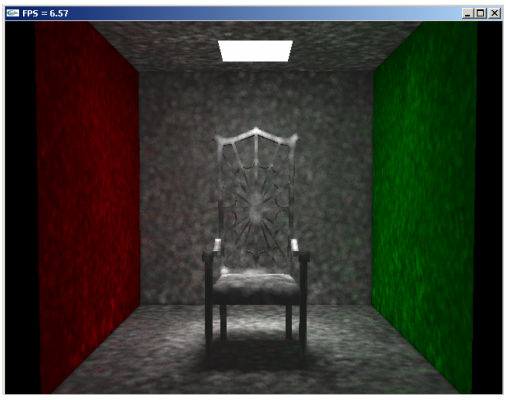


Рис. 12. Cornell Box со стулом, 260 тысяч фотонов, 1 проход 6.6 fps.

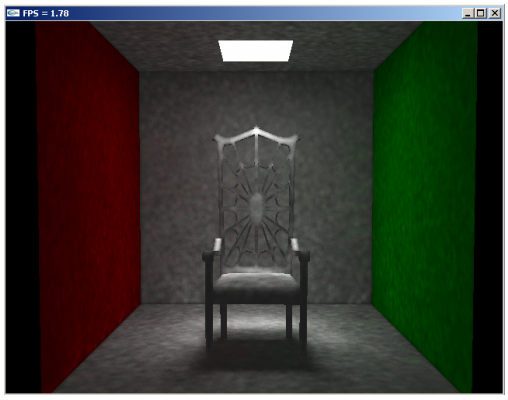


Рис. 13. Cornell Box со стулом, 1 миллион фотонов, 1 проход, 1.8 fps.

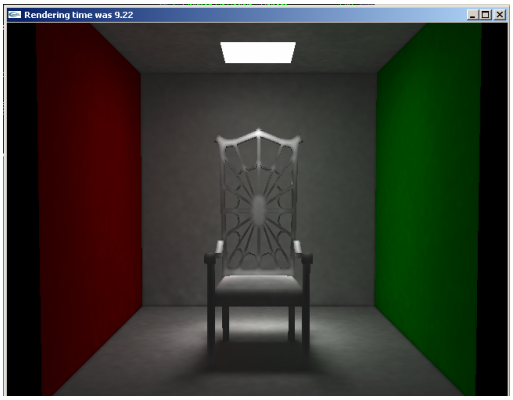


Рис. 14. Cornell Box со стулом, 2 миллиона фотонов, 8 проходов, время рендеринга – 9.26 секунд.

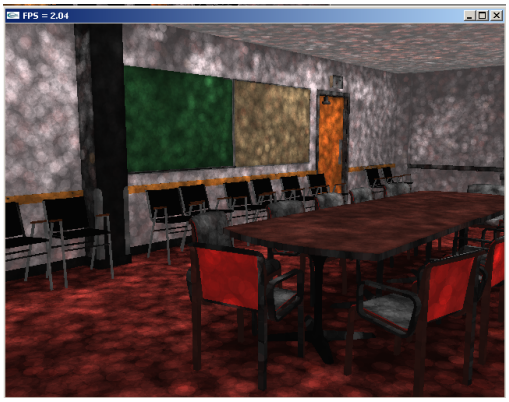


Рис. 15. Conference Room (280 тысяч треугольников), 260 тысяч фотонов, 2.0 fps.

Рис. 16. Conference Room, 1 миллион фотонов, время рендеринга – 1.64 секунды.

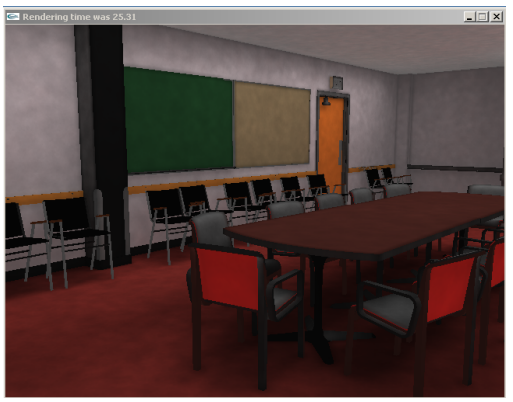


Рис. 17. Conference Room, 2 миллиона фотонов, 8 проходов, время рендеринга – 25 секунд.

На сцене Conference Room под потолком располагаются 4 протяженных источника света, которые не были визуализированы.

5. РЕЗУЛЬТАТЫ

На рисунках 9-17 представлены результаты работы нашей программы. Снимки получены на машине с GF8800GTX, Core 2 Quad 2.4 GHz. Глубина трассировки фотонов равнялась двум. Все наши демонстрационные программы могут быть найдены на сайте http://ray-tracing.ru. Системные требования – CUDA 2.1 или выше, любая видеокарта с поддержкой CUDA и достаточным количеством видеопамяти. В работе [6] экранизация (рендеринг) сцены Cornell Box занимал порядка минуты, но мы считаем, что проводить сравнение с работой [6] некорректно, так как в ней использовалась очень старая аппаратура (GeForce FX 5900 Ultra). Мы сравнивали нашу реализацию с работами [7], [8] и библиотекой OpenRT.

В работе [7] использовалась фотонная карта только для расчета каустиков. Тестировались две сцены размером 3 и 20 тысяч треугольников. В первой сцене для расчета каустиков было задействовано 200 тысяч фотонов и освещение собиралось с 50 ближайших – 12 fps. На второй сцене из 20 тысяч треугольников задействовано 400 тысяч фотонов, сбор освещенности с 40 ближайших - 7.5 fps. Результаты измерены на видеокарте GF8800 ULTRA в разрешении 800x600.

В работе [8] глобальное освещение вычислялось методом фотонных карт с последующим финальным сбором (Final Gathering - FG) и техникой кэширования вторичного освещения (irradiance caching). Прямое освещение вычислялось обратной трассировкой лучей. Использовались сцены размером от 20 до 80 тысяч треугольников. Было задействовано от 200 до 400 тысяч фотонов. На видеокарте GTX280 в разрешении 1024x1024 было достигнута скорость визуализации 1.5-4.2 fps. Можно условно считать, что GTX280 в 3 раза лучше, чем GF8800GTX. По крайней мере, это верно для нашей реализации трассировки лучей.

По сравнению с работой [8], мы можем обработать гораздо больше фотонов (примерно в 3-4 раза больше) за то же самое время потому что мы не перестраиваем ускоряющие структуры и упрощаем алгоритм сбора освещенности. Более того, мы можем обрабатывать произвольное число фотонов благодаря многопроходному алгоритму. Однако так как мы не использовали кэш освещения (что делалось в работе [8]) и считали прямое освещение фотонными картами, наши изображения достаточно шумные для малого числа фотонов. Также мы не реализовывали метод Final Gathering, поэтому нам сложно привести более точное сравнение с работой [8].

Из существующих решений стоит упомянуть библиотеку OpenRT, которая на сцене Conference Room показывает скорость 1.76 кадров в секунду на 17 двуядерных машинах. Так как разработчики OpenRT не выкладывают данные о количестве задействованных фотонов и о том, как было рассчитано первичное освещение, нам трудно привести точное сравнение.

6. ЛИТЕРАТУРА

[1] Jensen, H. W. 2004. *A practical guide to global illumination using ray tracing and photon mapping.* In ACM SIGGRAPH 2004 Course Notes (Los Angeles, CA, August 08 - 12, 2004). SIGGRAPH '04. ACM, New York, NY, 20.

[2] Popov S., Günther J., Seidel H.-P., Slusallek P. *Stackless KD-Tree Traversal for High Performance GPU Ray Tracing.* In Proceedings of the EUROGRAPHICS conference, vol. 26 (2007), Number 3.

[3] Günther J., Popov S., Seidel H.-P., Slusallek P. *Realtime Ray Tracing On GPU With BVH-Based Packet Traversal.* In Proceedings of the IEEE/Eurographics Symposium on Interactive Ray Tracing. IEEE Symposium on Volume, Issue, 10-12 Sept. 2007. p. 113 – 118.

[4] Horn, D. R., Sugerman, J., Houston, M., and Hanrahan, P. 2007. *Interactive k-d tree GPU raytracing.* In Proceedings of the 2007 Symposium on interactive 3D Graphics and Games (Seattle, Washington, April 30 - May 02, 2007). I3D '07. ACM, New York, NY, 167-174.

[5] Lauterbach C., Garland M., Sengupta S., Luebke D., Manocha D., *Fast BVH construction on GPU.* In Proceedings of the EUROGRAPHICS conference , vol. 28, number 2, 2009.

[6] Purcell, T. J., Donner, C., Cammarano, M., Jensen, H. W., and Hanrahan, P. 2005. *Photon mapping on programmable graphics hardware.* In *ACM SIGGRAPH 2005 Courses* (Los Angeles, California, July 31 - August 04, 2005). J. Fujii, Ed. SIGGRAPH '05. ACM, New York, NY, 258.

[7] Zhou, K., Hou, Q., Wang, R., and Guo, B. 2008. *Real-time KD-tree construction on graphics hardware.* In *ACM SIGGRAPH Asia 2008 Papers* (Singapore, December 10 - 13, 2008). J. C. Hart, Ed. SIGGRAPH Asia '08. ACM, New York, NY, 1-11.

[8] Wang, R., Zhou, K., Pan, M., and Bao, H. 2009. *An efficient GPU-based approach for interactive global illumination. ACM Trans. Graph.* 28, 3 (Jul. 2009), 1-8.

[9] Bounded A., Paulin M., Pitot P., Pratmarty D. *Low Memory Spectral Photon Mapping.* In Proceedings of the WSCG conference, 2004.

7. ABSTRACT

This paper is about GPU accelerated ray tracing and photon mapping. We present a modified photon mapping algorithm that is entirely executed on CUDA capable GPUs. We also propose an efficient approach to the GPU ray tracing. Our realization of backward ray tracing is as fast as in top papers on GPU ray tracing.

8. ABOUT THE AUTHORS

Vladimir Frolov is the fifth year student of Moscow State University. His contact e-mail is vfrolov@graphics.cs.msu.ru.

Alexey Ignatenko is a PhD researcher at Computational and Cybernetics department of Moscow State University. His research interests include photorealistic 3D rendering, 3D modeling and reconstruction, image based rendering and adjacent fields. His contact e-mail is ignatenko@graphics.cs.msu.ru.

Интерактивная трассировка лучей на графическом процессоре

Денис Боголепов, Виталий Трушанин, Вадим Турлапов
Нижегородский государственный университет им. Н.И. Лобачевского
Факультет вычислительной математики и кибернетики, Нижний Новгород, Россия
bogdencmc@inbox.ru, vitaly.trushanin@rambler.ru, vadim.turlapov@cs.vmk.unn.ru

Abstract

High-performance GPU-implementation of the backward ray tracing algorithm is proposed. High performance is provided by new ray-primitive intersection algorithms without dynamic branches, uniform grid accelerating structure with “proximity cloud” information; by some optimizing techniques for ray tracing algorithm on GPU architecture. The implementation was done as cross platform on GLSL and provides real-time rendering for scenes with 300K and more triangles.

Keywords: *Ray Tracing, Interactive Graphics, GPGPU, GPU Ray Tracing, GLSL*

1. ПРЕДШЕСТВУЮЩИЕ РАБОТЫ

Задача реализации алгоритма обратной трассировки лучей в реальном времени имеет достаточно богатую историю [1]-[5], новый импульс которой придало появление графических процессоров, способных выполнять вычисления общего назначения. Расширенные возможности программируемости графических процессоров и появление технологий NVIDIA CUDA и AMD Stream привели к пересмотру ряда принципов, заложенных в реализацию трассировки лучей для графического процессора. Интересным примером служит работа [6], в которой авторы одни из первых реализовали полноценный алгоритм трассировки лучей в *одном* фрагментном шейдере. Среди последних работ, использующих технологию NVIDIA CUDA, следует отметить проект [7], в которой трассировка лучей реального времени реализована с учетом теней, отражений и преломлений для достаточно сложных сцен (сотни тысяч треугольников).

Цель данной работы – реализация алгоритма обратной трассировки лучей на графическом процессоре с поддержкой всех основных возможностей метода: визуализация теней, отражений, преломлений, поддержка прозрачных объектов и текстурирования. В качестве платформы выбран графический интерфейс OpenGL и связанный с ним язык шейдеров OpenGL Shading Language (GLSL).

2. ТРАССИРОВКА ЛУЧЕЙ НА GPU

Каждая реализация трассировки лучей характеризуется набором *базовых геометрических примитивов*, поддержкой *статических* или *динамических* компьютерных сцен, а также применяемой *ускоряющей структурой*.

В данной работе предполагается, что все объекты сцены представлены набором *треугольников*, что увеличивает эффективность [8]. Кроме того, рассматриваемая реализация предназначена для визуализации сцен со *статической* геометрией, однако источники света и наблюдатель могут произвольно менять свое положение в пространстве во время визуализации.

Наряду с традиционной *регулярной сеткой* в данной работе использовалась регулярная сетка с информацией о *близости* (*proximity cloud*), позволяющая повысить эффективность поиска точек соударения [4]. Для этого всем пустым вокселям сетки приписывается расстояние до *ближайшего* вокселя, содержащего объекты сцены.

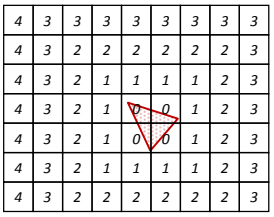


Рисунок 1. Пример регулярной сетки с информацией о близости *непустых* вокселей

Данные значения могут быть использованы во время прохода сетки для быстрой обработки больших пустых областей.

2.1 Алгоритмы ускоряющих структур

2.1.1 Пересечение треугольника и вокселя

В работе [9] предложен следующий эффективный алгоритм, основу которого составляет теорема “*О разделяющих осях*” двух *выпуклых* многогранников A и B. В качестве многогранника A выступает параллелепипед P, стороны которого параллельны осям системы координат, однозначно определяемый положением центра $\vec{c}$ и вектором половинных размеров $\vec{h}$. В качестве B - треугольник T, заданный своими вершинами $\vec{u}_0$, $\vec{u}_1$ и $\vec{u}_2$. Для упрощения теста вершины треугольника преобразуются в *локальную* систему координат параллелепипеда: $\vec{v}_i = \vec{u}_i - \vec{c}$, $i \in \{0, 1, 2\}$. Согласно теореме, существует 13 осей, вдоль которых параллелепипед P и треугольник T могут быть разделены:

- [*3 места*] $\vec{e_0} = (1, 0, 0)$, $\vec{e_1} = (0, 1, 0)$, $\vec{e_2} = (0, 0, 1)$ – нормали к граням параллелепипеда (тест на пересечение параллелепипеда P с *минимальным* ограничивающим параллелепипедом треугольника T).
- [*1 тест*] $\vec{n}$ – нормаль к треугольнику T (тест на пересечение плоскости T с параллелепипедом P).
- [*9 тестов*] $\vec{a}_{ij} = \vec{e_i} \times \vec{f_j}, i, j \in \{0, 1, 2\}$, где $\vec{f_0} = \vec{v}_1 - \vec{v}_0$, $\vec{f_1} = \vec{v}_2 - \vec{v}_1$, $\vec{f_2} = \vec{v}_0 - \vec{v}_2$ – ребра преобразованного треугольника T (ниже рассмотрен случай $i = j = 0$).

Если все перечисленные выше тесты пройдены, то нет ни одной разделяющей оси, и P *пересекается* с T.

Для теста из второго пункта достаточно рассмотреть *две* вершины, лежащие на концах диагонали, направление которой наиболее близко к нормали $\vec{n}$:

```
vec3 n = cross ( f0, f1 );
float d = -dot ( n, v0 );
vec3 p0 = -sign ( n ) · h;
if ( dot ( n, p0 ) < d ) {
    vec3 p1 = sign ( n ) · h;
    if ( dot ( n, p1 ) > d ) return “пересечение есть”;
}
return “пересечения нет”;
```

Рассмотрим один из девяти тестов из третьего пункта при $i = j = 0$. В этом случае будем иметь $\vec{a}_{00} = \vec{e}_0 \times \vec{f}_0 = (0, -\vec{f}_{0z}, \vec{f}_{0y})$. Проекцией треугольника T на потенциальную разделяющую ось (для краткости обозначим ее символом $\vec{a}$) является отрезок с концами в точках $min(p_0, p_1, p_2)$ и $max(p_0, p_1, p_2)$:

$p_0 = \vec{a} \cdot \vec{v}_0 = v_{0z} \cdot v_{1y} - v_{0y} \cdot v_{1z}$,

$p_1 = \vec{a} \cdot \vec{v}_1 = v_{0z} \cdot v_{1y} - v_{0y} \cdot v_{1z} = p_0$,

$p_2 = \vec{a} \cdot \vec{v}_2 = (v_{1y} - v_{0y}) \cdot v_{2z} - (v_{1z} - v_{0z}) \cdot v_{2y}$.

Однако в рассматриваемом случае $p_0 = p_1$, что позволяет определять только $min(p_0, p_2)$ и $max(p_0, p_2)$, используя аппаратные функции *min* и *max*. Затем вычислим "радиус" r проекции параллелепипеда P на ось $\vec{a}$, учитывая, что $a_x = 0$:

$$r = h_x \cdot |a_x| + h_y \cdot |a_y| + h_z \cdot |a_z| = h_y \cdot |a_y| + h_z \cdot |a_z|.$$

Если все необходимые величины вычислены, тест сводится к проверке следующих двух условий:

```
if ( min ( p0, p1 ) > r or max ( p0, p1 ) < -r ) return "пересечения нет";
```

2.1.2 Формирование регулярной сетки

Построение регулярной сетки предполагает задание ограничивающего параллелепипеда компьютерной сцены, который определяется своей *минимальной* и *максимальной* точкой B_{min} и B_{max} соответственно. Для построения равномерной сетки с числом разбиений m, n, k вдоль осей x, y и z соответственно можно использовать следующий эффективный алгоритм:

```
foreach ( Triangle T from Triangles ) {
    vec3 Tmin = vec3 ( min ( T.u0.x, T.u1.x, T.u2.x ),
                       min ( T.u0.y, T.u1.y, T.u2.y ),
                       min ( T.u0.z, T.u1.z, T.u2.z ) );
    vec3 Tmax = vec3 ( max ( T.u0.x, T.u1.x, T.u2.x ),
                       max ( T.u0.y, T.u1.y, T.u2.y ),
                       max ( T.u0.z, T.u1.z, T.u2.z ) );
    vec3 start = floor ( ( Tmin - Bmin ) / VoxelSize );
    vec3 final = ceil ( ( Tmax - Bmin ) / VoxelSize );
    start = clamp ( start, vec3 (1, 1, 1), vec3 (m, n, k) );
    final = clamp ( final, vec3 (1, 1, 1), vec3 (m, n, k) );
    for i, j, k ∈ [start, final]
        if ( T ∩ Voxel[i, j, k] ≠ ∅ )
            Voxel [i, j, k].Add ( T );
}
```

Для каждого треугольника определяется его *минимальный* ограничивающий параллелепипед с концевыми вершинами T_{min} и T_{max}. Данный параллелепипед используется для определения номера начального (*start*) и конечного (*final*) вокселя из окрестности треугольника, включающей только те воксели сетки, с которыми указанный треугольник может пересекаться. Далее треугольник тестируется на пересечение с каждым вокселем данной окрестности и, если пересечение имеет место, добавляется в список объектов вокселя.

2.1.3 Обход регулярной сетки

В работе [10] дается эффективный алгоритм обхода регулярной сетки, который включает себя два этапа: *инициализация* (*initialization*) и *пошаговый обход* (*incremental traversal*).

Инициализация. Номер начального вокселя на пути луча заносится в целочисленный вектор *voxel*:

```
ivec3 voxel = floor ( ( ray.Origin - Bmin ) / VoxelSize );
```

Вектор *step* содержит знак приращения номера вокселя при прохождении луча через его границу вдоль осей x, y и z соответственно:

```
ivec3 step = sign ( ray.Direction );
```

Время прохождения лучом одного вокселя вдоль осей x, y и z записывается в вектор *delta*:

```
vec3 delta = VoxelSize / abs ( ray.Direction );
```

Наконец, вычисляется время соударения луча с ближайшей гранью вокселя вдоль осей x, y и z:

```
vec3 out = delta * max ( step, 0 ) -
           mod ( ray.Origin - Bmin, VoxelSize ) / ray.Direction;
```

Наименьшее из трех найденных значений времени t определяет расстояние, которое следует пройти вдоль луча до соударения с границей вокселя.

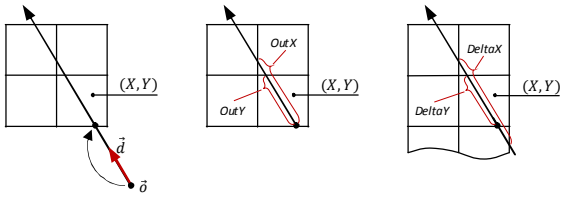


Рисунок 2. Инициализация регулярной сетки

Пошаговый проход. Этап пошагового прохода регулярной сетки выполняется весьма просто: в качестве следующего вокселя всегда выбирается *ближайший*:

```
vec3 next;
float min;
do {
    next = (0, 0, 1);
    min = out.z;
    if ( out.x < out.y )
        if ( out.x < out.z ) {
            next = (1, 0, 0);
            min = out.x;
        }
    else
        if ( out.y < out.z ) {
            next = (0, 1, 0);
            min = out.y;
        }
    Проверка всех треугольников текущего вокселя
    out += delta * next;
    voxel += step * next;
} while ( min < final );
```

Следует обратить внимание на важную особенность алгоритма: ближайшая точка соударения с треугольниками рассматриваемого вокселя может находиться *вне* данного вокселя. Предложенная программная реализация алгоритмов в значительной мере векторизована, что может дать существенный выигрыш в производительности как для графических, так и для центральных процессоров.

2.1.4 Формирование регулярной сетки близости

Построение регулярной сетки с информацией о близости отличается от формирования обычной регулярной сетки одним дополнительным шагом – построением *карты расстояний*. Для решения данной задачи был использован эффективный алгоритм [13].

2.1.5 Проход регулярной сетки близости

Отличие состоит в том, что в качестве следующего выбирается воксель, который находится на расстоянии $max(D, 1)$ от текущего вдоль направления распространения луча, где D – значение соответствующего элемента карты расстояний:

```
vec3 next;
float min;
do {
    ...
    Проверка всех треугольников текущего вокселя
    int proximity = Proximity ( voxel );
    for ( int i = 0; i < proximity - 1; i++ ) {
```

```
        if ( out.x < out.y )
            if ( out.x < out.z )
                next += (1, 0, 0);
            else
                next += (0, 0, 1);
        else
            if ( out.y < out.z )
                next += (0, 1, 0);
            else
                next += (0, 0, 1);
        out = delta * next;
    }
    voxel += step * next;
} while ( min < final );
```

2.2 Алгоритмы пересечения

2.2.1 Пересечение луча с параллелепипедом

Алгоритм используется для вычисления *начального* и *конечного* времени соударения луча с ограничивающим параллелепипедом сцены (`start` и `final` соответственно). В работе [11] дается эффективный алгоритм на основе "*полос*" ("*slabs*"), где под полосой подразумевается область пространства, заключенная между двумя плоскостями. Если начальное время соударения с наиболее удаленной полосой окажется больше конечного времени соударения с ближайшей полосой (`start` > `final`), то луч *не пересекается* с параллелепипедом:

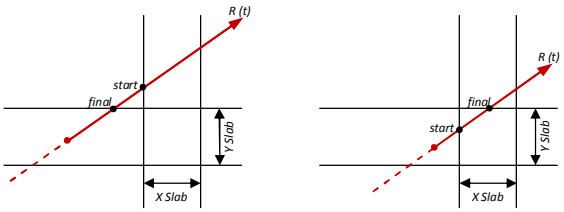


Рисунок 3. Варианты расположения луча и параллелепипеда

Программная реализация алгоритма была оптимизирована за счет использования свойств стандарта IEEE 754 и векторных возможностей современных центральных и графических процессоров:

```
vec3 omin = ( Bmin - ray.Origin ) / ray.Direction;
vec3 omax = ( Bmax - ray.Origin ) / ray.Direction;
vec3 tmax = max ( omax, omin );
vec3 tmin = min ( omax, omin );
final = min ( tmax.x, min ( tmax.y, tmax.z ) );
start = max ( tmin.x, max ( tmin.y, tmin.z ) );
return start < final and final > 0;
```

Приведенный выше псевдокод выполняет *12* элементарных арифметических операций и *12* операций сравнения и не содержит *явных* ветвлений, что повышает эффективность при исполнении на современной графической аппаратуре.

2.2.2 Пересечение луча с треугольником

В работе [12] предложен наиболее эффективный алгоритм, основанный на использовании *барицентрических координат* (α, β), в которых любую точку M треугольника $T = \{\vec{u}_0\vec{u}_1\vec{u}_2\}$ можно представить в следующем виде:

$$M(\alpha,\beta) = (1-\alpha-\beta)\cdot\vec{u}_0 + \alpha\cdot\vec{u}_1 + \beta\cdot\vec{u}_2,$$

$\alpha \geq 0$, $\beta \geq 0$ и $\alpha + \beta \leq 1$. Данные координаты активно используются на дальнейших стадиях визуализации для интерполяции нормалей, текстурных координат и любых других вершинных атрибутов.

Вычисление точки соударения луча $R(t) = \vec{o} + \vec{d}\cdot t$ с треугольником T эквивалентно решению уравнения $R(t) = M(\alpha,\beta)$, которое можно записать в виде:

$$\vec{o} + \vec{d}\cdot t = (1-\alpha-\beta)\cdot\vec{u}_0 + \alpha\cdot\vec{u}_1 + \beta\cdot\vec{u}_2.$$

Опуская промежуточные выкладки, приведем псевдокод варианта алгоритма, оптимизированного предварительным вычислением нормали к треугольнику $\vec{n}$:

```
vec3 e1 = u0 - u1;
vec3 e2 = u2 - u0;
vec3 r = u0 - o;
vec3 s = cross ( r, d );
res = vec3 ( dot ( r, n ), dot ( e2, s ), dot ( e1, s ) ) / dot ( d, n );
return res.x > 0 and res.y ≥ 0 and res.z ≥ 0 and res.y + res.z ≤ 1;
```

Предварительный расчет нормалей позволяет исключить одну операцию векторного произведения, и трудоемкость модифицированного алгоритма составляет *42* элементарных арифметических операции и *4* операции сравнения.

2.3 Формат данных и схема вычислений

Для расчета освещенности принята модель Уиттеда. Глубина трассировки является параметром, но по умолчанию для повышения производительности установлена трассировка *одного* отражения и *двух* преломлений луча. Кроме того, для повышения реалистичности изображения визуализируются тени посредством трассировки для каждого источника света одного *теневого* луча.

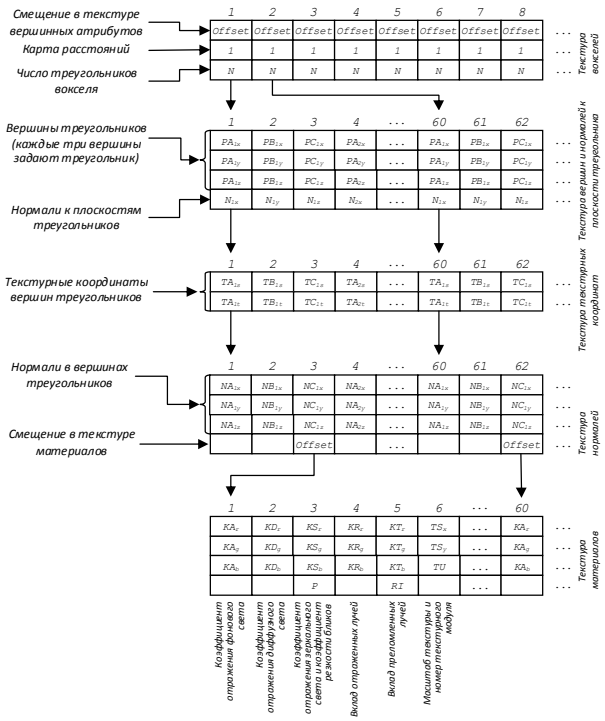


Рисунок 4. Текстуры, используемые для передачи данных

В соответствии с принципами вычислений на графическом процессоре все необходимые для расчетов данные должны быть переданы посредством текстур и глобальных `uniform`-переменных. В задаче трассировки лучей текстуры используются для передачи в шейдер модели компьютерной сцены, представленной в виде ускоряющей структуры. Первичной является *трехмерная текстура вокселей* равномерной сетки. В каждом текселе текстуры вокселей хранится смещение в *текстуре вершин*, *число треугольников*, перекрывающихся с данным вокселем, *радиус* окрестности пустых вокселей и т.д. Глобальные `uniform`-переменные используются для передачи положения и ориентации камеры и источников света.

Для инициирования вычислений следует установить параллельную проекцию и нарисовать прямоугольник, заполняющий всю область видимости. На этапе растеризации данный прямоугольник будет разбит на отдельные фрагменты, соответствующие пикселям в буфере кадра, а каждый фрагмент передан для обработки фрагментному шейдеру.

3. ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ

Для оценки производительности программы трассировки лучей использовалась тестовая сцена “Lexus”.

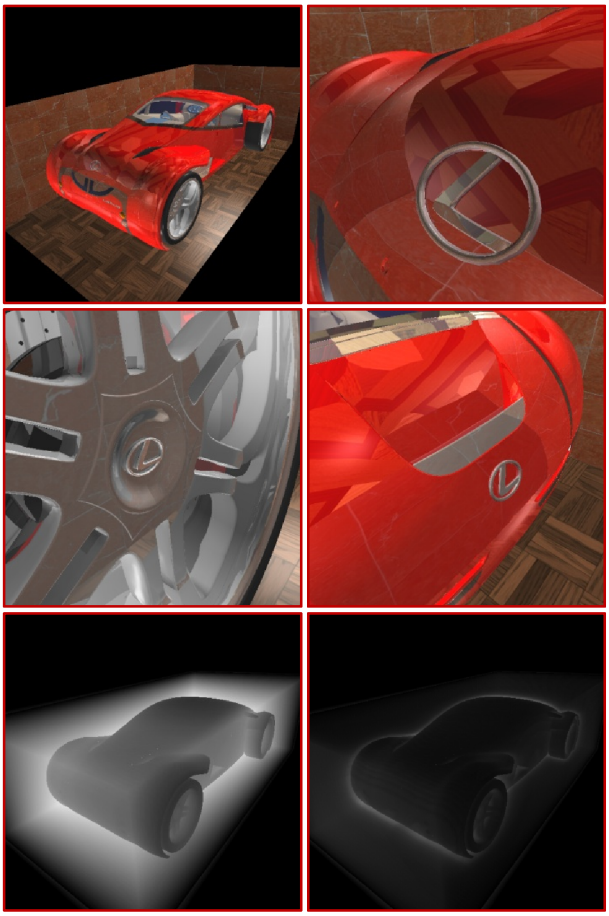


Рисунок 5. Тестовая сцена для визуализации

Модель состоит из ~325 000 треугольников, большинство материалов являются прозрачными и отражающими, установлено *два* точечных источника света, размер окна визуализации составляет 512 × 512 точек. Ниже в таблице даны замеры производительности (*кадр/сек*), соответствующие положению камеры для первого кадра (один из тяжелых случаев).

	Uniform Grid		*Proximity cloud*	
Размерность сетки	*AMD Radeon 4850*	*NVIDIA Quadro 5600*	*AMD Radeon 4850*	*NVIDIA Quadro 5600*
32 × 32 × 32	*12 / 7*	*21 / 11*	*11 / 7*	*20 / 11*
64 × 64 × 64	*22 / 12*	*37 / 20*	*29 / 16*	*51 / 26*
128 × 128 × 128	*25 / 14*	*42 / 22*	*34 / 20*	*68 / 35*

Первое значение соответствует расчету только *прямого* освещения, второе – расчету *прямого* и *вторичного* освещения. Два нижних кадра показывают оттенками серого цвета число обработанных вокселей сетки при отсутствии и наличии информации о близости.

4. ЗАКЛЮЧЕНИЕ

В данной работе построена эффективная реализация метода обратной трассировки лучей для исполнения на графическом процессоре в реальном времени. Данная реализация аккумулирует и развивает существенную часть опыта разработки трассировок реального времени (Real Time Ray Tracing или RTRT). При рядовых аппаратных возможностях реальное время обеспечено для достаточно сложных сцен (из 300К и более треугольников). Эксперимент позволяет сделать вывод о высокой эффективности: ускоряющей структуры на основе регулярной сетки с “облаком близости”; улучшенных алгоритмов пересечения “луч-примитив” без динамических ветвлений; других использованных и оптимизированных в данной работе техник. Подход на основе OpenGL и GLSL позволил задействовать все возможности современных графических процессоров и, одновременно, обеспечил межплатформенность.

5. ЛИТЕРАТУРА

[1] I. Wald, T. Purcell, J. Schmittler, C. Benthin, P. Slusallek. “*Real-time Ray Tracing and its use for Interactive Global Illumination*”/ Eurographics State of the Art Reports (2003).

[2] S. Woop, J. Schmittler, P. Slusallek. “*RPU: A Programmable Ray Processing Unit for Real-time Ray Tracing*”/ Proc. of SIGGRAPH (2005).

[3] T. Purcell. *“Ray Tracing on a Stream Processor”* (2004). (http://graphics.stanford.edu/papers/tpurcell_thesis/tpurcell_thesis.pdf)

[4] F. Karlsson, C. Ljungstedt. “*Ray tracing fully implemented on programmable graphics hardware*” (2004). (http://www.ce.chalmers.se/~uffe/xjobb/GPURT.pdf)

[5] M. Christen. “*Ray Tracing on GPU*” (2005). (http://gpurt.sourceforge.net/DA07_0405_Ray_Tracing_on_GPU-1.0.5.pdf)

[6] A. Adinetz, S. Berezin. “*Implementing Classical Ray Tracing on GPU - a Case Study of GPU Programming*”/ Proceedings of Graphicon (2006).

[7] В. Фролов. Проект “*CUDA Engine of Ray Force*”. (http://ray-tracing.ru/articles163.html)

[8] I. Wald, P. Slusallek, C. Benthin, and M. Wagner. “*Interactive Rendering with Coherent Ray Tracing*”// Computer Graphics Forum, 20(3), 153-164, 2001.

[9] T. Akenine-Moller. “*Fast 3D triangle-box overlap testing*”// Journal of Graphics Tools, 6(1), 29-33, 2001.

[10] J. Amantides, A. Woo. “*A Fast Voxel Traversal Algorithm for Ray Tracing*”. In proceedings of Eurographics’87, 3-10, New York, 1987.

[11] T. Kay, J. Kajiya. “*Ray tracing complex scenes*”. Computer Graphics, 20(4), 269–278, 1986.

[12] T. Akenine-Moller, B. Trumbore. “*Fast, Minimum Storage Ray/Triangle Intersection*”. Journal of Graphics Tools, 2(1), 21-28, 1997.

[13] T. Saito, J. Toriwaki, “New algorithms for n-dimensional Euclidean distance transformation”, Pattern Recognition, 27-11 (1994), 1551-1565.

Оптимизация представления карт освещенности и яркости для их интерактивной визуализации.

Б.Х. Барладян, А.Г. Волобой, Л.З. Шапиро
Институт прикладной математики им. М.В.Келдыша РАН, Москва

Аннотация

В статье рассматриваются различные способы представления карт освещенности и яркости, являющиеся результатом работы программ оптического моделирования. Предложен способ экспорта карт освещенности в VRML формат с упаковкой текстур. Благодаря этому экспорту возможно существенное снижение количества визуализируемых треугольников и памяти, используемой под текстуры, что позволяет добиться интерактивной скорости визуализации больших моделей практически при сохранении качества изображения.

Ключевые слова: *реалистичные изображения, карта освещенности, интерактивная визуализация, атласы, VRML.*

1. ВВЕДЕНИЕ

Получение карт освещенности и яркости в современных программах оптического моделирования, обычно, занимает значительное время. Это время особенно существенно, если речь идет о моделировании поверхностей со сложными оптическими свойствами, описываемыми двунаправленными функциями отражения и преломления, в условиях непрямого освещения. В то же время полученные результаты моделирования освещения хотелось бы визуализировать в интерактивном режиме, предоставить возможность пользователю передвигаться по сцене. Для этого карты освещенности должны быть загружаемы в независимые системы визуализации, основанные на использовании графических ускорителей, как например [1]. Другим актуальным направлением интерактивной визуализации является трассировка лучей в режиме реального времени [2, 3, 4, 5]. Возможность использовать рассчитанные заранее результаты моделирования освещенности существенно повышает качество и реалистичность визуализации. В связи с этим возникает задача экспорта полученных результатов моделирования, т.е. карт освещенности и яркости, в системы отображения результатов в достаточно эффективной форме.

В нашей системе оптического моделирования [6] вторичная освещенность вычисляется на треугольной сетке в виде значений освещенности в вершинах треугольников. Для получения точного решения используется достаточно мелкая, адаптивно создаваемая сетка. Дополнительное разбиение такой сетки производится в областях, где имеет место большой градиент освещенности. Таким образом, исходные результаты моделирования представляются в виде значений освещенностей в вершинах треугольников на детальной треугольной сетке. Следует также учитывать, что для получения фотореалистичных изображений в системах архитектурной или промышленной визуализации широко используются текстуры, модулирующие диффузное отражение поверхностей. Яркость результирующего изображения в данной точке должна вычисляться как произведение освещенности видимой точки поверхности на коэффициент диффузного рассеяния в ней. Освещенность в точке поверхности вычисляется путем интерполяции освещенности, сохраненной в вершинах треугольника, а значение коэффициента диффузного рассеяния в точке задается материалом поверхности и может модулироваться текстурой.

Одним из популярных, широко используемых в Интернете, и стандартизированных форматов представления трехмерных сцен является **VRML** (**Virtual Reality Modeling Language**), для которого существует большое количество бесплатных и коммерческих браузеров. Поэтому разработка экспорта рассчитанных карт яркости в VRML формат позволяет существенно расширить возможности визуализации результатов моделирования освещенности. VRML формат позволяет формировать изображения трехмерных сцен как с использованием цвета, определенного в вершинах треугольников, так и с использованием текстур. Однако он не позволяет использовать оба подхода одновременно, т.е. получать изображения в виде произведения текстуры на интерполированное значение цвета (освещенности). Некоторые браузеры поддерживают такое расширение, но в общем случае на это нельзя рассчитывать. Таким образом, учитывая, что в большинстве сцен, используемых в архитектуре и промышленности, широко применяются текстуры, единственным практическим подходом является представление карт освещенности и яркости с помощью текстур для всей поверхности сцены. Представление карт освещенности яркости с помощью текстур также позволяет уменьшить размерность треугольной сетки, экспортируемой в VRML. Поскольку генерируемое изображение теперь зависит в основном не от треугольной сетки, а от текстуры, то можно использовать существенно более грубую треугольную сетку.

2. ОСНОВНЫЕ ПРОБЛЕМЫ И ПУТИ ИХ РЕШЕНИЯ

Основной проблемой использования текстур для представления карт освещенности и яркости в **VRML** формате является проблема размерности. Графические ускорители, используемые всеми браузерами, поддерживающими **VRML** формат, имеют ограничения по объему используемых текстур. При превышении этого объема часть текстур перестает воспроизводиться, либо скорость воспроизведения существенно падает. Следует учесть, что при использовании обычных текстур в представлении сцены широко применяется их повторяемость. Небольшое по размеру изображение, например, поверхность дерева, обоев или, даже часть фасада дома повторяется много раз для представления большой поверхности – стены или здания. Использование этой повторяемости для

представления карт освещенности и яркости становится невозможным, поскольку одинаковые (повторяющиеся) части геометрии, как правило, имеют разную освещенность. В силу этих причин размерность используемых текстур возрастает многократно.

Другой причиной, требующей высокого разрешения результирующих текстур, является использование в сцене обычных текстур высокого разрешения. Результирующая текстура объединяет карту освещенности и обычную текстуру, поэтому ее разрешение должно быть достаточным, чтобы без искажений передать рисунок изначальной обычной текстуры.

И, наконец, третьей причиной, требующей высокого разрешения результирующих текстур, является наличие в сцене участков с большим градиентом яркости. Изображение на этих участках будет смазано, если разрешение текстуры будет недостаточно. На рис.1 представлено изображение двери в салоне самолета при недостаточном разрешении текстуры. Проблема вызвана тенью на двери, которая при недостаточном разрешении текстуры размывается и происходит так называемая ”утечка света” (тени в данном случае). На рис.2 представлено изображение двери со снятым наличником, где хорошо видна исходная тень, размытая на рис.1. На рис.3 представлено изображение той же двери в самолете, полученной при существенном увеличении разрешения текстуры. Проблема с ”утечкой света” практически исчезла.

Рис. 1. Изображение двери в самолете при недостаточном разрешении текстуры.

Рис. 2. Изображение двери в самолете со снятым наличником.

Рис. 3. Изображение двери в самолете при увеличенном разрешении текстуры.

Этот пример иллюстрирует основную проблему построения текстур для представления карт яркости в **VRML** формате. Высокое качество изображений требует использования текстур высокого разрешения, в то время как максимальный объем текстур, который может использоваться графическими ускорителями, ограничен.

Следующей проблемой является ограничение не только полного объема текстур, но также их количества и максимальной размерности отдельных текстур. Эти ограничения зависят от используемого графического ускорителя и драйвера. Типичные значения, при которых графический ускоритель может использоваться эффективно, – это несколько десятков текстур при разрешении отдельной текстуры не более 4096x4096 точек. Была реализована возможность контролировать эти параметры в процессе экспорта карт освещенности. Таким образом, пользователь может найти оптимальное соотношение между качеством итогового изображения и эффективностью его визуализации для достижения интерактивной скорости.

Следует отметить, что при подготовке текстур приходится разбивать всю геометрию сцены на компоненты – небольшие части, для которых можно определить такое отображение поверхности на плоскость текстуры, что искажения будут находиться в заданных пределах. Наиболее практическим и достаточно эффективным подходом для решения этой задачи является разбиение всей геометрии сцены на группы связанных между собой треугольников, нормали которых отличаются от выбранного направления не больше чем на заданную величину. Для каждой такой группы строится отдельная текстура. В результате этого шага получается очень большое количество текстур, необходимых для полного покрытия сцены. Для реальных сцен это число может составить тысячи или даже десятки тысяч. Многие из этих текстур могут иметь небольшие размеры, поэтому наиболее разумным решением является упаковка некоторого множества таких текстур в одну – так называемый Атлас. Микрософт DirectX SDK [7] предоставляет соответствующие процедуры, позволяющие эффективно упаковывать множество текстур в один такой атлас. Как правило, разрешение атласа по каждому измерению выбирается кратным степени двух (2^n), что позволяет избежать дополнительных расходов памяти при передаче текстур в графический ускоритель. Полное количество атласов и их конкретное разрешение выбирается пользователем в зависимости от требования к качеству отображения карт освещенности и возможностей оборудования, в первую очередь графического ускорителя. Задача программы, экспортирующей результаты моделирования освещенности в VRML формат, состоит в обеспечении максимально возможного качества экспортируемых карт освещенности при заданных ограничениях – количестве и разрешении атласов.

3. ОПТИМИЗАЦИЯ КАЧЕСТВА ТЕКСТУРЫ ПРИ ЗАДАННЫХ ОГРАНИЧЕНИЯХ

Оптимизация качества экспортируемых карт освещенности происходит в несколько этапов.

На первом этапе вся геометрия сцены (треугольная сетка) разбивается на компоненты так, чтобы для каждой из них

можно было определить отображение поверхности на плоскость текстуры, и искажения лежали бы в заданных пределах. В нашей системе используется разбиение на группы связанных между собой треугольников, поскольку текстура, представляющая карту освещенности, должна быть разрезана вдоль границ связанной компоненты. Здесь возможны скачки яркости. Для уменьшения искажения отображения в группу собираются треугольники, нормали у которых отличаются от выбранного направления не больше чем на заданную величину.

На втором этапе для каждой такой группы строится карта яркости – текстура заданного разрешения в RGB пространстве. Для построения текстуры может использоваться как линейный, так и не линейный операторы сжатия динамического диапазона яркостей [8]. Исходное разрешение текстуры выбирается достаточно высоким (~1-0.1см в физических единицах на реальных поверхностях сцены) так, чтобы возможно было получить изображения высокого качества. Если для данной группы треугольников в сцене назначена обычная текстура, то разрешение результирующей текстуры выбирается не ниже разрешения обычной текстуры, чтобы избежать ее размывания.

На третьем этапе происходит сжатие построенных текстур до минимально допустимого значения. Вычисляется среднее значение яркости в RGB пространстве для всех последовательных четверок текстурных пикселов. Разрешение текстуры можно уменьшить в четыре раза (в два раза по каждому измерению), если для каждой четверки это среднее значение отличается от значения каждого пиксела из четверки меньше заданного пользователем порога. Процедура повторяется столько раз, сколько это допустимо. Использование такой процедуры сжатия существенно сокращает необходимый объем памяти потому, что в реальных сценах всегда существуют не освещенные поверхности (внутренние поверхности реальных объектов). Для них текстуры можно сжать до минимального (16x16) размера.

На четвертом этапе построенные текстуры упаковываются в атласы с использованием DirectX [7]. Для этого множество построенных текстур необходимо разбить на группы, каждая из которых будет упакована в отдельный атлас. Количество групп должно быть меньше или равно максимальному числу атласов, заданному пользователем. В простейших случаях мы можем просто собрать в одну группу текстуры с суммарным разрешением примерно равным максимальному разрешению атласа. Но в действительности это возможно только для небольших и простых сцен. Для больших реальных сцен суммарный объем (разрешение) текстур, построенных на третьем этапе, существенно превосходит суммарное разрешение максимально допустимого количества атласов. Поэтому в реальности приходится использовать меньшее разрешение текстур для того, чтобы вписаться в заданные ограничения по размерности. Желательно при этом уменьшить разрешение всех текстур примерно одинаково, чтобы их качество было бы равномерно по всей сцене. Для этого желательно разбить исходное множество текстур на группы с приблизительно одинаковым суммарным разрешением, каждая из которых затем будет упакована в отдельный атлас. Для этой цели используется следующий алгоритм:

(1) Все текстуры, построенные на этапе 3, сортируются в порядке убывания их размерности (количеству пикселов).

(2) Выбирается некоторая величина **A,** определяющая максимальную суммарную размерность текстур, которые можно собрать в одну группу. Максимальное разрешение атласа используется в качестве начального приближения для этой величины.

(3) Мы добавляем текстуры в заданную группу до тех пор, пока их суммарное разрешение не превысит заданную величину **A.** Группа начинает формироваться с наибольших текстур. Если данная текстура не может быть добавлена в формируемую группу, то мы пытаемся добавить следующую, меньшую по размеру. Как результат этой процедуры некоторое количество групп **Na** будет получено.

(4) Процедура, описанная в п. 1-3, фактически определяет некоторую функцию **Na = F(A),** которая задает результирующее количество групп (атласов) для максимального суммарного разрешения группы **A**. Максимальное количество атласов задано пользователем, поэтому для вычисления максимального суммарного разрешения группы **A** необходимо решить соответствующее нелинейное уравнение **F(A) = N,** где **N** – максимальное количество атласов, заданное пользователем. Решение находится методом деления отрезка пополам, поскольку функция **F(A)** не возрастающая ($\mathbf{F(A_1) \le F(A_2)}$ при $\mathbf{A_1 > A_2}$).

При решении этого уравнения определяется состав групп исходных текстур для каждого из будующих атласов.

(5) На пятом этапе каждая из определенных выше групп текстур упаковывается в отдельный атлас. При упаковке определяются текстурные координаты внутри данного атласа для всех треугольников, которые к нему отнесены. После этого создается результирующая текстура для данноto атласа, т.е. текстура, представляющая карту освещенности или яркости для той части поверхности сцены (тех треугольников), которая отнесена к данному атласу. Текстура строится путем растеризации карты освещенности для каждого из треугольников на текстуре атласа. При этом в каждом пикселе текстуры суммируются значения освещенностей (либо яркостей) от всех треугольников повехности сцены, чьи проекций на текстурную плоскость его пересекают. Процедура повторяется несколько раз со сдвигом текстурных координат вершин растеризуемых треугольников для сглаживания результирующей текстуры атласа.

4. РЕЗУЛЬТАТЫ

Предложенная схема построения карт освещенности и яркости с упаковкой их в атласы была опробована на сценах размерности 1-3 млн. треугольников. Количество исходных текстур (компонент) составляло ~10000. Количество создаваемых атласов составляло ~50 при разрешении каждого 2048x2048.

На приведенных рисунках представлен салон самолета Boeing. Практически все освещение салона – непрямое. На рис. 4 показана изначально рассчитанная карта освещенности, сохраненная в вершинах треугольников. На рис. 5 представлена визуализация полученного в результате

экспорта VRML файла с упакованными в атласы текстурами. Для визуализации использовался VRML браузер Cortona [1].

Из сравнения рисунков видно, что качество представления освещенности практически не ухудшилось. В то же время полученный VRML файл может быть визуализирован в интерактивном режиме, и пользователь может путешествовать по салону самолета в реальном времени.

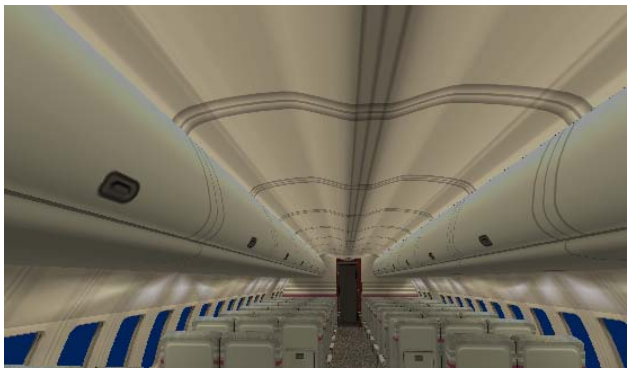


Рис. 4. Изначальная карта освещенности салона самолета.

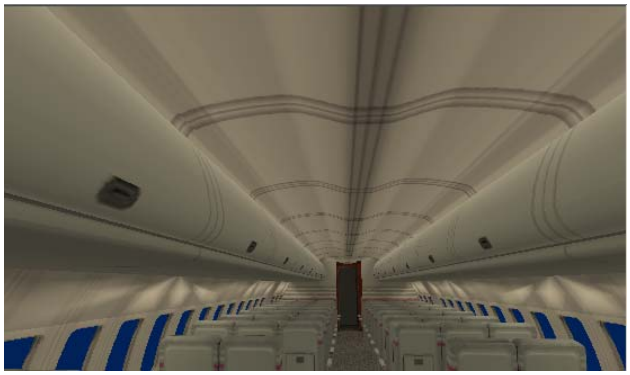


Рис. 5. Та же карта освещенности, экспортированная в VRML формат с упакованными текстурами.

Работа поддержана грантом РФФИ № 09-01-00472, а также фирмой Integra Inc. (Япония).

Электронный вариант статьи с цветными иллюстрациями размещен на сайте http://www.keldysh.ru/pages/cgraph/publications/cgd_publ.htm.

5. ЛИТЕРАТУРА

[1]http://www.parallelgraphics.com/products/cortona/

[2] Andreas Dietrich, Ingo Wald, Holger Schmidt, Kristian Sons, and Philipp Slusallek. Realtime Ray Tracing for Advanced Visualization in the Aerospace Industry Proceedings of the 5th Paderborner Workshop Augmented and Virtual Reality in der Produktentstehung, 2006 (http://graphics.cs.uni-sb.de/~dietrich/)

[3] Ingo Wald and Philipp Slusallek. State-of-the-Art in Interactive Ray-Tracing in State of the Art Reports, EUROGRAPHICS-2001, Manchester, United Kingdom, September 3-7, 2001, pp. 21-42.

[4] Ingo Wald, Philipp Slusallek, and Carsten Benthin. Interactive distributed ray tracing of highly complex models. In Proceedings of the 12th EUROGRPAHICS Workshop on Rendering, June 2001, London, pp. 274 -285.

[5] Ingo Wald, William R. Mark, Johannes Günther, Solomon Boulos, Thiago Ize, Warren Hunt, Steven G. Parker, Peter Shirley. State of the Art in Ray Tracing Animated Scenes in State of the Art Reports, EUROGRAPHICS-2007, http://www.sci.utah.edu/~wald/Publications/2007///Star07/download//star07rt.pdf

[6] А.Г. Волобой, В.А. Галактионов, Машинная графика в задачах автоматизированного проектирования. «Информационные технологии в проектировании и производстве» № 1, 2006, стр. 64-73.

[7] http://msdn2.microsoft.com/en-us/library/bb206321.aspx

[8] Б.Х. Барладян, А.Г. Волобой, В.А. Галактионов, Э.А. Копылов. Эффективный оператор сжатия динамического диапазона яркостей. "Программирование", № 5, с. 35-42, 2004.

Abstract

Luminance and illuminance representation via illumination maps (i-maps) or shaded texture is considered. The method of export of i-maps calculated by optical simulation program into VRML format with texture compressing and packing is described. Different optimization methods of quality and size of these textures are considered. VRML files can be browsed in any available browser. Due to proposed method it is possible to reduce the number of triangles and the volume of texture memory. This leads to interactive speed of rendering of huge models preserving high quality of illuminance distribution.

Keywords: *realistic images, illumination maps, interactive visualization, atlas, VRML.*

Authors:

Boris H. Barladyan, PhD, senior researcher, Keldysh Institute for Applied Mathematics RAS.

E-mail: obb@gin.keldysh.ru

Alexey G. Voloboy, PhD, senior researcher, Keldysh Institute for Applied Mathematics RAS.

E-mail: voloboy@gin.keldysh.ru

Lev Z. Shapiro, PhD, senior researcher, Keldysh Institute for Applied Mathematics RAS.

Ускорение метода световых сеток за счет использования графического процессора

Виктор Дебелов, Илья Новиков
Институт вычислительной математики и математической геофизики СО РАН,
Новосибирск, Россия
debelov@oapmg.sscc.ru, iliya.novikov@gmail.com

Аннотация

Метод световых сеток (МСС) является модификацией классической обратной рекурсивной лучевой трассировки Виттеда (ОРЛТ), использующейся для рендеринга пространственных сцен. Изображение, рассчитанное по МСС, является аппроксимацией изображения, рассчитанного алгоритмом Виттеда, при этом МСС генерирует мягкие тени. Алгоритмы теневых карт (АТК) и теневых объемов (АТО) разработаны для определения теней точечных источников света. Их основная привлекательность – быстрый расчет изображений при использовании операций z-буфер и стенсил-буфер графического процессора (ГП).

В работе предложены гибридные алгоритмы МСС-АТК и МСС-АТО, в которых часть теневых расчетов осуществляется на ГП с использованием подходов АТК или АТО, соответственно. Приводятся данные численных экспериментов, которые показывают, что ускорение МСС достигает в среднем 20-34% для МСС-АТК и 5-28% для МСС-АТО в зависимости от типа сцены.

Ключевые словаs: *реалистический рендеринг, трассировка лучей Виттеда, теневые карты, теневые объемы, метод световых сеток, мягкие тени.*

1. ВВЕДЕНИЕ

МСС является модификацией трассировки Виттеда [1], использующейся для рендеринга пространственных сцен. Первая версия МСС была введена в работах [2, 3]. В работах [4, 5] сформировался его современный вид, который рассматривается в данной работе. В работе [5] дается обширное исследование МСС и его сравнение с известными алгоритмами генерации мягких теней в рамках ОРЛТ как по скорости, так и по качеству получаемых изображений.

АТК [6] и АТО [7] – два алгоритма генерации четких теней, получившие широкое распространение благодаря аппаратной поддержке расчета изображения. Современные ГП позволяют рассчитывать изображения для сцен, состоящих из сотен тысяч и миллионов треугольников, меньше чем за секунду. С другой стороны, у этих алгоритмов много недостатков: они страдают от артефактов, не применяются для рекурсивных расчетов, не работают для полупрозрачных объектов.

В данной работе ставится и решается задача ускорения МСС за счет применения ГП, используя подходы упомянутых алгоритмов. Предполагается, что читатель в основных чертах знаком с алгоритмами ОРЛТ, АТК, АТО.

1.1 Обратная рекурсивная лучевая трассировка

Пространственная сцена – это кусочно-непрерывная поверхность, представленная набором из nO объектов или примитивов, обладающих следующими свойствами:

- В каждой точке поверхности определены свойства отражения, которые задаются коэффициентами: k_d – коэффициент диффузного отражения, k_s – коэффициент зеркального отражения, k_t – прозрачность.
- Для поверхности определена операция пересечения с лучом.
- В каждой точке поверхности P определена нормаль $\vec{n}(P)$.
- Все объекты сцены задаются в декартовой *мировой* системе координат.

Сцена освещается nL точечными источниками освещения, специфицированными интенсивностями излучения I_i и позициями в пространстве LP_i, $i = 1,...,nL$. Камера – это набор параметров, характеризующих наблюдателя: позиция и ориентация в пространстве, высота и ширина экрана.

Пусть луч из камеры имеет ближайшее пересечение с поверхностью сцены в точке P. Базовую формулу расчета значения интенсивности в точке P по ОРЛТ можно кратко записать, как (следуя [8]):

$$I_{RT}(P) = A + \sum_{i=1}^{nL} V(LP_i, P)\Omega(L_i, P) + k_s I_r + k_t I_t = A + U_{RT}(P) + k_s I_r + k_t I_t, \quad (1)$$

здесь A – интенсивность рассеянного света, $\Omega(L_i, P)$ характеризует конкретную *локальную модель освещенности*, I_r – интенсивность, пришедшая с направления отраженного вектора; I_t – интенсивность, пришедшая через поверхность из-за прозрачности. $V(LP_i, P)$ – булева функция видимости i-го точечного источника из точки P.

1.2 Алгоритм теневых карт

Буфер глубины $DB[N, M]$ – прямоугольник на плоскости изображения, совпадающий с прямоугольником изображения, разрешение буфера глубины $N \times M$ пикселей. Пиксель буфера глубины хранит расстояние от камеры до ближайшего пересечения луча из камеры, проходящего через центр этого пикселя, с объектами сцены.

В АТК для определения видимости изображаемой точки источником используются теневые карты. Теневая карта – это буфер глубины, рассчитанный для камеры, находящейся в

позиции источника освещения. Чтобы определить видимость изображаемой точки, расстояние от точки до источника сравнивается со значением глубины из теневой карты. Если значение из теневой карты меньше, источник не виден – изображаемая точка находится в тени источника. Подробное исследование АТК можно найти в работах [5, 6].

1.3 Алгоритм теневых объемов

АТО для определения видимости изображаемой точки использует дополнительные геометрические построения, называемые теневыми объемами. Теневые объемы – это полубесконечные поверхности, получаемые вытягиванием геометрии силуэтов объектов от источника на бесконечность. Если изображаемая точка попадает внутрь хотя бы одного теневого объема, значит она в тени источника.

Стенсил-буфер $SB[N,M]$ – еще один буфер на прямоугольнике изображения. Каждый пиксель стенсил-буфера можно интерпретировать как счетчик, который увеличивается, когда луч из камеры через центр этого пикселя входит в теневой объем, и уменьшается, когда луч выходит из теневого объема. Более подробно см. [5, 7].

1.4 Метод световых сеток

Световая сетка $LM = \{x_k\}_{k=1}^{N^{LM}}$ – это равномерная сетка с шагом h и размером $N^{LM} = N_x^{LM} \times N_y^{LM} \times N_z^{LM}$ в пространстве сцены, каждая точка световой сетки x_k (*световая точка*) хранит шкалу видимости данной точки пространства сцены для каждого источника. Для расчета видимости в точке сцены P строится *интерполяционное множество* $D_{LM}(P,r)$ по следующему правилу

$$D_{LM}(P,r) = \{x : x \in LM, \|P-x\| < r, \\ (\vec{n}(P), x-P) > 0, V(P,x) = 1\}. \quad (2)$$

Аналогично формуле (1) базовую формулу расчета значения интенсивности в точке P по МСС представим как

$$U_{LM}(P) = \sum_{i=1}^{nL} \left\lfloor \left(\frac{1}{m} \sum_{k=1}^{m} V(LP_i, x_k) \chi(P, L_i) \right) \Omega(L_i, P) \right\rfloor, \quad (3)$$

$$I_{LM}(P) = A + U_{LM}(P) + k_s I_r + k_t I_t$$

здесь $\chi(P,L_i) = \begin{cases} 1, (\vec{n}(P), (LP_i - P)) \geq 0 \\ 0, \text{иначе} \end{cases}$ – функция, которая говорит о том, освещает источник точку спереди (значение 1) или сзади (значение 0). m – число световых точек в интерполяционном множестве. Если $m = 0$, тогда значение интенсивности в точке P по МСС рассчитывается по формуле (1), т.е. в этом случае $U_{LM}(P) = U_{RT}(P)$.

В формуле (3) используется функция значения видимости источника световой точкой $V(LP_i, x_k)$, которая требует значительных вычислительных затрат. Для уменьшения этих затрат рассматриваются два алгоритма: 1) комплексирование МСС с АТК; 2) комплексирование МСС с АТО. Основная идея: заимствовать для МСС положительное качество АТК и АТО – быстрый тест видимости источника.

2. КОМПЛЕКСИРОВАНИЕ МСС С АТК

Рассмотрим этот алгоритм по шагам для сцены с одним источником L: интенсивность I, позиция в сцене LP.

1. Строим световую сетку $LM = \{x_k\}_{k=1}^{N^{LM}}$.
2. Строим камеру зрителя Cam. Оператор перевода координат точки сцены из системы камеры в мировую систему координат Tr_{Cam}: $P(u,v,w) \Rightarrow P(x,y,z)$, т.е. $Tr_{Cam}(u,v,w) = (x,y,z)$. Порт вывода Π_{Cam}. Рассчитываем буфер глубины DB_{Cam}.
3. Для источника L строим камеру $LCam$. Порт вывода Π_L. Оператор перевода координат Tr_L. Строим теневую карту DB_L.
4. Строим изображение сцены FB_{Cam} в камере Cam, опираясь на обе карты глубины. Если глубина изображаемой точки $w = DB_{Cam}(u,v) = -1$, полагаем $FB_{Cam}(u,v) = -1$, т.е. цвет фона. Выход.
5. Иначе вычисляем $P = (x,y,z) = Tr_{Cam}(u,v,w)$ – координаты изображаемой точки в мировой системе.
6. Для точки P строим интерполяционное множество $D_{LM}(P,r)$ по правилу (2), размер множества m.
7. Если $m > 0$, т.е. $D_{LM}(P,r) \neq \varnothing$, определяем видимость источника $V(LP, x_k)$ для каждой световой точки $x_k \in D_{LM}(P,r) = \{x_k\}_{k=1}^{m}$:
   - 7.1. Если видимость для световой точки уже известна, т.е. была вычислена ранее, переходим к следующей световой точке интерполяционного множества.
   - 7.2. Иначе вычисляем расстояние $d = \|x_k - LP\|$.
   - 7.3. Переводим координаты световой точки $x_k = (x_k{}^x, x_k{}^y, x_k{}^z)$ в систему координат порта вывода камеры источника $LCam$: $(uL, vL, wl) = Tr_L^{-1}(x_k{}^x, x_k{}^y, x_k{}^z)$.
   - 7.4. Выбираем из карты глубины камеры источника значение $d_L = DB_L(uL, vL)$.
   - 7.5. Если $d_L < d$, то источник заслонен от световой точки x_k, т.е. $V(LP, x_k) = 0$. Иначе световая точка освещена источником, т.е. $V(LP, x_k) = 1$.
8. Рассчитываем интенсивность изображаемой точки P по формуле (3) и добавляем в $FB_{Cam}(u,v)$.
9. Если $m = 0$, т.е. $D_{LM}(P,r) = \varnothing$, используем стандартный ОРЛТ: рассчитываем интенсивность точки P по формуле (1) и добавляем в $FB_{Cam}(u,v)$.

Шаги 2 и 3 выполняются на ГП. Остальные шаги выполняются на центральном процессоре.

3. КОМПЛЕКСИРОВАНИЕ МСС С АТО

По шагам алгоритм можно записать следующим образом:

1. Рассмотрим световую сетку LM размера $N_x^{LM} \times N_y^{LM} \times N_z^{LM}$, как набор прямоугольных слоев световых точек вдоль одной из координатных осей. Пусть это будет ось Z, тогда каждый слой имеет размеры $N_x^{LM} \times N_y^{LM}$.

2. Введем новый порт вывода Π_{LM}, на котором построим растр размером $N_x^{LM} \times N_y^{LM}$. Вводим буфер глубины DB_{LM} и стенсил-буфер SB_{LM} на Π_{LM}.
3. Для каждого слоя световых точек LM_z^k строим камеру $CamLM$ так, что луч из верхушки O_{LM} камеры в каждую световую точку $LM_z^k[i,j]$ проходит через центр пикселя $[i,j]$ прямоугольника Π_{LM} (рис. 1).
4. Рассчитываем буфер глубины DB_{LM} в камере $CamLM$ для сцены, состоящей из световых точек LM_z^k.
5. Строим теневой объем сцены $SV(L)$ относительно источника L.
6. В точках $(u,v) \in \Pi_{LM}$, где $DB_{LM}(u,v) = -1$ полагаем $SB_{LM}(u,v) = -1$. Иначе $SB_{LM}(u,v) = c$, где c – некоторый счетчик, вычисляемый по алгоритму:
   6.1. Находим точку $P(u,v,w)$ в системе координат камеры, где $w = DB_{LM}(u,v) \neq -1$ (не фон), переводим ее координаты в мировую систему $(x,y,z) = Tr_{LM}(u,v,w)$. Пускаем луч $\overrightarrow{O_{LM}P}$ в пространстве сцены и подсчитываем число пересечений этого луча с поверхностями из теневого объема $SV(L)$. Вначале $c = 0$.
   6.2. После нахождения очередного пересечения луча с теневым объемом в некоторой точке Q проверяем по карте глубины: если $\left\|\overrightarrow{O_{LM}Q}\right\| > DB_{LM}(u,v) = \left\|\overrightarrow{O_{LM}P}\right\|$, то переходим к поиску следующего пересечения, поскольку найденная точка не может заслонять тестируемую точку P, она дальше от камеры.
   6.3. Иначе анализируем скалярное произведение $t = \left(\overrightarrow{O_{LM}P}, \vec{n}(Q)\right)$.
   6.4. Если $t = 0$, то счетчик не изменяется. Если $t < 0$, то увеличиваем счетчик $c = c + 1$. Это означает, что луч вошел в очередной теневой объем очередного примитива.
   6.5. Если $t > 0$, то уменьшаем счетчик $c = c - 1$. Это означает, что луч вышел из очередного теневого объема очередного примитива.
7. Заполняем значения видимости для световых точек слоя LM_z^k. Если $SB_{LM}[i,j] > 0$, эта световая точка в тени источника L, т.е. $V(LP, LM_z^k[i,j]) = 0$. Иначе световая точка освещена источником, т.е. $V(LP, LM_z^k[i,j]) = 1$.
8. Выполняем шаги алгоритма 3 – 7 для каждого слоя LM_z^k, $k = 1...N_z^{LM}$. Получаем заполненную световую сетку LM.
9. Строим камеру зрителя Cam. Оператор перевода координат Tr_{Cam}. Порт вывода Π_{Cam}. Рассчитываем буфер глубины DB_{Cam}.
10. Строим изображение сцены FB_{Cam} в камере Cam. Если глубина изображаемой точки $w = DB_{Cam}(u,v) = -1$, тогда $FB_{Cam}(u,v) = -1$, т.е. цвет фона. Выход.
11. Иначе вычисляем $P = (x,y,z) = Tr_{Cam}(u,v,w)$ – координаты изображаемой точки в мировой системе.
12. Для точки P строим интерполяционное множество $D_{LM}(P,r)$ по правилу (2), размер множества m.
13. Если $m > 0$, рассчитываем интенсивность изображаемой точки P по формуле (3) и добавляем в $FB_{Cam}(u,v)$.
14. Если $m = 0$, используем стандартный ОРЛТ: рассчитываем интенсивность точки P по формуле (1) и добавляем в $FB_{Cam}(u,v)$.

Шаги 3 – 9 выполняются на ГП, остальные шаги выполняются на центральном процессоре.

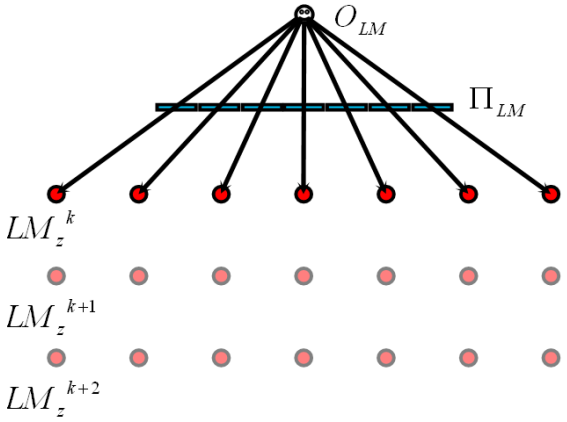


Рис. 1. Камера $CamLM$, прямоугольник Π_{LM}, слои световых точек

4. ЧИСЛЕННЫЕ ЭКСПЕРИМЕНТЫ

В табл. 1 – 3 представлены времена расчета для сцен: “Gazebo” – 300 треугольников, “Child park” – 27000 треугольников, “Garden” – 234000 треугольников. В ходе расчетов варьировались: разрешение изображения $N_{Cam} \times M_{Cam}$, шаг световой сетки h и радиус интерполяционной сферы r. Представлены времена расчета изображений по базовому МСС, гибридному алгоритму МСС-АТК, гибридному алгоритму МСС-АТО, как и обозначены соответствующие столбцы.

Для сцены “Gazebo” комплексирование МСС с АТК позволяет сократить время расчета в среднем на 20%, комплексирование МСС с АТО – на 5%.

Таблица 1. Времена расчета сцены “Gazebo” (с)

$N_{Cam} \times M_{Cam}$	h	r/h	МСС	МСС-АТК	МСС-АТО
1024 x 1024	2.0	2.1	13,2	11,0	12,3
		3.1	20,5	16,7	18,0
	1.0	2.1	18,4	11,6	19,8
		3.1	28,9	17,7	25,9
2048 x 2048	2.0	2.1	45,4	41,9	43,9
		3.1	69,9	63,9	65,5
	1.0	2.1	51,1	43,0	51,6
		3.1	79,1	66,1	74,3

Для сцены “Child park” комплексирование МСС с АТК позволяет сократить время расчета в среднем на 21%, комплексирование МСС с АТО – на 11%.

Таблица 2. Времена расчета сцены “Child park” (с)

$N_{Cam} \times M_{Cam}$	h	r/h	МСС	МСС-АТК	МСС-АТО
1024 x 1024	2.0	2.1	27,0	22,3	24,0
		3.1	34,0	26,7	28,4
	1.0	2.1	37,0	23,0	31,0
		3.1	50,8	28,0	37,3
2048 x 2048	2.0	2.1	90,6	84,4	96,6
		3.1	109,6	101,0	102,3
	1.0	2.1	101,8	87,0	94,1
		3.1	127,5	103,5	110,6

Для сцены “Garden” комплексирование МСС с АТК позволяет сократить время расчета в среднем на 34%, комплексирование МСС с АТО – на 28%.

Таблица 3. Времена расчета сцены “Garden” (с)

$N_{Cam} \times M_{Cam}$	h	r/h	МСС	МСС-АТК	МСС-АТО
1024 x 1024	2.0	2.1	24,1	15,3	16,9
		3.1	31,4	21,0	22,3
	1.0	2.1	37,2	16,4	21,3
		3.1	55,3	22,8	27,7
2048 x 2048	2.0	2.1	65,4	56,2	58,1
		3.1	89,0	76,8	77,9
	1.0	2.1	81,4	57,8	63,2
		3.1	115,0	79,2	83,3

5. ЗАКЛЮЧЕНИЕ

Гибрид МСС-АТК позволяет рассчитывать изображения в среднем на 20 – 34 % (в зависимости от типа сцены) быстрее в сравнении с базовым МСС. Для гибрида МСС-АТО среднее ускорение в сравнении с базовым МСС составляет 5 – 28 % в зависимости от типа сцены. С увеличением сложности сцены гибридные алгоритмы дают больший выигрыш по времени в сравнении с базовым МСС.

Благодарности

Работа выполнялась при финансовой поддержке РФФИ по грантам 06-07-89216_а, 08-07-12094_офи.

6. REFERENCES

[1] Whitted T. An Improved Illumination Model for Shaded Display / Commun. ACM. – 1980. – Vol. 23, № 6. – P. 343-349.

[2] Дебелов В.А., Севастьянов И.М. *Оригинальный подход к имитации мягких теней и учету диффузных переотражений в лучевой трассировке // Тр. 11-й междунар. конфер. по компьютерной графике и машинному зрению Графикон'2001. – Нижний Новгород, 2001. – С. 18–24. 21.*

[3] Debelov V.A., Sevastyanov I.M. *Soft shadows as interpolation of visibility // Future Generation Computer Systems. - 2004. - Vol. 20, № 8. - P. 1299-1315.*

[4] Дебелов В.А., Васильева Л.Ф., Новиков И.Е. *Развитие метода световых сеток для алгоритма лучевой трассировки: аппроксимация решения, реализация на графическом акселераторе // Тр. 15-й междунар. конфер. по компьютерной графике и ее приложениям Графикон'2005. – Новосибирск, 2005. – С. 355–359.*

[5] Дебелов В.А., Новиков И.Е. *Генерация мягких теней при использовании алгоритма трассировки лучей // Вестник НГУ. Серия: Информационные Технологии. – 2009. – Т. 7, № 2. – С. 18–41.*

[6] Williams L., *Casting curved shadows on curved surfaces // Computer Graphics. – 1978. – Vol. 10, № 2. – P. 270-274.*

[7] Crow F. *Shadow Algorithms for Computer Graphics // Computer Graphics. – 1977. – Vol. 11, № 2. – P. 242-247.*

[8] Foley J., Van Dam A., a.o. *Computer Graphics Principles and Practice / 2nd Edition. – Addison Wesley. – 1990.*

Об авторах

Дебелов Виктор Алексеевич – доктор технических наук, ведущий научный сотрудник лаборатории численного анализа и машинной графики Института вычислительной математики и математической геофизики СО РАН, профессор Новосибирского государственного университета.
E-mail: debelov@oapmg.sscc.ru.

Новиков Илья Евгеньевич – аспирант Института вычислительной математики и математической геофизики СО РАН.
E-mail: iliya.novikov@gmail.com.

Light Meshes Method Acceleration Using GPU

Abstract

This work is devoted to Light meshes method (LMM) which is a modification of Whitted's ray tracing algorithm. LMM allows simulating soft shadows from point light sources that increases realism of synthesized images, but requires additional computational costs. Shadow maps (SM) and Shadow volumes (SV) are two popular algorithms for hard shadows generation. The main advantage of SM and SV is hardware acceleration by GPU. The idea of this work is to use fast light visibility detection by SM and SV for LMM, i.e. to construct hybrid algorithms based on LMM that fulfill visibility tests on GPU giving us an image generation speed-up while the same image quality.

Keywords: *photorealistic rendering, ray tracing, shadow map, shadow volume, light mehes method, soft shadows*.

About the authors

Victor A. Debelov is a doctor of technical sciences and holds position of leading researcher at Computer Graphics Lab of the Institute of Computational Mathematics and Mathematical Geophysics (http://www.sscc.ru). Also he is a professor at Novosibirsk State University.
E-mail: debelov@oapmg.sscc.ru.

Iliya Novikov is a Ph.D. student of the Institute of Computational Mathematics and Mathematical Geophysics (http://www.sscc.ru).
E-mail: iliya.novikov@gmail.com.

Indirect illumination on curve surfaces.

Boris Barladian
Keldysh Institute of Applied Mathematics RAS, Russia

Abstract

In this paper we suggest improved method of photon registration on curve surfaces, presented by triangulated mesh with “true” normals in mesh vertices. Such presentation is widely used for simulation the differing effects of light and color across the surface of an object (Phong shading). It was found that direct photon registration, which does not take into account interpolated (smooth) normal in given mesh point, creates visual artifacts. In the paper the modification of photon registration method is suggested. The modified method cures the problem with artifacts.

Keywords: *Global illumination, photon maps, Illumination maps, smooth shading.*

1. INTRODUCTION

Most of the modern methods of global illumination computation use photon registration on illuminated surfaces. The photon registration is used in photon map methods [1, 2] and in illumination maps technique. Illumination maps technique [3, 4] stores results of global illumination simulation (that is distribution of diffuse illuminance in the whole scene) in view-independent way. These data (called "illumination maps" or i-maps) allows generating a series of high-quality images (differing by observer position and viewing parameters) after time consuming global illumination simulation was done only once. The illumination maps provide also real-time walk-through with account for global illumination results with OpenGL based hardware. Certainly global illumination will be correctly taken into account only for pure diffuse surfaces, but it is acceptable in most cases when indirect illumination is significantly lower than direct one or when diffuse luminance is the dominant one. The two-pass method should be used for global illumination computation in case when pointed above conditions are not satisfied. At the first pass, the lighting distribution over scene surfaces is computed using radiosity [5, 6, 7] or photon mapping [2, 1, 8] methods. The illumination maps also can be used here. They differ from classic photon maps mainly in registration approach. In the second (rendering) pass the so-called final gathering [9, 5, 6, 7] approach should be used to provide high spatial resolution of lighting details for a given camera view. Only photon registration specific on curve surfaces presented by triangulated mesh with “true” normals in mesh vertices is considered here. It is not essential is the registration done in the form of photon map (hit is registered directly) or illumination map (hits are registered on the triangle vertices). The nature of problem is the same. Although in the case of final gathering the artifact problem will be partially hidden by illuminance averaging from different directions. Below the illumination maps will be used for problem illustration, because they are used in our system [4] directly.

2. PHOTON REGISTRATION ON ILLUMINATION MAP

The result of global illumination analysis in our system is represented by the so called "illumination maps" which describes illuminance distribution in the scene. Illumination maps keep illuminance values at each vertex of triangle mesh. Then these values can be linearly interpolated inside each triangle. So, illumination maps can represent an arbitrary continuous piece-wise linear function of illuminance distribution. For global illumination simulation our system uses Forward Monte Carlo Ray Tracing with unit energy photons. Each event that may change ray energy (a partial absorption in material or surface) is treated in the probabilistic way, by the Russian Roulette Rule: either the ray survives with the unchanged energy or it completely disappears. By a proper choice of probabilities of these two events we can simulate any rate of light absorption. In this sense we follow the global idea of using a random choice everywhere without decrease of simulation accuracy [3]. This idea was explicitly stated in [10] (in application to the Backward Monte Carlo ray tracing).

During Forward ray tracing ray-surface intersections (photon hits registration) are processed in the following way: if a ray hits a triangle that keeps i-maps then appropriate element(s) of the i-maps is modified to account the energy brought by the ray.
Let us denote triangle vertices as P1, P2 and P3 and illumination values for these vertices as V1, V2 and V3. These variables accumulate so called raw illumination maps. These values are proportional to the number of rays fired by scene light sources. Finally physical values are calculated multiplying by so called “elementary flux” which is equal to the total scene luminous flux divided on the number of traced rays. Taking into account our approach about linear interpolation of illuminance inside triangle the energy delivered by the given photon hit to the triangle should be distributed between the V1, V2 and V3 according the following formula:

V1 += B1; V2 += B2; V3 += B3; **(1)**

where **B1**, **B2** and **B3** are barycentric coordinates of ray/triangle intersection point in the triangle coordinate system.

3. CURVE SURFACE PROBLEM

The formula (1) is correct if triangulated mesh represents the flat surface, but it becomes incorrect if triangulated mesh represents curve surface. Let us consider simplified scheme of photon registration on curve surface represented by triangulated mesh with “true” normals in the mesh vertices:

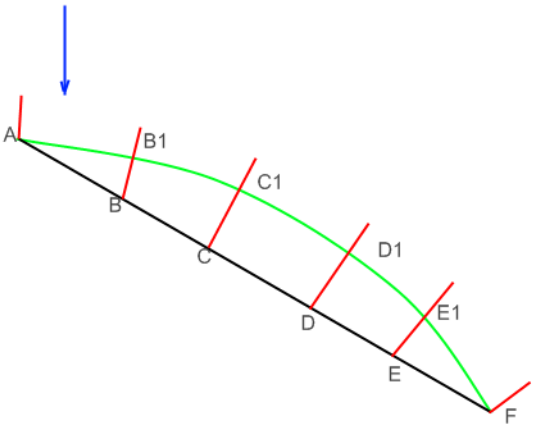


Fig. 1. Simplified scheme of photon registration.

We illustrate the problem on 2D drawing (fig. 1) instead of 3D one. The green curve AF denotes the curve surface from initial scene geometry. The set of grey segments (AB, BC, and so on) denote the triangulated mesh representing the green curve surface. The red segments (BB_1, CC_1, and so on) denote the "true" normals in triangle vertices of AF mesh. Let us blue arrow is the direction of parallel light illuminated our curve surface. From the fig. 1 we have that illumination of segment AB is proportional to the cosine between illumination direction and the segment AB normal, while illumination of curved segment AB_1 (let us consider it as sufficiently small one) is proportional to the cosine between the illumination direction and normal to the segment AB_1. Unfortunately the formula (1) does not take into account "true" normals in the triangle vertices. It corresponds to the appropriate flat geometry described by the triangulated mesh. So indirect illumination calculated by Forward Monte Carlo Ray Tracing will produce the "flat" image as it is presented in fig. 2 for test scene with cylinder and sphere. In the same time the image calculated for the curved surface illuminated by the parallel light using Phong shading is smooth (illustrated in fig. 3 for test scene).

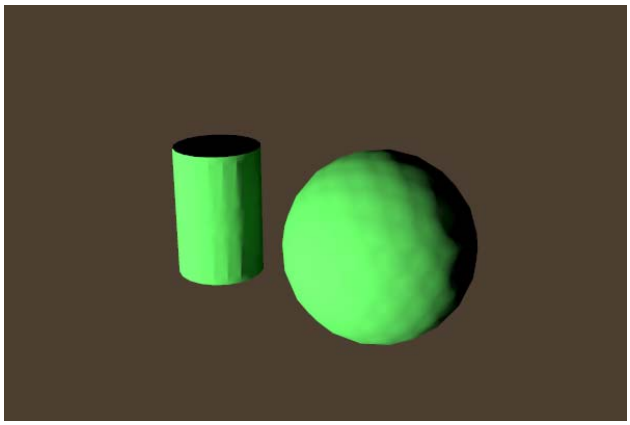


Fig. 2. Image obtained by Forward Monte Carlo Ray Tracing.

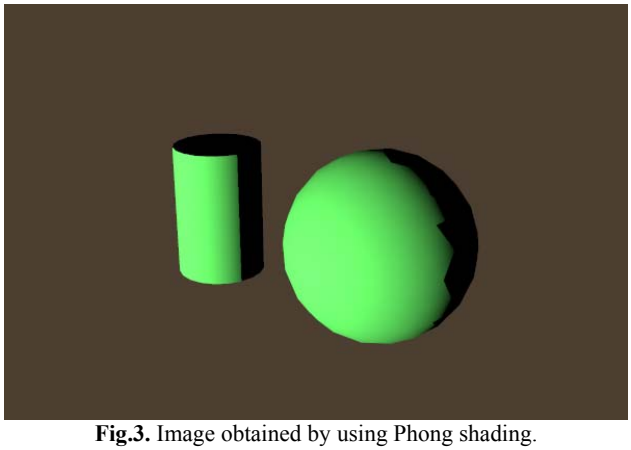


Fig.3. Image obtained by using Phong shading.

4. PROBLEM SOLUTION

As it was already pointed the illumination of the small surface area (fig.1, segments AB_1, B_1C_1, and so on) should be proportional to the cosine between the illumination direction and the "true" surface normal in the given point while the formula (1) uses in fact the cosine between illumination direction and the "flat" surface normal. So this difference can be compensated by multiplying the registered photon energy by the ratio of the cosine between illumination direction and the "true" surface normal to the cosine between illumination direction and the "flat" surface normal:

m = DotProd(ray, s_norm)/DotProd(ray, s_norm) **(2)**

Finally instead formula (1) we should use the formula (3):

V1 += B1*m; V2 += B2*m; V3 += B3*m; **(3)**

Fig. 4 shows our test scene calculated applying "true" normal compensation (formula (3)) during i-maps calculation. It is visible that artifacts disappear and surfaces look smooth.

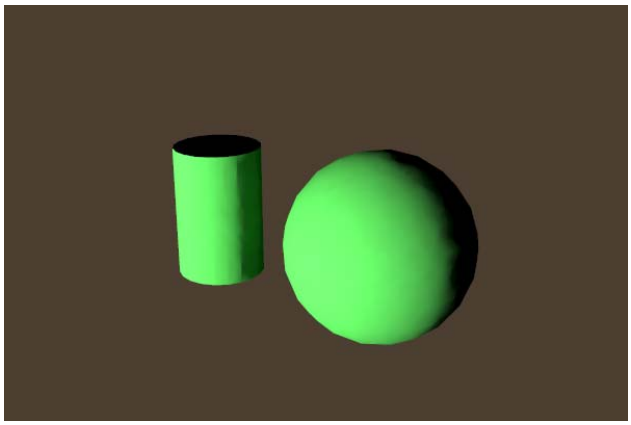


Fig. 4. Image obtained by modified Forward Monte Carlo Ray Tracing.

Using Forward Monte Carlo Ray Tracing for direct illumination calculation is inefficient and was done for problem demonstration only. More realistic example is illumination of this scene by

secondary light – by light reflected from a surface. We replace the parallel light source in this scene by the disk which reflects light. The obtained results are shown on fig.5 and fig.6.

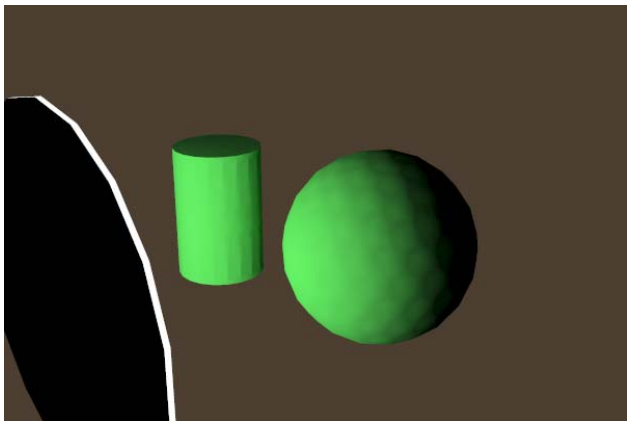


Fig. 5. Image obtained by original Forward Monte Carlo Ray Tracing.

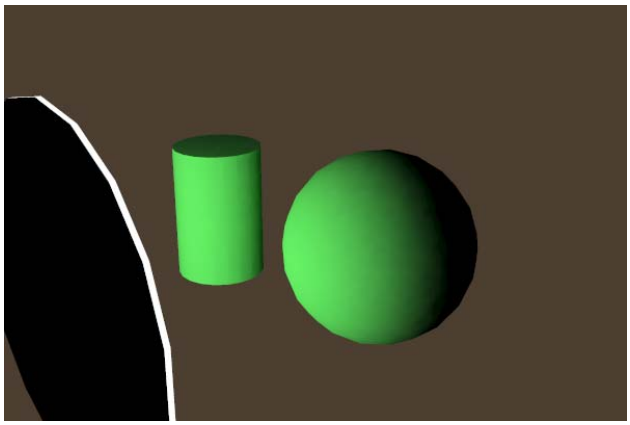


Fig. 6. Image obtained by modified Forward Monte Carlo Ray Tracing.

It should be noted that suggested method compensates artifacts produced by triangulated mesh representation of curve surface only partially. First of all it can not compensate the artifacts produced by some "shading" of triangulated mesh edges, which does not exist for real curve surface. At the second it also can not correct the artifacts produced by difference in ray paths for real and triangulated surfaces. This difference is critical for rays close to the tangential one to the real surface. The ray can miss the triangulated surface while intersect the real one and vice versa. Nevertheless overall image quality is significantly improved by the suggested method.

5. PRACTICAL RESULTS

The problem described above takes place for real scenes, when the curve surfaces are illuminated mainly by indirect illumination. The images below were obtained for the cabin of aircraft by original (fig. 7) and modified Forward Monte Carlo Ray Tracing (fig. 8). The whole illumination in cabin of aircraft is indirect one. It is typical illumination for the cabin of aircraft.

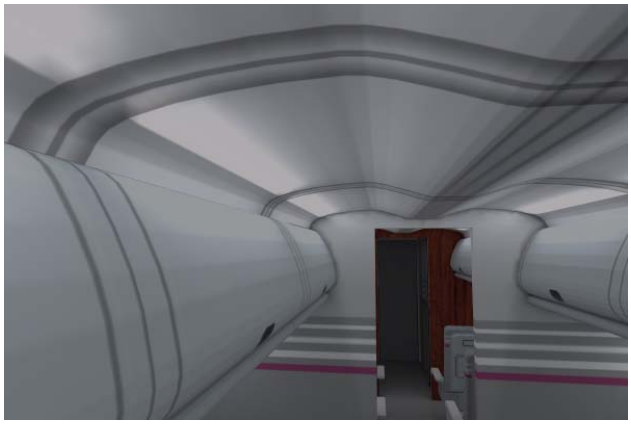


Fig. 7. Image obtained by original Forward Monte Carlo Ray Tracing.

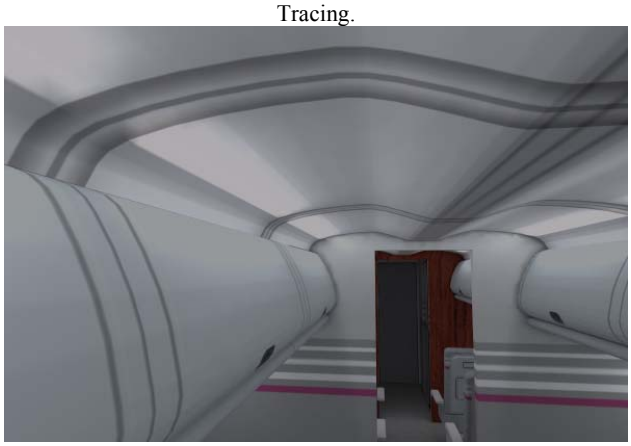


Fig. 8. Image obtained by modified Forward Monte Carlo Ray Tracing.

Image quality was essentially improved, but some artifacts are still visible. These artifacts can be removed almost completely by using illumination maps filtration. The result is shown on fig.9.

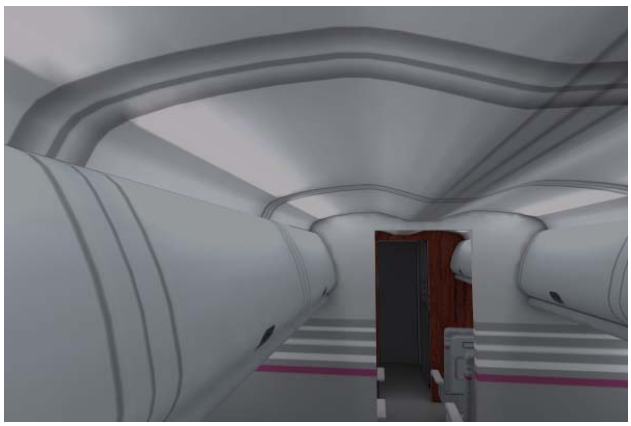


Fig. 9. Image obtained by modified Forward Monte Carlo Ray Tracing with illumination maps filtration.

6. ACNOWLEGMENTS

This work was supported by RFBR, grants № 07-01-00450, 09-01-00472, and INTEGRA, Inc.

7. REFERENCES

[1] JENSEN H.: *Global illumination using photon maps.*In Rendering Techniques '96, Proc. 7th Eurographics Workshop on Rendering (1996), Springer-Verlag, pp. 21–30.

[2] JENSEN H.: *Realistic Image Synthesis using Photon Mapping.* A. K. Peters, 2001.

[3] A.Khodulev, E.Kopylov *Physically accurate lighting simulation in computer graphics software /* Proc. GraphiCon'96 - The 6-th International Conference on Computer Graphics and Visualization, St.Petersburg, 1996, p. 111-119.

[4] Kopylov E., Khodulev A., Volevich V. *The Comparison of Illumination Maps Technique in Computer Graphics Software* / Proc. 8th International Conference on Computer Graphics and Visualization, Russia, Moscow, 1998, p. 146-153.

[5] D. Lischinski, F. Tampieri, and D. P. Greenberg. Combining hierarchical radiosity and discontinuity meshing. In *Proc. Of Siggraph'93*, pages 199–208, 1993

[6] B. Smits. Efficient Hierarchical Radiosity in Complex Environments. Ph.D. thesis, Cornell University, 1994.

[7] P. Christensen, D. Lischinski, E. Stollnitz, and D. Salesin. Clustering for glossy global illumination. *ACM Transactions on Graphics*, 16(1):3–33, 1997.

[8] P. Christensen. Photon mapping tricks. In Siggraph 2002, Course Notes No. 43, A Practical Guide to Global Illumination *using Photon Mapping organized by Jensen, H.W.*, pages 93–121, 2002.

[9] M. Reichert. *A Two-Pass Radiosity Method to Transmitting and Specularly Reflecting Surfaces*. M.Sc. thesis, Cornell University, 1992.

[10] R.L.Cook, T.Porter, L.Carpenter. Distributed Ray Tracing. Comp. Graph. (SIGGRAPH'84 Proc.), V.18(3), p.137-145, 1984.

About the author

Boris H. Barladian, PhD, senior researcher of the Keldysh Institute for Applied Mathematics RAS.

E-mail: obb@gin.keldysh.ru

Метод активного скелета в задаче распознавания формы изображений

Ирина Бакина, Леонид Местецкий, Арчил Цискаридзе *
Московский государственный университет имени М.В. Ломоносова, Москва, Россия
Московский физико-технический институт, Москва, Россия
irina_msu@mail.ru, l.mest@ru.net, achikotsi@gmail.com

Аннотация

В работе рассматривается аналог метода активных контуров для задачи сравнения формы изображений. Считается, что тестовые объекты представляют собой замкнутые односвязные области; для описания эталонов используется модель гибкого объекта.

Ключевые слова: Метод активных контуров, активный скелет, гибкий объект, распознавание формы.

1. ВВЕДЕНИЕ

В работе предлагается подход к распознаванию формы изображений, основанный на идеях метода активных контуров[1].

Описание эталонного образца в виде гибкого объекта является удобной формой его представления [2], которая включает в себя описание циркулярного графа (или циркуляра) объекта с заданием множества допустимых деформаций. В двухмерном случае циркулярный граф представляет собой осевой граф (или скелет), с каждой вершиной которого связан круг, отражающий ширину объекта в точке — центре круга. В этой модели силуэт объекта рассматривается как замкнутая область, ограниченная огибающей семейства кругов циркуляра. С помощью применения трансформаций к элементам скелета происходит моделирование различных движений объекта и его частей, которые влекут за собой изменение силуэта объекта. Таким образом, можно пытаться "шевелить" скелет эталона с целью получения наилучшего совмещения силуэта гибкого объекта с тестовым изображением. Указанная идея переносится также на трехмерный случай, где циркулярный граф является пространственным. Здесь уже появляется задача распознавания формы объекта по его двухмерному изображению.

Мотивация для выбора термина "активный скелет" объясняется двумя моментами, определяющими предлагаемый подход и аналогию с методом активных контуров. Они сводятся к следующему:

- Осуществляется трансформация эталона под измеренный тестовый объект, а не наоборот.
- Применение деформаций производится с целью подгонки форм эталонного и тестового образцов. При этом в случае активного контура форма описывается своей границей, а в случае активного скелета — осевым графом циркуляра.

*Работа выполнена при финансовой поддержке РФФИ, проекты №08-01-00670, №08-07-00305 и №08-07-00270.

2. ПОСТАНОВКА ЗАДАЧИ

Имеется некоторая база эталонных объектов, которая формируется offline, может проходить некоторую предварительную обработку экспертами и представляет собой совокупность моделей реальных объектов. Например, в случае распознавания жестов в качестве такой базы может быть рассмотрена азбука жестов глухонемых.

Распознаваемый на изображении тестовый образец представляет собой замкнутую односвязную область, для которой необходимо подобрать наиболее похожий эталон из базы.

Обозначим исходный набор эталонных образцов, заданных в форме гибкого объекта, через $E = \{E_i\}_{i=1}^n$, где n — общее число эталонов. Пусть $V_i = \{v_i^\alpha\}_{\alpha \in A}$ — совокупность возможных трансформаций i-ого эталона, а $v_i^\alpha(E_i)$ — гибкий объект, получающийся в результате применения к эталону E_i трансформации $v_i^\alpha \in V_i$ (здесь $A \subset \mathbb{R}$ — множество индексов). Пусть T_i — осевой граф циркуляра эталонного образца, а S_i — его силуэт. Тогда каждый эталон может быть описан следующей парой:

$$E_i = (T_i, V_i) \tag{1}$$

Пусть F — распознаваемая фигура. Введем функцию сходства $\mu(E_i, F)$ исходного эталонного и предъявляемого тестового образцов (аналог функции энергии в методе активных контуров). Тогда в рассмотренных обозначениях задача минимизации имеет вид:

$$(i^*, v_{i^*}^{\alpha^*}) = \underset{v_i^\alpha \in V_i}{argmin}\, \mu(v_i^\alpha(E_i), F) \tag{2}$$

Здесь i^* — номер ближайшего эталона, а $v_{i^*}^{\alpha^*}$ — трансформация, при которой достигается его наилучшее совпадение с распознаваемой формой.

Будем считать, что множество трансформаций $V = \{V_i\}_{i=1}^n$ для всех эталонных объектов определяется совокупностью параметров $\vec{a} = (a_1, \ldots, a_k)$, т.е. $V = \{v(\vec{a})\}_{\vec{a} \in D}$, где $D = \bigcup_{i=1}^n D_i \subset R^k$, а $V^i = \{v(\vec{a})\}_{\vec{a} \in D_i}$. При этом каждый тестовый объект описывается набором соответствующих параметров $\vec{a_i} = (a_1^i, \ldots, a_k^i)$: $E_i = E_i(\vec{a_i})$. Тогда задача минимизации (2) сводится к нахождению похожего эталона E_{i^*} и параметров его трансформации $\vec{a_{i^*}}$ таких, что:

$$(i^*, \vec{a_i^*}) : \mu(v(\vec{a_i^*}), F) = \min_{i=1,\ldots,n} \min_{\vec{a^i} \in D_i} \mu(v(\vec{a^i}), F) \tag{3}$$

3. ДЕКОМПОЗИЦИЯ ЗАДАЧИ

Переход от задачи в форме (2) к ее представлению в виде (3) позволяет рассматривать процесс глобальной оптимизации как совокупность двух подзадач.

Задача нахождения внешнего минимума является задачей классификации тестового объекта. Вначале определяется эталонный образец E_i и соответствующее его набору деформаций множество D_i. В рамках этого множества осуществляется нахождение внутреннего минимума.

Задача нахождения внутреннего минимума — задача подгонки. При ее решении происходит поиск множества трансформаций активного скелета с целью получения наилучшего совпадения с тестовым образцом. Так как каждая трансформация описывается набором параметров $\vec{a} = (a_1, \ldots, a_k)$ и $v = v(\vec{a})$, а каждый эталон задается конкретным набором их значений $E_i = E_i(\vec{a_i})$, то задача подгонки сводится к задаче поиска локального минимума функции $\mu(v(\vec{a}), F)$ при начальном приближении $\vec{a_i}$.

4. МЕТОД АКТИВНОГО СКЕЛЕТА

Как отмечалось выше, задача нахождения внутреннего минимума в (3) является задачей подгонки, которую предлагается решать методом активного скелета.

Рассмотрим двухмерный случай, когда имеется двухмерный гибкий объект (эталон) и сравниваемая фигура. Если совместить силуэт эталона с фигурой (рисунок 1а), то они сильно отличаются, хотя являются двумя представлениями ладони одного человека. Если же незначительно "пошевелить" скелет эталона,

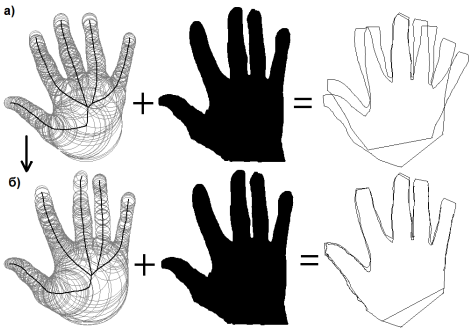


Figure 1: Гибкий объект (левая колонка), сравниваемая фигура (в центре) и результат их совмещения (правая колонка).

моделируя движение пальцев руки, то силуэт эталона и сравниваемое изображение практически совпадут (рисунок 1б).

В трехмерном случае наблюдается похожая ситуация. Предположим, что имеется пространственная эталонная поза человека, описывающаяся моделью пространственного гибкого объекта. Пусть человека попросили повторить эту позу, после чего сделали ее снимок. В результате, если рассмотреть силуэт гибкого объекта в плоскости камеры и на снимке, то мы увидим их значительное отличие (рисунок 2а). Однако, как и в двухмерном случае, путем "шевеления" скелета пространственного гибкого объекта-позы, можно добиться хорошего совпадения его силуэта со снимком (рисунок 2б).

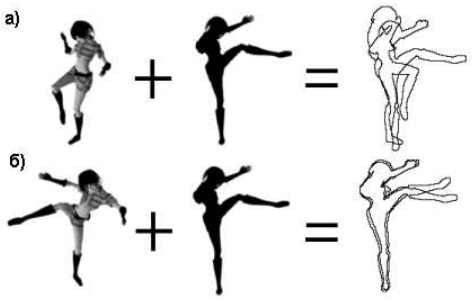


Figure 2: Пространственная поза (левая колонка), сравниваемая фигура (в центре) и результат совмещения их силуэтов (правая колонка).

В данной работе рассматриваются примеры приложения метода активного скелета к двум задачам сравнения формы. Это двухмерная задача распознавание формы ладони человека и трехмерная задача определения позы человека.

5. РАСПОЗНАВАНИЕ ФОРМЫ ЛАДОНИ

Имеется база эталонных бинарных изображений ладоней группы людей. Считается, что эталоны являются "хорошими" изображениями — на них отсутствуют артефакты (кольца и браслеты, длинные ногти, "склеенные" пальцы и т.д.). В то же время, распознаваемые изображения могут обладать ими.

База эталонных ладоней формируется в offline-режиме и содержит в себе модели эталонных изображений ладоней в форме гибкого объекта. Таким образом, имеем $E = \{E_i\}_{i=1}^{n}$, где n — число эталонных моделей. Тестовые изображения ладоней являются бинаризованными изображениями, полученными с web-камеры.

5.1 Ладонь как гибкий объект

На рисунке 1 в левом столбце представлены циркуляры ладоней. Чтобы задать ладонь в форме гибкого объекта, необходимо дополнительно указать группу деформаций V_i. Определим ее как множество допустимых поворотов пальцев вокруг их оснований (за исключением большого пальца).

Приведение ладони в стандартное положение и процедура нахождения оснований пальцев описаны в работе [3]. На рисунке 3 найденные основания пальцев обозначены звездочками.

Определим параметры трансформации. Рассмотрим углы между осями пальцев и осью среднего пальца руки. На рисунке 3 это углы α_1, α_2, α_3 и α_4. Тогда $E_i = E_i(\vec{\alpha})$, $\vec{\alpha} = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)$. Возможные значения углов поворота пальцев определяют множество $D_i \subset R^4$, а множество трансформаций есть $V_i = V_i(\vec{\alpha})$, где $\vec{\alpha} \in D_i$.

5.2 Сравнение ладоней

Для тестового изображения ладони, также как и для эталонного, можно построить циркулярный граф и привести его в стандартное положение.

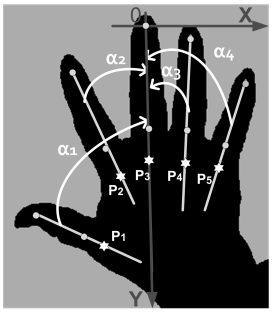


Figure 3: Определение параметров ладони.

По циркулярам эталонного и тестового изображений ладоней далее строится силуэт каждого из них. Через точки поворота указательного пальца и мизинца на эталоне проводится прямая линия, и части силуэтов, лежащие ниже этой прямой, отсекаются.

В качестве функции энергии активного скелета рассматривается площадь симметрической разности:

$$\mu(E_i, F) = Area(S_i \setminus F) + Area(F \setminus S_i) \qquad (4)$$

При применении трансформаций энергия скелета меняется, и задача подгонки для (3) имеет вид:

$$\vec{\alpha_i^*} : \mu(v_{\vec{\alpha_i^*}}(E_i), F) = \min_{\vec{\alpha_i} \in D_i} \mu(v_{\vec{\alpha_i}}(E_i), F) \qquad (5)$$

При сравнении ладоней осуществляются только попарные сравнения пальцев: мизинца с мизинцем, безымянного пальца с безымянным и т.д. То есть этап поиска начального приближения присутствует неявно.

Минимизация (5) выполняется за один шаг — мы измеряем углы между пальцами на тестовом образце, а затем "шевелим" пальцы эталона с целью получения не нем тех же самых углов.

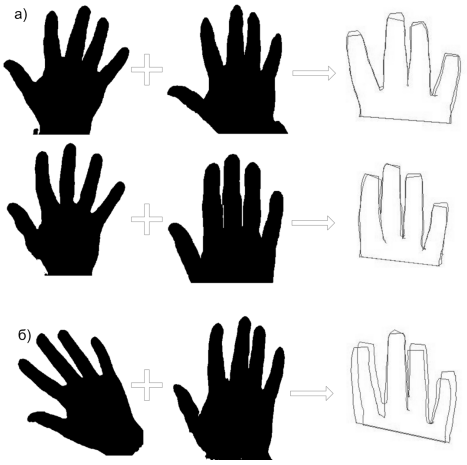


Figure 4: Сравнение ладоней: а) одного человека и б) разных людей.

На рисунке 4 представлен пример сравнения эталонных ладоней (первая колонка) с тестовыми (вторая колонка). В правой колонке представлен результат наилучшего совмещения их силуэтов.

6. РАСПОЗНАВАНИЕ ПОЗЫ ЧЕЛОВЕКА

Фигуру человека можно приближенно рассматривать как совокупность шарнирно закрепленных цилиндрических тел в пространстве, а движение моделировать путем вращениями этих тел вокруг точек крепления. Поза человека — взаимное расположение твердых частей, составляющих его фигуру. Будем считать, что пространственные формы этих тел уже известны и представлены моделью пространственного гибкого объекта.

Предполагается, что имеется пара откалиброванных относительно друг друга web-камер, и эти камеры позволяют с достоверностью выявлять лишь силуэты представленных на изображении объектов. Требуется подобрать такое "шевеление" исходного трехмерного объекта, чтобы его силуэт был похож на каждое из изображений, полученных с камер.

6.1 Метод активного скелета в пространстве

Будем описывать фигуру человека как пространственный гибкий объект $G = (\bigcup_{i=1}^{K} C_k, V)$ состоящий из $C_k = (O_{k,\alpha}, r_{k,\alpha}), \alpha \in A_k : k = 1, \ldots, K$ твердых частей, каждая из которых является семейством сфер с центрами $O_{k,\alpha}$ и радиусами $r_{k,\alpha}$.

Шевеление пространственного гибкого объекта представим в виде совокупности поворотов R_k и переносов t_k, соответствующих его частям C_k. Шарнирные крепления можно записать в виде $(a_l, i_l, b_l, j_l) : \{a_l, b_l\} \in R^3, \{i_l, j_l\} \in 1, \ldots, K, l = 1, \ldots, L$. В результате имеем L уравнений: $R_{i_l} a_l + t_{i_l} = R_{j_l} b_l + t_{j_l}$.

Задачу распознавания позы по двум силуэтам (F_1, F_2) запишем в виде:

$$(R_i^*, t_i^*) = \underset{\substack{R_i; t_i \in R^3 \\ R_{i_l} a_l + t_{i_l} = R_{j_l} b_l + t_{j_l} \\ i=1,\ldots,K, l=1,\ldots,L}}{argmin} \mu(\bigcup_{i=1}^{K} (R_i C_i + t_i), (F_1, F_2)) \qquad (6)$$

Здесь μ сумма Хаусдорфовых мер различия между силуэтами.

Основная трудность заключается в том, что на плоских силуэтах могут присутствовать окклюзии, и задача (6) имеет много локальных минимумов. Данная трудность может быть преодолена путем предварительной классификации наблюдаемых силуэтов — в случае окклюзий создается база эталонов для их различных конфигураций. В данной задаче эталоном является пространственная поза $v(G), v \in V$.

Для тестовой пары силуэтов, задача минимизации (6) решается в два этапа:

- Определение наиболее похожей конфигурации окклюзии в базе эталонов.
- Осуществление подгонки выбранного на предыдущем шаге эталона.

Для сравнения пары скелетов из рисунка 5а используем метод выделения изоморфной части [4] (похожий подход предложен в [5]), после чего пара скелетов имеет изоморфные оси (рисунок 5б).

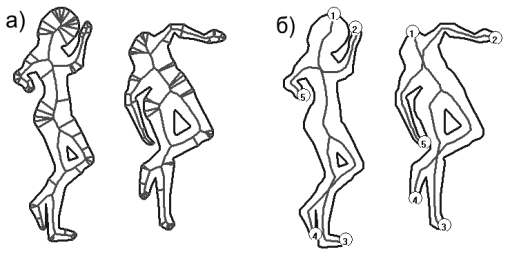


Figure 5: а) Скелеты силуэтов для разных поз и б) общая изоморфная часть скелетов.

Каждая точка силуэта может быть задана в системе координат скелета таким образом, что при деформации скелета ее координаты не меняются. Зная изоморфизм скелетов, можно каждой точке одного силуэта сопоставить точку на другом силуэте, требуя, чтобы у них были одинаковые координаты. Это позволяет строить начальное приближение.

Весь цикл выглядит следующим образом:

- Проецируем пространственный эталон на плоскости камер.
- Строим скелеты для полученных силуэтов.
- Строим изоморфизм между скелетом эталона и скелетами тестовых силуэтов.
- По изоморфизму скелетов строим отображение точек силуэтов.
- Определяем переход проекций осей пространственного эталона при преобразовании силуэтов. По двум проекциям с помощью эпиполярной геометрии [6] строим новые оси в пространстве, получая пространственную деформацию осей эталона.
- Полученную деформацию берем как начальное приближение и запускаем методы локальной минимизации.

6.2 Схема точной подгонки

Поскольку точки каждой твердой части должны сохранять расстояния между собой, скорректируем начальное приближение, поставив задачу определения поворота R_i и переноса t_i для каждой i-ой части $i = 1..K$ как:

$$(R_i^*, t_i^*) = \underset{R_i}{argmin} \sum_{\alpha \in A_i} |R_i O_{i,\alpha} + t_i - O'_{i,\alpha}|^2 \qquad (7)$$

Здесь $O'_{i,\alpha}$ — оси начального приближения.

Наличие шарнирных точек накладывает на R_i^*, t_i^* дополнительные ограничения: $R_{i_l} a_l + t_{i_l} = R_{j_l} b_l + t_{j_l}$. Задачу точной подгонки запишем как:

$$(R_i^*, t_i^*) = \lim_{\lambda \to \infty} \underset{R_i}{argmin}\, \mu(\bigcup_{i=1}^{K}(R_{ii} + t_i), F) + \lambda \sum_{l=1}^{L} |R_{i_l} a_l + t_{i_l} - R_{j_l} b_l - t_{j_l}|^2 \qquad (8)$$

Таким образом, получив скорректированное начальное приближение, можно запустить градиентный метод для задачи (8).

7. ВЫВОДЫ

В работе рассмотрена общая постановка задачи сравнения формы объектов в случае, когда эталоны представлены моделью гибкого объекта, а тестовые образцы являются замкнутыми односвязными областями. Проведена декомпозиция задачи на две подзадачи — классификации и минимизации (задача классификации актуальна только для случая присутствия окклюзий). Рассмотрен подход к решению обеих подзадач. Для решения задачи минимизации предложен метод активного скелета как аналог метода активных контуров. Основная идея подхода состоит в "шевелении" скелета эталона с целью получения наилучшего совпадения его силуэта с силуэтом тестового экземпляра. Приведены примеры приложения этого метода к задачам сравнения формы для случаев, когда эталонный объект задан в двухмерном и трехмерном пространствах.

8. REFERENCES

[1] M. Isard A. Blake, “Active contours: The application of techniques from graphics, vision, control theory and statistics to visual tracking of shapes in motion,” Springer, 2000.

[2] Л. Местецкий, “Непрерывная морфология бинарных изображений: фигуры, скелеты, циркуляры,” Москва: ФИЗМАТЛИТ, 2009.

[3] Л. Местецкий И. Бакина, “Метод сравнения формы ладоней при наличии артефактов,” Математические методы распознавания образов (ММРО-14), 2009.

[4] A. Okhlopkov L. Domakhina, “Shape comparison based on skeleton isomorphism,” VISAPP, pp. 237–242, 2009.

[5] A. L. Yuille S. C. Zhu, “Forms: A flexible object recognition and modeling system,” IJCV, vol. 20, no. 3, pp. 187–212, 1996.

[6] A. Zisserman R. Hartley, “Multiple view geometry in computer vision,” Cambridge University Press, 2004.

ACTIVE SKELETON FOR IMAGE FORM RECOGNITION

Abstract

Analogy to the method of active contours is presented for the problem of form recognition in the case when etalon objects are described in terms of flexible object, while test objects are closed 1-connected domains.

Keywords

The method of active contours, active skeleton, flexible object, form comparison.

Очищение текстур фасадов зданий с использованием их структуры

Владимир Кононов, Вадим Конушин, Антон Якубенко, Антон Конушин
Факультет Вычислительной Математики и Кибернетики
Московский Государственный Университет им. М.В. Ломоносова, Москва, Россия
{vladimir.kononov, vadim, toh, ktosh}@graphics.cs.msu.ru

Аннотация

В настоящее время активно развивается область трехмерной реконструкции городов по изображениям. Ее важной составной частью является создание реалистичных текстур фасадов городских зданий по фотографиям. Но на реальных снимках фасад зачастую загорожен различными объектами переднего плана, такими как деревья, провода, дорожные знаки и т.д. Поэтому для качественной реконструкции необходимо очищение текстуры фасадов от таких объектов. Современные методы очистки текстур работают очень медленно и не подходят для реального применения. В этой работе предлагается интерактивный алгоритм для нахождения и удаления объектов переднего плана на текстурах фасадов, основанный на использовании информации об их структурах.

Ключевые слова: *восстановление текстур, обработка изображений, 3D-реконструкция зданий, анализ структуры фасадов.*

1. ВВЕДЕНИЕ

Одним из этапов построения трехмерных моделей зданий как составной части трехмерной реконструкции городов является очищение текстур, полученных из фотографий, от различных объектов переднего плана, загораживающих фасад. Актуальность данной задачи вытекает из невозможности во многих случаях сфотографировать здание без посторонних объектов перед ним.

В этой статье предлагается алгоритм очистки текстуры зданий от объектов переднего плана с использованием информации о структуре фасада. На вход алгоритму подается ректифицированная текстура фасада, а также координаты и типы окон. Ректифицированная текстура может быть, например, получена указанием четырех углов стены на исходном изображении с последующим расчетом гомографии. На выходе алгоритма – текстура фасада с восстановленными областями под найденными объектами переднего плана. Ключевым требованием к алгоритму является скорость работы, достаточная для интерактивного взаимодействия.

Положение и типы окон могут быть получены с помощью автоматических или полуавтоматических алгоритмов, как [2]; или же размечены вручную. В предложенном алгоритме структура фасада задается в виде сетки из прямоугольных ячеек, каждая из которых содержит одно окно. Внешний вид всех ячеек одного типа считается одинаковым. Количество типов ячеек определяется количеством заданных типов окон, поэтому если в двух разных по внешнему виду ячейках находятся одинаковые окна, они должны быть отмечены как два разных типа окна (Рис. 1). Это может быть сделано вручную, или с использованием алгоритмов, умеющих различать типы ячеек, а не окон. В любом случае возможные усилия, затрачиваемые пользователем на этом этапе значительно меньше усилий, затрачиваемых на очищение текстуры существующими методами.

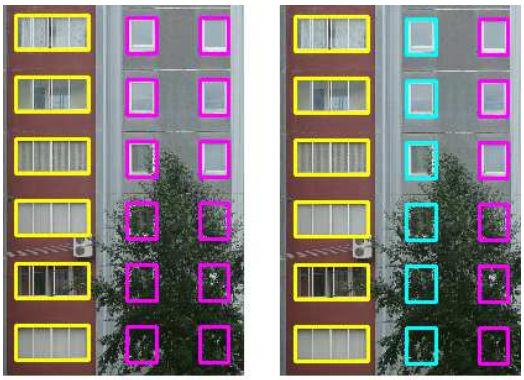


Рис. 1: Пример входных данных алгоритма: a) неправильно – типы окон не соответствуют типам ячеек; b) правильно – типы окон соответствуют типам ячеек.

2. СУЩЕСТВУЮЩИЕ МЕТОДЫ

К настоящему времени сформировалась целая исследовательская область по задаче восстановления текстуры (Image Completion). Эта задача состоит в заполнении некоторого участка изображения, называемого неизвестной областью, с использованием оставшейся части изображения, причем результат визуально не должен выглядеть неестественно. За последние десять лет в области восстановления текстуры появилось множество различных алгоритмов, которые можно грубо разделить на два больших класса.

Методы первого класса основаны на составлении и решении уравнений в частных производных [1]. Алгоритмы этого класса хорошо подходят только для восстановления узких и небольших областей, таких как царапины на фотографиях. В противном случае результат получается очень размытым.

Методы второго класса основаны на копировании информации с остальной части изображения на неизвестную область попиксельно или небольшими фрагментами. Некоторые алгоритмы жадно заполняют неизвестную область, часто с использованием функции приоритета [3]. Другие же формулирует специальную функцию энергии, и находят такое копирование фрагментов, которое минимизирует эту функцию [4].

В любом случае всем алгоритмам восстановления текстур необходима размеченная вручную или другими алгоритмами неизвестная область. Также они работают непозволительно долго для использования их в интерактивных приложениях (от нескольких десятков минут до нескольких часов на изображениях в 1 – 3 мегапикселя). Поэтому в чистом виде они не могут применяться для решения поставленной задачи.

В статье [9] авторы также имеют дело с фасадами. Однако их алгоритм использует размеченную вручную неизвестную область, а также ограничивается работой с фасадами, высоты всех этажей которых одинаковые.

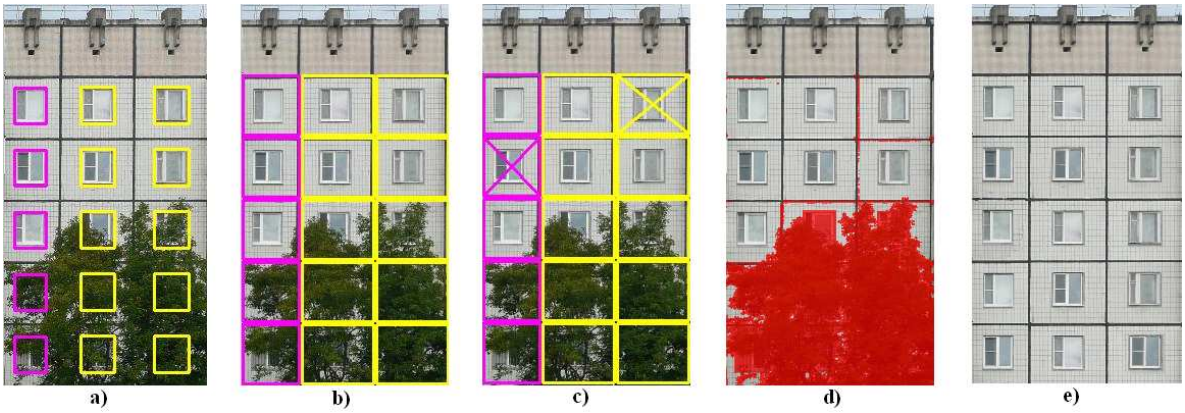


Рис. 2: Схема работы предложенного алгоритма: a) входная текстура, координаты и типы окон, b) построение структуры, c) нахождение чистой ячейки каждого типа, d) сегментация, e) восстановление текстуры.

Кора и Расмуссен в [6] недавно предложили алгоритм для автоматического восстановления структуры фасада с последующим нахождением и удалением объектов переднего плана, загораживающих здание. Схема этого метода похожа на схему предлагаемого алгоритма. Однако этот метод работает только для простых случаев загораживания. Например, он не справляется со случаями, когда объекты переднего плана загораживают более половины или близкое к этому число структурных ячеек. Также все примеры результатов работы показаны только для простых фасадов.

В данной статье предложен алгоритм, единственным ограничениям которого является наличие хотя бы одной полностью чистой ячейки каждого типа. Он не нуждается в ручной разметке неизвестной области, и может работать как полностью автоматически, так и интерактивно с минимальным пользовательским взаимодействием.

3. ОПИСАНИЕ ПРЕДЛОЖЕННОГО МЕТОДА

Предложенный алгоритм состоит из четырех шагов (Рис. 2):

1. Построение структуры фасада по окнам.
2. Нахождение чистой ячейки каждого типа.
3. Сегментация объектов переднего плана в остальных ячейках.
4. Восстановление текстуры в областях, отсегментировавшихся как объекты переднего плана.

3.1 Построение структуры фасада по окнам

На первом шаге алгоритм строит сетку фасада, то есть разбивает его на ячейки, используя информацию о координатах и типах окон. Изначально ячейки фасада считаются равными окнам. Затем они постепенно расширяются во все стороны до касания соседними элементами друг друга или границ фасада, при этом сохраняются размеры всех ячеек одного типа (Рис. 2b). Предложенный алгоритм подразумевает, что все ячейки одного типа имеют с точностью до освещения одинаковую текстуру, не считая возможных объектов переднего плана. Однако если для стен такое предположение естественно и выполняется для большинства зданий, то окна в одинаковых ячейках могут заметно отличаться по текстуре из-за разных занавесок, штор и возможности быть открытыми или закрытыми. Поэтому дальнейшая работа ведется с ячейками без окон, а сами окна обрабатываются отдельно.

3.2 Нахождение чистой ячейки каждого типа

После построения сетки фасада алгоритм работает отдельно с каждым типом ячеек. Следующим шагом является поиск полностью чистой ячейки. Он основан на использовании метрики сходства ячеек одного типа. Особенностью задачи сравнения является то, что ячейки могут отличаться друг от друга не только из-за загораживающих объектов переднего плана. Неточность разметки и неточность самого фасада часто приводит к сдвигу в несколько пикселей между текстурами ячеек одного типа. Также отличия могут быть вызваны разностью в освещении. Корректная метрика сравнения должна быть как можно более инварианта к этим особенностям.

Было рассмотрено несколько различных метрик, в том числе метрика на основе взаимной информации и сумма квадратов попиксельных разностей. На практике лучшие результаты показала метрика, сравнивающая близости краев. Для вычисления самих краев используется алгоритм Canny, а карты близости строятся с помощью алгоритма Distance Transform. Затем получившиеся карты сравниваются попиксельно (Рис. 3). Нормализованная сумма квадратов разностей по всем пикселям карт близости является значением метрики.

Рис. 3: Схема работы метрики сравнения ячеек. От краев к центру: исходные ячейки, результат алгоритма Canny, результат Distance Transform, попиксельное сравнение.

Далее решается проблема нахождения абсолютно чистой ячейки среди всех элементов одного типа. Один из вариантов решения состоит в том, чтобы создать синтетическую ячейку, каждый пиксель которой является медианой соответствующих точек всех элементов данного типа. Затем, используя метрики, можно посчитать расстояние от всех ячеек до синтетической, и наиболее близкую назвать чистой. Проблема данного метода в том, что полученная синтетическая ячейка часто сильно размыта, из-за чего чистая ячейка может найтись неправильно. Кроме того, такой метод работает только в случае, когда более половины ячеек являются чистыми, что само по себе очень жесткое ограничение. Из-за этих же недостатков нельзя сразу решать задачу целиком простой медианной заменой (Рис. 6a).

Поэтому был предложен другой метод, не создающий в процессе работы синтетических ячеек, а использующий только

расстояния между реально существующими элементами. Чистой ячейкой объявляется та, сумма расстояний от которой до других по метрике минимальна. Этот подход использует тот факт, что чистые ячейки похожи друг на друга, а содержащие объекты переднего плана – нет. Например, ячейка, содержащая дерево, не похожа на ячейку, содержащую фонарный столб. Для этого метода ограничение на входные данные формулируется следующим образом: на фасаде должна быть хотя бы одна полностью чистая ячейка каждого типа.

Другим плюсом нахождения полностью чистой ячейки является возможность клонирования ее на все остальные элементы. В ряде случаев этого бывает достаточно, но часто результат бывает заметно синтетическим и неестественным (Рис. 6b). Поэтому необходимы дальнейшие шаги по поиску объектов переднего плана и восстановлению текстуры под ними.

3.3 Сегментация ячеек с нахождением объектов переднего плана

На этом этапе пиксели ячеек сегментируются на два класса: фасад и объекты переднего плана. Предложенный алгоритм формулирует задачу сегментации в виде задачи минимизации энергии специального вида, состоящей из унарных и бинарных слагаемых.

$$E = \sum_{p} E_u(p, l) + \sum_{p,q} E_b(p, q, l_p, l_q)$$

Здесь p, q – различные пиксели, а l – метка, которая может принимать значение "фасад" или "объект переднего плана".

Чтобы полностью включить информацию о системе, надо учитывать в минимизируемой энергии пиксели со всех ячеек одного типа. То есть считать бинарные слагаемые не только для соседних пикселей внутри ячейки, но и для всех пикселей на одинаковых позициях в разных ячейках. Однако на практике этот метод оказывается неприменим из-за большой вычислительной сложности минимизации такой энергии.

Поэтому предложенный метод сегментации работает с каждой ячейкой поодиночке. Для формулирования энергии остается воспользоваться найденной абсолютно чистой ячейкой. К минусам такой модели относятся неполное использование знаний о системе, а также зависимость от правильного нахождения полностью чистой ячейки. К плюсам – высокая скорость работы и приемлемые на практике результаты.

Значения бинарных слагаемых для соседних точек должны отражать тот факт, что близкие по значению пиксели скорее всего относятся к одному классу: или оба "объекты переднего плана" \ или оба "фасад". Другими словами, объекты переднего плана выделяются на фоне фасада. Поэтому в случае различных меток точек энергия должна зависеть от цвета соседних пикселей и быть обратно пропорциональной их похожести. В случае одинаковых меток энергия просто равна нулю.

$$E_b(p, q, l_p, l_q) = \begin{cases} \exp(-\|I(p) - I(q)\|), & l_p \neq l_q \\ 0, & l_p = l_q \end{cases}$$

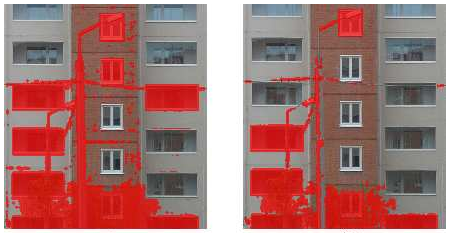
Унарные слагаемые должны отражать вероятности пикселей быть частью объекта переднего плана или частью фасада. Известна только одна чистая ячейка, соответственно можно считать, что чем ближе пиксель по цвету к соответствующему пикселю чистой ячейки, тем больше у него шансов быть частью фасада. Этот факт можно записать как значение унарного слагаемого при метке "фасад". При метке "объект переднего плана" значение унарного слагаемого является просто некоторой константой.

$$E_u(p, l) = \begin{cases} \exp(-\lambda * \|I(p) - I_{best}(p)\|), & l = facade \\ \exp(-\lambda * C), & l = foreground \end{cases}$$

I_{best} – абсолютно чистая ячейка, и C – экспериментально подобранная константа. λ остается единственным параметром системы, который показывает важность бинарных слагаемых по сравнению с унарными (Рис. 4). Изменяя только его, можно добиться приемлемого результата. Единого значения λ для всех случаев не существует. Это обусловлено различиями в освещенности ячеек и характере текстуры.

Для минимизации используется алгоритм разреза графов [7][8]. Используемая функция энергии удовлетворяет условиям регулярности, и, следовательно, может быть корректно минимизирована этим алгоритмом.

Окна на этом этапе сегментируются отдельно. Для этого применяется следующая эвристика. Окно считается либо целиком чистым, либо целиком загороженным в зависимости от количества пикселей по периметру, отсегментировавшихся как объекты переднего плана. Такой подход применяется из-за того, что окна могут сильно отличаться на разных ячейках, даже без наличия объектов переднего плана. Поэтому как сегментацию, так и восстановление текстуры, необходимо производить для всего окна целиком.



a) b)

Рис. 4: Результаты сегментации: a) $\lambda = 5$, b) $\lambda = 10$.

3.4 Восстановление текстуры под найденными объектами переднего плана

На этом шаге алгоритм восстанавливает текстуру фасада под найденными объектами переднего плана. Для этого используется найденная чистая ячейка, а также другие ячейки, оказавшиеся полностью чистыми после сегментации. Существует несколько подходов к решению задачи данного этапа.

Самый простой вариант – это заменить пиксели, отсегментированные как объекты переднего плана, на соответствующие им пиксели из чистой ячейки. Такой метод очень прост и быстро работает. Однако получающиеся результаты часто плохи. Это обусловлено тем, что чистая и целевая ячейки могут различаться по освещенности и точности разметки. Сходные проблемы возникают у всех методов, непосредственно копирующих пиксели из чистых ячеек.

Существует метод визуально естественной вставки части одного изображения в другое, который называется Poisson Image Editing [5]. Он основан на построении и решении уравнения Пуассона и учитывает перепады цвета во вставляемой части и краевые условия из целевого изображения. Этот метод можно непосредственно применять для вставки пикселей из чистой ячейки на места пикселей объектов переднего плана в других ячейках. Минусом данного метода является чувствительность к точности сегментации. Даже если один пиксель отсегментировался неправильно, он может войти в краевые условия, и Poisson Image Editing сработает не совсем кор-

ректно. Отчасти эту проблему можно решить сглаживанием краевых условий. Плюсом данного метода являются лучшие результаты (Рис. 5).

Рис. 5: Сравнение восстановления текстуры методом простой вставки и методом Poisson Image Editing.

4. РЕЗУЛЬТАТЫ

Данный метод был протестирован на различных зданиях г. Москвы и г. Кириши и показал неплохие результаты (Рис. 6c). Другим его достоинством является интерактивное время работы, а также предоставление пользователю возможности пошагово следить за ходом работы и поправлять результаты каждого шага по необходимости. Время работы предложенного метода составило в среднем 20 – 30 секунд в полностью автоматическом режиме для одного изображения 1400x1200 при реализации на C++ на компьютере Intel Celeron 1.5 GHz, 512 Mb RAM. С пользовательской коррекцией общее время обработки одного изображения возрастает до одной минуты. Для сравнения, реализованный на Matlab алгоритм [3] работает несколько часов, не считая ручного выделения объектов переднего плана. Метод из статьи [6] схож с предложенным в целом, но подходит для более узкого класса текстур. Сравнительное тестирование невозможно ввиду отсутствия доступной реализации метода, а также отсутствия данных по скорости работы в самой статье.

5. ЗАКЛЮЧЕНИЕ

В данной статье предложен алгоритм очищения текстур фасадов зданий от загораживающих их объектов переднего плана. Он может работать как в полностью автоматическом режиме, так и в интерактивном с минимальным пользовательским взаимодействием. Алгоритм показал хорошие результаты по качеству итоговой текстуры и по времени работы. В будущем планируется развить предложенный метод, расширив его применение на случай отсутствия абсолютно чистых ячеек. Также планируется провести корректное сравнение метода с [6].

6. БЛАГОДАРНОСТИ

Работа была выполнена по заказу компании "Медиа Софт Интегро" для Платформы CITY-3D и при частичной поддержке грантов РФФИ 08-01-00883-а, 09-01-92470-МНКС_а, и гранта Фонда "Научный потенциал" (Договор №184 от 14 января 2009 года).

7. ЛИТЕРАТУРА

[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, *Image Inpainting*, International Conference on Computer Graphics and Interactive Techniques, pp. 417–424, 2000

[2] M. Park, R. Collins, Y. Liu, *Deformed Lattice Discovery Via Efficient Mean-Shift Belief Propagation*, Proc. of the 10th European Conference on Computer Vision (ECCV), Part II, pp. 474–485, 2009

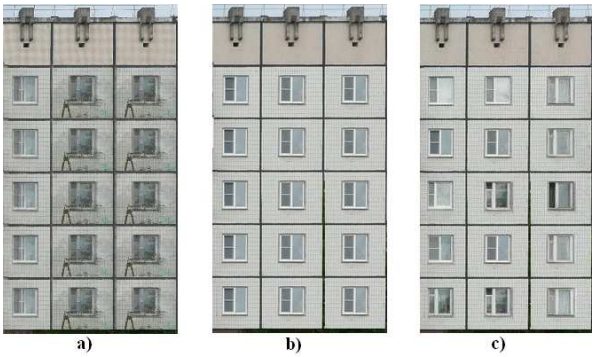


Рис. 6: a) Результат замены всех ячеек медианой; b) Результат замены всех ячеек абсолютно чистой; c) Результат работы предложенного алгоритма

[3] A. Criminisi, P. Perez, K. Toyama, *Object removal by exemplar-based inpainting*, In Conf. Computer Vision and Pattern Recog (CVPR) 2003, vol. 2, pp. 721–728, 2003

[4] Y. Wexler, E. Shechtman, M. Irani. *Space-Time Completion of Video*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 2, pp. 463–476, 2007

[5] P. Perez, M. Gangnet, A. Blake. *Poisson Image Editing*, ACM Transactions on Graphics (TOG), vol. 22, n. 3, 2003

[6] T. Korah, C. Rasmussen, *Analysis of Building Textures for Reconstructing Partially Occluded Facades*, Proc. of the 10th European Conference on Computer Vision (ECCV), Part I, pp. 359–372, 2008

[7] Y.Boykov, O. Veksler, R. Zabih, *Fast approximate energy minimization via graph cuts*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 23 (11): 1222-1239, 2001

[8] Y. Boykov, V. Kolmogorov, *An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Computer Vision*, In Third International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition, 2001

[9] V. Konushin, V. Vezhnevets, *Automatic building texture completion*, Proc. of Graphicon'2007, pp. 174-177, 2007

ОБ АВТОРАХ

Владимир Кононов является студентом Лаборатории компьютерной графики и мультимедиа Факультета вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова. Вадим Конушин и Антон Якубенко являются аспирантами той же лаборатории. Антон Конушин является сотрудником той же лаборатории.

ABSTRACT

Recently, 3D reconstruction of urban scenes has received much attention. One of its important problems is the creation of realistic facade textures from photos. Unfortunately, buildings facades are often partially occluded by various foreground objects such as trees, road signs, wires, etc. Therefore image completion is required. Modern image completion methods are too slow and not suitable for real applications. In this paper, we propose an interactive algorithm for foreground objects segmentation and occluded texture reconstruction, based on facade structure.

Сегментация формы пространственных изображений

Денис Хромов
Факультет Вычислительной математики и кибернетики
Московский государственный университет им. М.В. Ломоносова, Москва, Россия
khromovd@mail.ru

Аннотация

В статье предлагается подход к решению задач сегментации пространственных изображений, в которых анализируемые объекты состоят из пространственно неразделимых, слитных между собой компонент. Метод основан на использовании непрерыных скелетов плоских сечений пространственной модели. В качестве демонстрации взята конкретная практическая задача сегментации трёхмерного изображения слепка человеческой челюсти.

Ключевые слова: сегментация, форма объекта, полигональная модель, трёхмерное сканирование.

1. ВВЕДЕНИЕ

Прстранственным изображением будем называть геометрическое описание трёхмерного тела, полученное измерением реального физического объекта, например, при помощи 3D-сканера. Одной из задач, относящихся к анализу таких объектов, является задача сегментации. Сегментацией формы называется разбиение формы на осмысленные компоненты. Осмысленность разбиения определяется субъективным человеческим восприятием (см., например, [2], [3]). Существуют различные подходы к сегментации трёхмерных изображений. Так, в статье [3] предлагается решение, основанное на построении скелета изображения — одномерного многообразия в пространстве, удачно схватывающего топологические и геометрические свойства рассматриваемого объекта. А в [4] описывается другой способ сегментации, связанный с выделением особых областей, называемых выступами (protrusions). Все эти методы объединяет одна важная особенность — компоненты исследуемого изображения должны быть пространственно разделены между собой. Примером такого изображения может служить описание формы животного, в котором искомыми сегментами являются конечности.

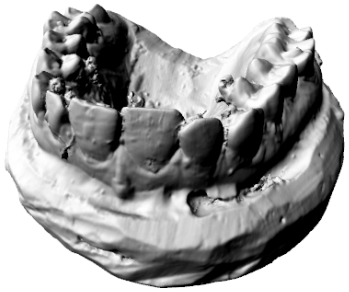


Рис. 1: Слепок челюсти.

В настоящей статье предлагается подход к решению задач сегментации, спецификой которых являются формы со смыкающимися между собой компонентами. В качестве приложения рассматривается задача сегментации зубов на слепке человеческой челюсти (см. рис. 1), изображение которого получено при помощи 3D-сканера. Слепок представлен нерегулярной полигональной моделью, содержащей около $7 \cdot 10^5$ вершин и $1,5 \cdot 10^6$ граней. Решение, представленное в данной работе, основывается на идее сведения задачи сегментации 3D объекта к анализу скелетов его двумерных сечений. Реализация этой идеи ставит две задачи: во-первых, выбор небольшого числа сечений, достаточных для анализа формы пространственного объекта; во-вторых, собственно сегментация трёхмерного объекта на основе построенных сечений.

2. ПОСТАНОВКА ЗАДАЧИ

Дано описание пространственной формы слепка человеческой челюсти, представленное полигональной моделью с треугольными гранями (см. рис. 1). Это описание получено при помощи 3D-сканера с гипсового слепка реальной челюсти и содержит порядка полутора миллионов граней. Отметим, что поверхность меша не является замкнутой (подробнее см. [5]). Требуется решить для этой формы задачу сегментации зубов, а именно: для каждой вершины v полигональной модели требуется указать номер её сегмента $\mathscr{T}(v)$, так, что вершины каждого из сегментов образуют или коронку[1] одного из зубов, или часть слепка, не содержащую в себе коронок. Сегменты зубов нумеруются от 1 до n (количество зубов n заранее неизвестно и должно быть также определено алгоритмом), сегмент, не содержащий зубов (он описывает поверхность дёсен и гипсовой подставки), имеет номер 0.

3. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

В этом разделе будут рассмотрены некоторые понятия, важные для дальнейшего изложения.

3.1 Срез

Пусть П — произвольная плоскость, пересекающая множество граней рассматриваемой формы. Соответствующие пересечения образуют на плоскости множество отрезков. Если задать на П ортонормированную систему координат, то для всех этих отрезков можно задать плоские двумерные координаты. Такое множество отрезков плоскости будем называть срезом по плоскости П.

На рис. 2 изображены примеры срезов. Видно, что отрезки образуют ломаные. При определённых положениях секущей плоскости (см. нижний пример) эти ломаные замкнуты и ограничивают собой плоские многоугольные области, необязательно односвязные. Это об-

[1] В стоматологии коронкой зуба называется выступающая над десной часть зуба.

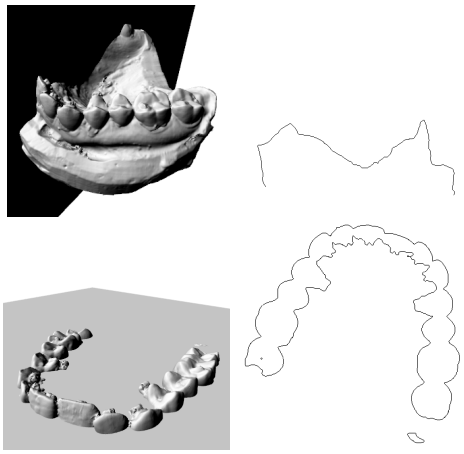


Рис. 2: Сечение формы плоскостью (слева) и соответствующий срез (справа).

стоятельство позволяет использовать различные средства анализа двумерных областей, например, плоские непрерывные скелеты.

3.2 Плоский непрерывный скелет

Пусть Ω — замкнутая плоская область с границей $\partial\Omega$. Пустым кругом области называется круг, полностью содержащийся в ней:

$$\tilde{S}_r(p) = \{q : q \in R^2, dist(p,q) \leq r\} \subset \Omega.$$

Максимальным пустым кругом называется пустой круг, не содержащийся ни в каком другом пустом круге. Наконец, скелетом Ω называется множество центров всех её максимальных кругов (см. [1]). Пример скелета приведен на рис. 4.

Следует отметить, что скелет можно рассматривать как плоский граф, вершинами которого являются центры максимальных пустых кругов, имеющих либо одну общую точку с $\partial\Omega$, либо три и более общих точки, а рёбрами — линии, состоящие из центров тех пустых кругов, которые касаются $\partial\Omega$ ровно в двух точках. Таким образом, скелет области, как и её граница, является одномерным многообразием. Но в отличие от границы это многообразие имеет более сложную структуру — это не набор замкнутых линий без самопересечений, а плоский граф.

Поскольку скелет конструируется на основании множества кругов, вписанных в Ω, с каждой точкой скелета можно связать «ширину» области в этой точке. А именно: радиальной функцией точки скелета называется величина радиуса максимального пустого круга с центром в этой точке.

4. СТРУКТУРА МЕТОДА

4.1 Математическая модель

Ниже предполагается, что трёхмерная система координат задана следующим образом: слепок лежит в плоскости XZ, а зубы направлены в положительном направлении оси Y, независимо от того, нижняя это челюсть или верхняя. При необходимости модель поворачивается в такое положение, при котором указанное требование выполняется. Для этого к координатам вершин модели применяется метод главных компонент: оси найденного таким образом эллипсоида рассеяния задают для модели новый ортонормированный базис, причём ось, вдоль которой рассеяние минимально, совпадает с искомой осью Y.

Для каждого из зубов $\mathscr{T}_k$ можно определить область пространства, содержащую соответствующее ему множество вершин. В описываемом алгоритме строится приближенная аппроксимация этой области Ω_k, представляющая собой пересечениее трёх полупространств, заданных тремя плоскостями $\{\Pi_k^0, \Pi_k^1, \Pi_k^2\}$, которые далее мы будем называть секущими плоскостями зуба. Одна из них отделяет коронку зуба от десны. Мы будем называеть её горизонтальной плоскостью и обозначать через Π_k^0. Две другие плоскости перпендикулярны горизонтальной и отделяют зуб от соседних с ним, такие плоскости назовём вертикальными и обозначим через Π_k^1 и Π_k^2.

При таком представлении нельзя утверждать, что множество всех точек формы, попавших в Ω_k, относится к этому зубу: возможны ситуации, при которых в эту область попадут вершины с противоположной стороны челюсти (см. рис. 3). Поэтому, помимо области пространства, необходимо специальное корректирующее условие принадлежности вершины, учитывающее, на какой стороне челюсти расположена точка . Множество точек пространства, для которых корректирующее условие выполняется, обозначим через C_k. При этом C_k таковы, что

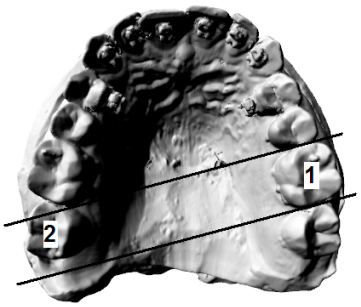


Рис. 3: В область Ω_k зуба 1 попадает часть зуба 2 (плоскости Π_k^1, Π_k^2 изображены прямыми линиями).

$$(\Omega_i \cap C_i) \cap (\Omega_j \cap C_j) = \emptyset$$

при $i \neq j$. Таким образом, принадлежность точки к сегменту каждого зуба определяется конъюнкцией двух условий:

$$\mathscr{T}(v) = \begin{cases} k, \text{если } \exists k, 1 \leq k \leq n : v \in \Omega_k \cap C_k; \\ 0, \text{иначе.} \end{cases}$$

4.2 Определение вертикальных секущих плоскостей

Рассмотрим теперь срезы по горизонтальным плоскостям, проходящим через все зубы и не проходящим через дёсны. Отрезки таких срезов образуют многоугольник, возможно, с некоторыми дополнительными шумовыми отрезками. На рисунке 2 приведен пример именно такого среза.

В этом многоугольнике хорошо заметны чередующиеся широкие и узкие места. Ясно, что широкие участки соответствуют зубам, а узкие промежутки между ними — границам зубов. Отсюда возникает идея провести вертикальные секущие плоскости зубов через узкие участки, перпендикулярно плоскости среза.

Математически умозрительные выводы относительно формы многоугольника в срезе можно описать в терминах непрерывных скелетов (см. рис. 4). Видно, что в

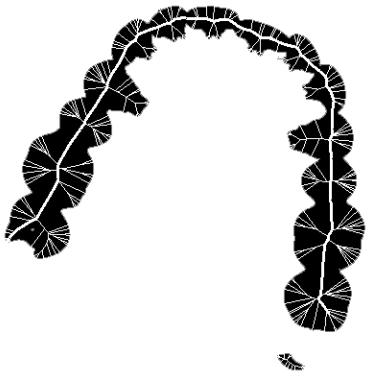


Рис. 4: Скелет для среза с рис. 2. Более толстой линией показана главная ветвь.

соответствующем скелете существует основная, наиболее длинная ветвь, проходящая вдоль всего многоугольника. Её мы также будем называть линией зубов. Сужения и расширения означают, что радиусы максимальных кругов также будут то увеличиваться, то уменьшаться вдоль этой ветви. Поэтому задача определения сужений в срезе сводится к поиску локальных минимумов радиальной функции на главной ветви скелета.

4.2.1 Построение скелета

Как видно, например, из нижнего примера на рис. 2, срез по главной секущей плоскости может содержать несколько областей, которые, к тому же, в общем случае не являются односвязными (это происходит из-за наличия достаточно глубоких углублений в зубах, дно которых находится ниже уровня главной секущей плоскости). Поэтому перед построением скелета необобходимо осуществить фильтрацию внутренних и шумовых внешних контуров. Отбор интересующих областей можно производить на основании содержательных критериев, оценивающих размеры области: площадь, периметр и т.п. После предварительной селекции контуров можно применять алгоритм скелетизации (см. [1]).

4.2.2 Поиск промежутков между зубами

Будем рассматривать скелет как взвешенный граф $G = \{V, E\}$, в котором вес каждого из рёбер равен длине соответствующей линии в скелете. Самый длинный простой путь в этом графе будем называть главной ветвью скелета. На ней определена непрерывная радиальная функция, характеризующая изменение ширины области вдоль линии зубов. Необходимо определить точки, в которых достигаются локальные минимумы этой функции. Одним из возможных способов решения этой задачи является поиск локальных минимумов в дискретной последовательности $\{r_k\}$, состоящей из значений радиальной функции в последовательности вершин главной ветви скелета. Отметим, что для последующего использования полезен не центр найденного круга с локально минимальным радиусом, а хорда c^1, c^2, соединяющая точки касания круга на противоположных сторонах многоугольника. Такую хорду будем называть граничным отрезком.

4.2.3 Определние вертикальных плоскостей и корректирующих правил

На предыдущем шаге мы определили набор граничных отрезков $\{c_k^1, c_k^2\}$ в плоскости $\hat{\Pi}$. Вертикальные секущие плоскости строятся как проходящие через эти отрезки перпендикулярно главной секущей. При этом нормали плоскостей выбираются сонаправленно, т.е. так, что каждый зуб находится в положительном полупространстве плоскости Π_k^1 и отрицательном полупространстве плоскости Π_k^2.

Корректирующие правила, привязывающие вершины к нужным участкам линии зубов, определяются следующим образом. Рассмотрим произвольную точку пространства p и её проекцию на плоскость $\hat{\Pi}$ — точку $\tilde{p}$. Выберем среди середин отрезков $\{c_k^1, c_k^2\}$ ближайшую точку к $\tilde{p}$:

$$\tilde{c} = \arg \min_{c \in \{\frac{c_k^1 + c_k^2}{2}\}} dist(c, \tilde{p}).$$

Корректирующее правило определяется так: $p \in C_k$ тогда и только тогда, когда секущая плоскость, проведенная через содержащий точку $\tilde{c}$ граничный отрезок $\{\tilde{c}_k^1, \tilde{c}_k^2\}$, является одной из двух секущих плоскостей Π_k^1 или Π_k^2, отделяющих k-й зуб.

4.3 Определение главной секущей плоскости

Как написано выше, при определении вертикальных секущих плоскостей предполагается, что плоскость $\hat{\Pi}$ уже задана. Автоматический выбор главной секущей плоскости происходит путём перебора. Для этого, во-первых, определяется множество возможных положений плоскости $\hat{P}$, и, во-вторых, функционал качества $\Psi(\Pi)$, характеризующий набор вертикальных секущих плоскостей, получаемый при выборе Π в качестве главной секущей. Значение функционала вычисляется для каждой из плоскостей $\Pi \in \hat{P}$; та из них, для которой достигается максимум, выбирается в качестве главной секущей:

$$\hat{\Pi} = \arg \max_{\Pi \in \hat{P}} \Psi(\Pi).$$

Множество состоит $\hat{P}$ из заранее заданного в качестве параметра алгоритма (обычно хватает 10–20) количества горизонтальных плоскостей, равномерно рассекающих верхнюю половину модели. Критерий качества $\Psi(\Pi)$ оценивает качество сегментации с позиции здравого смысла и опирается на ряд соображений, таких, как правдоподобное количество обнаруженных зубов (даже если некоторые зубы в данном образце отсутствуют, то промежутки, как правило, можно также сегментировать и посчитать), сопоставимые размеры зубов (естесственная разница в размерах зубов, конечно же, имеет место, но по сравнению с разницей, вызванной ошибочной сегментацией, она незначительна), как можно более высокое положение главной секущей плоскости (т.к. при этом сильнее выражены локальные минимумы радиальной функции на линии зубов, что позволяет точнее определить границы между зубами).

4.4 Определение горизонтальных секущих плоскостей

Для каждой пары вертикальных секущих плоскостей Π_k^1, Π_k^2 можно вычислить их биссекторную плоскость Π_k^{12}. Рассмотрим срез по этой плоскости (см. рис. 5). Если про-

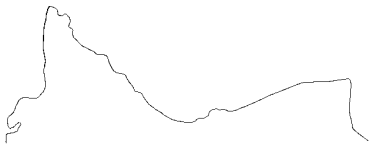


Рис. 5: Срез по плоскости Π_k^{12} для одного из передних зубов.

следить, как меняется кривизна вдоль линии передней стенки зуба, то можно отметить достаточно заметный максимум на границе зуба и десны. Это наблюдение лежит в основе предлагаемого алгоритма нахождения горизонтальной секущей плоскости зуба при условии известной пары вертикальных и главной секущей плоскости.

Для поиска точки наибольшей кривизны используется следующая разностная схема. Передняя стенка в срезе описывается ломаной

$$L = \{(l_x^1, l_y^1), \ldots, (l_x^m, l_y^m)\}.$$

Для каждого отрезка $l^{k-1}l^k$ ломаной можно определить y-компоненту нормали n_y^k к нему. Кривизна кривой в точке оценивается как перепад величины n_y^k в её окрестности.

5. РЕАЛИЗАЦИЯ

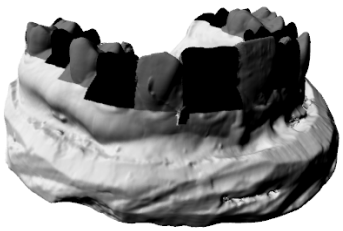


Рис. 6: Результат сегментации.

Алгоритм решения описанной в разделе 2 задачи был реализован и протестирован на выборке, состоящей из 15 различных образцов челюстей. На части образцов, представляющих собой качественные снимки хороших челюстей, полученная сегментация соответсвует ожиданиям (см. пример на рис. 6, где зубы раскрашены в тёмно-серый и чёрный цвета в чередующемся порядке). Недостатки реализации заметны на плохих образцах (отсутствующие зубы, сильно искривлённая челюсть). При этом почти все возникающие ошибки связаны с не очень удачным выбором главной секущей плоскости, что влечёт за собой ошибки в определении границ между отдельными зубами.

Было измерено время выполнения различных операций, осуществляемых алгоритмом. Дольше всего происходит построение множества отрезков среза по заданной плоскости — примерно 0,7 секунды. На втором месте по длительности находится построение скелета — около 0,2 секунды. Остальные операции, такие, как анализ скелетов или отрезков среза, влияют на общую производительность пренебрежимо мало.

6. ВЫВОДЫ

Для поставленной задачи сегментации пространственных форм предложено алгоритмическое решение. С одной стороны, оно опирается на распространённые в данной области подходы (скелеты, поиск выступов); с другой — учитывает специфику и особенности объектов, анализируемых в данной задаче (слабо выраженный характер выступов, некоторая предварительно известная информация об искомых сегментах). Программная реализация алгоритма показала хорошие результаты на качественных образцах и удовлетворительные — на некачественных. Практическая ценность работы заключается в выработке подхода к решению определённого класса задач сегментации, работающих с формами, которые, во-первых, получены в результате измерения реальных, материальных объектов, во-вторых, не обладают ярко выраженными, хорошо различимыми традиционными методами, областями.

7. БЛАГОДАРНОСТИ

Работа осуществлена при поддержке РФФИ по программе 08-01-00670.

8. СПИСОК ЛИТЕРАТУРЫ

[1] Местецкий Л.М. Непрерывная морфология бинарных изображений: фигуры, скелеты, циркуляры. — М.: ФИЗМАТЛИТ, 2009.

[2] D. D. Hoffman and M. Singh. Salience of visual parts. Cognition, 63(1):29–78, April 1997.

[3] Dennie Reniers, Alexandru Telea. Skeleton-based Hierarchical Shape Segmentation. Shape Modeling International, pages 179-188, 2007.

[4] Sebastien Valette, Ioannis Kompatsiaris, Michael G. Strintzis. A polygonal mesh partitioning algorithm based on protrusion conquest for perceptual 3D shape description. Workshop towards Semantic Virtual Environments SVE 2005.

[5] V. A. Knyaz, S. Yu. Zheltov. Photogrammetric techniques for dentistry analysis, planning and visualisation. ISPRS Congress Beijing 2008, Volume XXXVII, Part B5, Commission V.

9. SPATIAL SHAPE SEGMENTATION

In this paper we present a new approach to segmentation of 3D shapes with spatially non-divided segments. The method is based on usage of continuous skeletons build in flat sections of a 3D model. We also consider an application of our algorithm to the problem of segmentation of 3D human's dental model.

Keywords: segmentation, shape, polygonal model, 3D scanning.

Многоклассовая классификация в задаче семантической сегментации

Sobolev Alexander, Barinova Olga
Факультет Вычислительной Математики и Кибернетики
Московский Государственный Университет им. Ломоносова, Москва, Россия
alexandersobolev@yahoo.com

Аннотация

Данная работа посвящена рассмотрению существующих методов многоклассовой классификации на основе бинарного классификатора AdaBoost и усовершенствованию алгоритмически простого подхода «один против всех» с целью улучшения качества его работы в терминах значений ошибки на тестовых наборах данных в задаче семантической сегментации.

Ключевые слова: *многоклассовая классификация, семантическая сегментация, «один против всех», AdaBoost, Platt Scaling.*

1. ВВЕДЕНИЕ

Пусть I – произвольное цифровое изображение, состоящее из пикселей $x \in I$; $\{1, 2, \dots, K\}$ – множество из K классов визуальных объектов (например, $\{\text{Шум}, \text{Земля}, \text{Здание}, \text{Человек}, \text{Небо}\}$). Тогда задача семантической сегментации сводится к построению алгоритма, реализующего отображение $f^*: \{x | x \in I\} \to \{1,2, \dots K\}$, согласующееся с реальным визуальным расположением объектов на изображении. Другими словами, необходимо построить алгоритм, приписывающий каждому пикселю входного изображения одну из предопределенных меток класса согласно реальному нахождению объекта на изображении.

Таким образом, при решении задачи семантической сегментации, самой сегментации (то есть разбиения изображения на односвязные в дискретном смысле непересекающиеся множества пикселей, в объединении составляющие все изображение) может и не происходить. Тогда итоговыми сегментами можно считать наибольшие односвязные в дискретном смысле множества одинаково размеченных пикселей изображения.

Решения задачи семантической сегментации часто используют методы многоклассовой классификации.

Пусть задано K классов с соответствующими метками $\{1, 2, \dots, K\}$, пространство $\mathbb{R}^m$ векторов признаков – числовых описаний объектов и априорно размеченная тренировочная выборка $S \subset \mathbb{R}^m \times \{1,2, \dots, K\}$ – конечное множество известных пар вида $\{\text{Вектор признаков}, \text{Метка класса}\}$. А также, пусть $x \in \mathbb{R}^m$ – некоторый вектор признаков, необязательно принадлежащий тренировочной выборке. Предполагается, что существует некоторое неизвестное отображение $f: \mathbb{R}^m \to \{1,2, \dots K\}$, согласно которому размечена тренировочная выборка S. Тогда задача многоклассовой классификации состоит в построении по тренировочной выборке S алгоритма, реализующего функцию $f^*: \mathbb{R}^m \to \{1,2, \dots K\}$ такую, чтобы $E([\![f^*(x) \neq f(x)]\!])$ было минимальным. Стоит отметить, что x в данном случае не обязательно должен быть из тренировочной выборки S. Другими словами по тренировочной выборке необходимо построить функцию f^*, обобщающую неизвестную зависимость векторов признаков и меток классов из S.

В данной статье в разделе 2 приводится обзор методов многоклассовой классификации, основанных на идее усиления бинарных гипотез AdaBoost [1]. В разделе 3 рассмотрен предложенный метод многоклассовой классификации и проведена оценка сложности классификации рассмотренными алгоритмами. В разделе 4 приведено экспериментальное сравнение рассмотренных методов на примере работы в реальной системе семантической сегментации.

2. ОБЗОР МЕТОДОВ МНОГОКЛАССОВОЙ КЛАССИФИКАЦИИ

Данный обзор фокусируется на методах разрешения проблем многоклассового машинного обучения, основанных на идее усиления слабых гипотез AdaBoost [1].

Большинство существующих методов многокласовой классификации либо базируются на бинарных классификаторах, либо к ним сводятся. Общая идея такого сведения заключается в использовании набора бинарных классификаторов (не обязательно AdaBoost), обученных разделять различные группы объектов друг от друга. При классификации используются различные схемы голосования.

В общем, все современные многоклассовые методы машинного обучения можно подразбить на 3 группы:

1. Элементарные – простые подходы, использующие набор бинарных классификаторов «один против всех» и «каждый против каждого» [2];
2. Классические – методы, сводимые к группе элементарных подходов (AdaBoost.MH, AdaBoost.M2, AdaBoost.MR [1]);
3. Методы, основанные на кодах с коррекцией ошибки (ECC – error correcting code). К ним относятся: AdaBoost.ECC [3], AdaBoost.ERP [4], AdaBoost.ERC [5].

Метод «один против всех» состоит в построении k бинарных классификаторов h_i, каждый из которых обучается на тренировочной базе вида $\{(x, 1) | y = i, (x, y) \in S\} \cup \{(x, -1) | y \neq i, (x, y) \in S\}$, $i = 1, k$, где i – порядковый номер бинарного классификатора. Классификация произвольного вектора признаков $x \in \mathbb{R}^m$ определяется голосованием: $\mathcal{F}(x) = \text{argmax}_i(h_i(x))$.

В статье «Statistical view of boosting» [1] показано, что группа «классических» методов сводится к методу «один против всех».

Основной идеей методов, основанных на самокорректирующихся кодах, состоит в кодировке меток классов уникальными бинарными словами одинаковой

длины, которые образуют кодирующую матрицу M (ее строки совпадают с кодирующими словами). Далее для каждого столбца строится бинарный классификатор, умеющий отделять классы с соответствующим значением в столбце, равным единице, от классов с нулем.

В сравнении классификаторов AdaBoost.ECC, ERP и ERC [5] было показано, что AdaBoost.ERP и ERC – справляются с задачей классификации значительно лучше, чем классический метод «один против всех».

3. ОПИСАНИЕ ПРЕДЛОЖЕННОГО МЕТОДА

Для решения поставленной задачи проанализируем простейший метод «один против всех» и усовершенствуем его, не сильно усложнив процесс классификации.

3.1 Улучшение метода «один против всех»

Классификация в методе «один против всех» происходит следующим образом: $\mathcal{F}(\mathrm{x}) = \mathrm{argmax_i}(\mathrm{h_i}(\mathrm{x}))$. Но из схемы настройки классификатора ясно, что все бинарные классификаторы $\mathrm{h_i}(\mathrm{x})$ настраиваются каждый независимо друг от друга.

Бинарные классификаторы на выход выдают некоторый аналог расстояния до построенной ими разделяющей поверхности (так, в случае слабых учеников – порогов по одной из компонент векторов признаков – выход бинарного Gentle AdaBoost соответствует L1-расстоянию до разделяющей классы поверхности), а не точную оценку вероятности принадлежности вектора x 1-ому классу.

Таким образом, в методе «один против всех» бинарные классификаторы получаются несогласованными, и встает вопрос о «согласовании» их выходов.

Более того, в подходе «один против всех» встает проблема несбалансированности классов. Так, даже если изначально задача многоклассовой классификации сбалансирована (то есть имеет одинаковое число тренировочных прецедентов каждого класса), то при обучении бинарного классификатора прецедентов с меткой 1 будет $\mathrm{N/K}$, а прецедентов с меткой -1 будет $\mathrm{N(N-1)/K}$. Таким образом, отношение количества прецедентов в каждой бинарной задаче будет равно $(\mathrm{N}-1)$, а оно увеличивается с ростом числа классов, что будет особенно сильно сказываться на задаче семантической сегментации с большим числом классов.

Существует метод масштабирования или шкалирования выходов $\mathcal{F}(\mathrm{x})$ уже настроенных бинарных классификаторов – Platt Scaling [6]. Он преобразовывает выходы бинарных классификаторов в оценку вероятности принадлежности прецедента x к 1-ому классу, т.е. в оценку $P(1|x)$. Для работы самого метода необходима так называемая валидационная выборка (для нее априорно известны метки классов и при этом классификатор не настраивался на ней). Ее часто создают из обучающей выборки, немного сокращая последнюю.

Улучшенный метод «один против всех»

Задача с K классами, обучающей выборкой D.

Для каждого класса $\mathrm{k} = \overline{1,\mathrm{K}}$ выполнить:

1. Случайным образом проредить или продублировать данные D_k вида $D_k = \{(x,1)|y = k, (x,y) \in D\} \cup \{(x,-1)|y \neq k, (x,y) \in D\}$
2. Выделить валидационную выборку V_k, как часть D_k
3. Настроить бинарный классификатор h_k на данных $D_k \backslash V_k$
4. Получить с помощью Platt Scaling сигмоиду $S_k = \{\mathrm{A_k}, \mathrm{B_k}\}$ по выходам классификатора h_k на данных V_k

Выход классификатора:

$$argmax_k S_k(h_k(x)) = argmax_k \frac{1}{1 + e^{A_k h_k(x) + B_k}}$$

Алгоритм 1: *улучшенный метод «один против всех».*

Для борьбы с несбалансированностью классов можно использовать простейшие методы случайного прореживания или дублирования данных [7][8].

Улучшим метод «один против всех», определив схему работы согласно Алгоритму 1 (добавив шкалирование по платту к каждому бинарному классификатору и проредим или продублируем данные для решения проблемы несбалансированности классов).

3.2 Оценка сложности классификации различными методами

Пусть все бинарные классификаторы многоклассовых классификаторов имеют тип Gentle AdaBoost и используют фиксированное число T одинаковых слабых учеников. Пусть также число классов равно K. Тогда сложность классификации будем оценивать по числу необходимого количества слабых учеников для работы методов.

Оценим сложность классификации полученным методом в выше определенных терминах. Она составляет $\mathrm{O(K} * (2 \text{ операции произведения} + 2 \text{ суммирования} + \text{одно возведение в степень} + \mathrm{T}) = \mathrm{O\big(K(T + const)\big)} = \mathrm{O(TK)}$.

Сравним сложности классификации методом AdaBoost.ECC/ERP и «один против всех».

Утверждение 1 – о сложности классификации методом AdaBoost.ECC/ERP

В общем случае классификация алгоритмом «один против всех» как минимум не сложнее классификации методом AdaBoost.ECC/ERP.

Доказательство.

Сведем классификацию методом «один против всех» к классификации методом AdaBoost.ECC/ERP, чем докажем поставленное утверждение.

Пусть есть конкретный классификатор на основе метода «один против всех». Пусть он состоит из K бинарных классификаторов $\mathrm{h_i}$, $\mathrm{i} = \overline{1,\mathrm{K}}$ и с T слабыми учениками каждый.

Построим кодирующую матрицу M классификатора AdaBoost.ECC следующим образом: $\mathrm{M} = 2 * \mathrm{E^{K\times K}} - 1$, где $\mathrm{E^{K\times K}}$- единичная квадратная матрица размерами K × K. Таким образом, кодирующая матрица M представляет собой квадратную матрицу, на главной диагонали которой находятся единицы, а в остальных ячейках – минус единицы.

Определим бинарные классификаторы метода AdaBoost.ECC как $\mathrm{f_t} = \mathrm{h_t}, \mathrm{t} = \overline{1,\mathrm{K}}$, а веса $\alpha_\mathrm{t} = 1, \mathrm{t} = \overline{1,\mathrm{K}}$. Тогда процесс классификации упростится:

$$\operatorname{argmin}_k \Delta\left(M(k), \mathcal{F}(x)\right) = \sum_{t=1}^{T} \alpha_t * \frac{1 - M(k,t) f_t(x)}{2}$$

$$= \frac{1}{2}\left\{K - \sum_{t=1}^{K} M(k,t) h_t(x)\right\} \leftrightarrow$$

$$\leftrightarrow \operatorname{argmax}_k \left\{\sum_{t=1}^{K} M(k,t) h_t(x)\right\}. \quad (*)$$

Не ограничивая общности, пусть $h_1(x) > h_2(x), .. h_K(x)$. Тогда в задаче «один против всех» решением является 1 класс. Рассмотрим построенный классификатор AdaBoost.ECC. Сравним значение $(*)$ при $k = 1$ с произвольным $k = i$.

$$h_1 - h_2 - .. - h_i - .. - h_K \bigvee -h_1 - .. + h_i - .. - h_K,$$

$$2h_1 \bigvee 2h_i,$$

$$h_1 > h_i, \forall i \neq 1$$

Таким образом, решение конкретной задачи «один против всех» является решением построенной задачи AdaBoost.ECC ($\operatorname{argmax}_k = 1$) Но все указанные рассуждения справедливы и в обратном направлении – решение построенной задачи AdaBoost.ECC является решением задачи «один против всех».

Согласно утверждению о сложности алгоритмов из теории математической оптимизации, получаем доказательство представленного утверждения (классификация «один против всех» корректно сводится к классификации AdaBoost.ECC, что говорит о том, что задача классификации AdaBoost.ECC как минимум не проще метода «один против всех»).

Сложность алгоритмов классификации в указанных терминах приведена в Таблице 1.

Таблица 1 – сравнение сложности классификации различными многоклассовыми методами

Метод	Сложность классификации
«Один против всех» и улучшенный «один против всех»	$O(TK)$
AdaBoost.M1, M2, MH и др.	$\sim O(TK)$
«Каждый против каждого»	$O(TK^2)$
AdaBoost.ECC/ERP	Не проще, чем $O(TK)$

Таким образом, был получен новый метод многоклассовой классификации, незначительно отличающийся по алгоритмической сложности классификации от простого метода «один против всех».

4. ЭКСПЕРИМЕНТЫ

Сравним полученный метод с рассмотренными алгоритмами многоклассовой классификации.

Оценкой качества работы рассмотренных методов будем считать значение ошибки построенных классификаторов на тестовой выборке со скользящим окном в зависимости от числа слабых учеников, входящих в классификатор. Для семантической сегментации под уровнем ошибки подразумевается попиксельная ошибка на тестовой выборке размеченных изображений.

В качестве тестовой системы семантической сегментации был выбран упрощенный вариант системы TextonBoost [9] (попиксельная система семантической сегментации).

В качестве тренировочных и тестовых изображений были взяты базы Sowerby [10] – 5 классов и MSRC21 [11] – 21 класс. Эти базы являются одними из наиболее распространенных для сравнения систем семантической сегментации. В результирующих изображениях все обученные классификаторы состоят из одинакового числа слабых учеников (рис. 1,2).

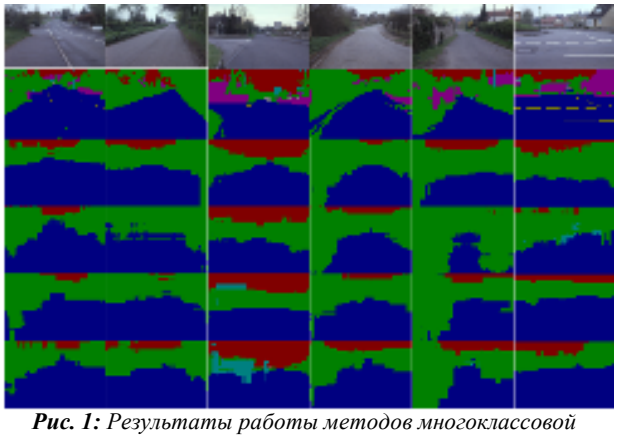


Рис. 1: *Результаты работы методов многоклассовой классификации в системе TextonBoost, база Sowerby. Сверху вниз – оригинал изображения, ручная разметка, JointBoost в реализации авторов TextonBoost, AdaBoost.ERP, «один против всех» и собственный улучшенный «один против всех».*

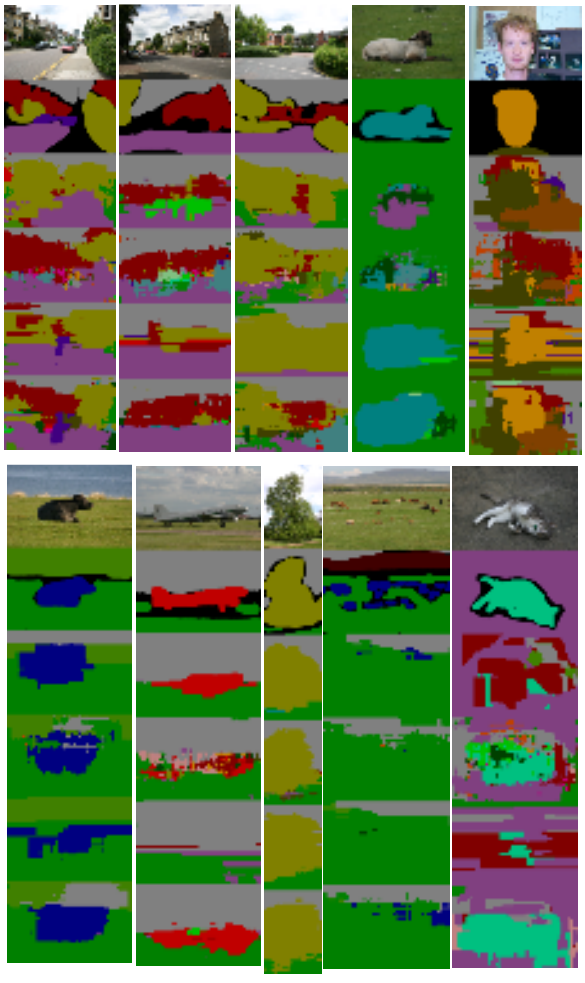


Рисунок 2: *Результаты работы методов многоклассовой классификации в системе TextonBoost, база MSRC21. Сверху вниз – оригинал изображения, ручная разметка, JointBoost в реализации авторов TextonBoost, AdaBoost.ERP, «один против всех» и собственный улучшенный «один против всех». Области черного цвета (в ручной разметке) не учитываются при подсчете ошибки классификации.*

Из результатов видно, что собственный подход на базе метода «один против всех» значительно улучшил работу классического «один против всех». Он стал чаще находить ранее не найденные классы и точнее выделять области. При этом собственный подход иногда работает даже лучше таких сложных методов как JointBoost (в реализации авторов TextonBoost) и adaboost.ERP.

Также целесообразно рассмотреть фактический уровень попиксельной ошибки работы TextonBoost с рассмотренными классификаторами на этих базах (рис. 3).

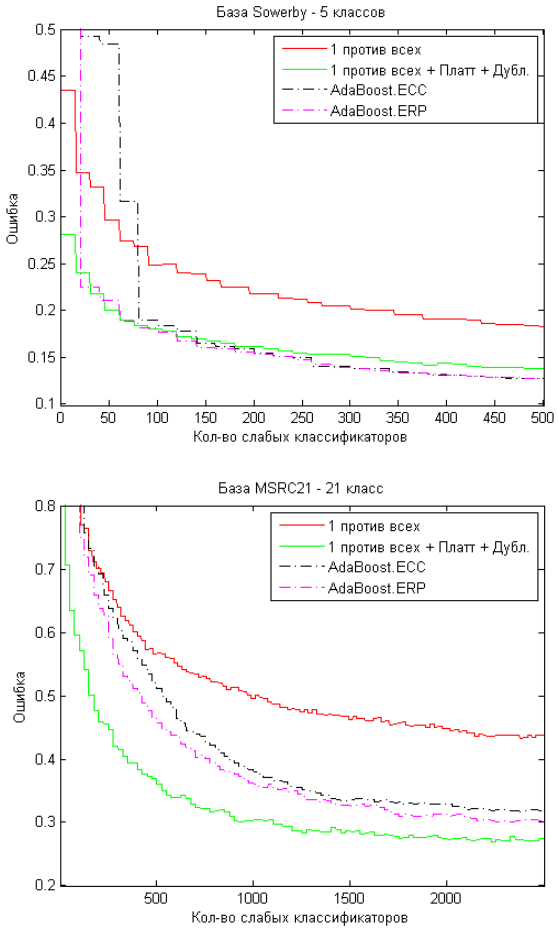


Рисунок 3: *Уровень попиксельной ошибки классификаторов на тестовой выборке при работе с TextonBoost на соответствующих базах.*

5. ВЫВОДЫ

В результате улучшения метода многоклассовой классификации «один против всех» был получен новый, более точный метод при незначительном изменении алгоритмической сложности классификации (она наименьшая среди рассмотренных методов). Как показали проведенные эксперименты, он сравним по качеству работы с такими сложными методами, как AdaBoost.ERP и JointBoost.

Вне рамок данной работы было проведено сравнение построенного метода с рассмотренными алгоритмами многоклассовой классификации на базах данных репозитория UCI [12]. В этих экспериментах улучшенный метод «один против всех» также показал сравнимое качество со сложными методами Adaboost.ERP/ERC.

6. ССЫЛКИ

[1] Jerome Friedman, Trevor Hastie, Robert Tibshirani *Additive logistic regression: a statistical view of boosting* // Annals of Statistics. 2000. Volume 28. № 2. P. 337-407

[2] Ryan Rifkin, Aldebaro Klautau *In Defense of One-Vs-All Classification* // The Journal of Machine Learning Research. 2004. Volume 4. P. 101-141

[3] Venkatesan Guruswami, Amit Sahai *Multiclass learning, boosting, and error-correcting codes* // Proceedings of the twelfth annual conference on Computational learning theory. 1999. P. 145-155

[4] Ling Li *Multiclass boosting with repartitioning* // Proceedings of the 23rd international conference on Machine learning. 2006. P. 569-576

[5] Chung-Chih Lin, Yuh-Show Tsai, Yu-Shi Lin, Tai-Yu Chiu, Chia-Cheng Hsiung, May-I. Lee, Jeremy C. Simpson, Chun-Nan Hsu *Boosting multiclass learning with repeating codes and weak detectors for protein subcellular localization* // Bioinformatics. 2007. Volume 23. № 24. P. 3374-3381

[6] Alexandru Niculescu-mizil, Rich Caruana *Obtaining Calibrated Probabilities from Boosting* // In proceedings of the 21st Conference on Uncertainty in Artificial Intelligence. 2005

[7] А. Кузьмишкина, О. Баринова, А. Конушин *Выбор объектов для обучения в условиях сильной несбалансированности классов* // Труды конференции Graphicon'2008. 2008. стр. 213-216

[8] Jason Van Hulse, Taghi M. Khoshgoftaar, Amri Napolitano *Experimental perspectives on learning from imbalanced data* // Proceedings of the 24th international conference on Machine learning. 2007. Volume 227. P. 935 – 942

[9] J. Shotton, J. Winn, C. Rother, A. Criminisi *Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation* // Proceedings of European Conference of Computer Vision. 2006. Volume 3951. P. 1-15

[10] D. Collins, W. A. Wright *The Sowerby Image Database* // Seventh International Conference on Image Processing And Its Applications. 1999. Volume 1. P. 306 - 310

[11] MSR Cambridge *Pixel-wise labelled image database* [HTML] (http://research.microsoft.com/en-us/um/people/antcrim/data_objrec/msrc_objcategimagedatabase_v2.zip)

[12] UCI Machine Learning Repository [HTML] (http://archive.ics.uci.edu/ml/)

About the authors

Sobolev Alexander has graduated from Lomonosov Moscow State University in 2009; Barinova Olga is a junior researcher in Graphics&Media Lab at Lomonosov Moscow State University.

Проволочная модель бедра и её использование для планирования остеотомии

Василий Гончаренко[1], Вячеслав Архипов[1], Александр Тузиков[1], Олег Кривонос[2], Олег Соколовский[3]
[1]Объединённый институт проблем информатики НАН Беларуси, Минск, Беларусь;
[2]Институт вычислительной медицины, Мангейм, Германия;
[3]Республиканский научно-практический центр травматологии и ортопедии, Минск, Беларусь.

Резюме

В статье приводится описание проволочной модели бедра, которая используется при планировании косой и ротационной остеотомии в системе компьютерной поддержки планирования лечения в ортопедии и травматологии. Эта система использует в качестве исходных данных результаты КТ обследования. Она даёт пользователю возможность сегментации и визуализации объектов, а также проведения измерений анатомических параметров этих объектов. Также, с помощью этой системы можно планировать и проводить виртуально (моделировать) операции по коррекции анатомической формы костей: виртуальные остеотомию и остеосинтез.

Ключевые слова: *система поддержки планирования операций, обработка изображений КТ, моделирование операций.*

1. ВВЕДЕНИЕ

Одной из основных задач ортопедии является лечение деформаций и нарушений функций костно-мышечной системы. Для диагностики этих деформаций используются методы измерения параметров, характеризующих форму костей и величину отклонения от нормы, показывающую степень и характер деформаций. Форма кости может быть описана путем задания последовательности точек и последующем измерении углов, расстояний и других геометрических параметров с их использованием. Так как методика измерений принципиально не отличается для различных костей, приведем оценку геометрических параметров на примере диагностики и лечения деформаций тазобедренного сустава [1, 2].

2. АНАТОМИЧЕСКАЯ СИСТЕМА КООРДИНАТ

Для того чтобы иметь возможность оценки параметров бедра и устанавливать количественные отношения между любыми точками в пределах бедра, необходимо задать систему координат. Анатомическая система координат бедра устанавливается следующим образом. Центр координат находится в точке *CORA* (точка *O* на рисунке 1). Ось *Oz* совпадает с проксимальной осью диафиза бедра (*CO* на рисунке 1) - она задает вертикальное (краниальное) направление. Ось *Ox* является проекцией оси мыщелков (CC_1 на рисунке 1) на плоскость, ортогональную оси *Oz* и проходящую через центр координат, - она задает латерально-медиальное направление. Ось *Oy* ортогональна осям *Ox* и *Oz* и проходит таким образом, что тройка *Ox*, *Oy*, *Oz* является правой. Ориентация векторов в анатомической системе координат задается путем указания двух углов:

1) между прямой OT и её проекцией на фронтальную плоскость (φ на рисунке 1);
2) между прямой OT и её проекцией на сагиттальнуюплоскость (ψ на рисунке 1).

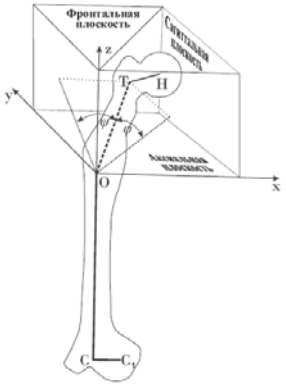


Рисунок 1: Анатомическая система координат бедра.

3. ПРОВОЛОЧНАЯ МОДЕЛЬ БЕДРА

Форма бедра (в той мере, которая достаточна для измерения анатомических углов) полностью определяется анатомическими ориентирами, определяющими головку бедра, ось шейки бедра, обе оси диафиза, а также ось мыщелков. Если последовательно соединить все эти ориентиры, то получится так называемая проволочная модель бедра, которая несет в себе положение всех анатомических ориентиров в системе координат бедра (рисунок 1). Такая модель данных является компактной, понятной с точки зрения геометрии бедра и удобной для анализа степени патологии, принятия решения о необходимости остеотомии и вычисления её параметров.

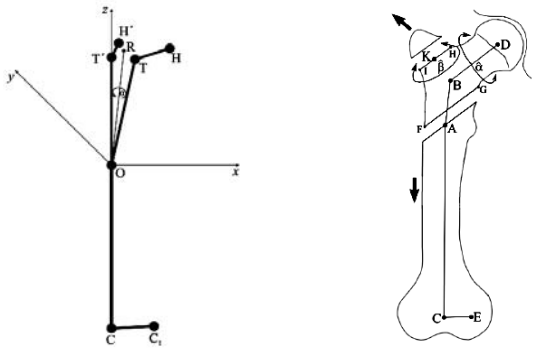
4. МОДЕЛИРОВАНИЕ ОПЕРАЦИИ

Различные типы операций требуют анализа различных анатомических параметров. Большинство операций на бедре требуют коррекции различных геометрических характеристик объектов, описывающих деформацию, путем приведения этих характеристик в диапазон наиболее приемлемых значений. Указанные характеристики обычно составляют положение набора анатомических ориентиров,

углы между прямыми, проходящими через анатомические ориентиры, а также другие более сложные атрибуты. В большинстве случаев основную информацию об объекте планирования можно получить из анализа проволочной модели бедра. В более сложных случаях требуется анализ формы части поверхности как самой кости, так и ее соседей. В этом случае необходимо выделить контур кости и оценить положение некоторой последовательности точек, лежащих на этом контуре.

4.1 Косая остеотомия

Вначале хирург помечает 5 анатомических ориентиров для определения формы кости и вида возможных деформаций (*H*, *T*, *O*, *C*, C_1 на рисунке 2ф). Затем производится анализ двух параметров: шеечно-диафизарного угла (угол между векторами *TH* и *CO*) и угла торсии бедра (угол между проекциями осей *TH* и CC_1 на плоскость, ортогональную вектору *CO*). Если шеечно-диафизарный угол находится вне заданных пределов τ_1, τ_2 и/или угол торсии превышает заданный порог β, тогда необходимо выполнить коррекцию деформаций путем остеотомии. При этом положение плоскости разрезания кости соответствует точке *O* (рисунок 2а). Ориентация этой плоскости и угол поворота бедра α относительно неё определяется автоматически на основании решения задачи двумерной оптимизации.



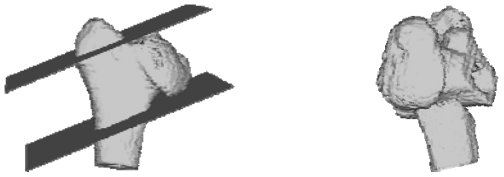
а) б)

Рисунок 2: а) косая остеотомия (проволочная модель); б) ротационная остеотомия бедра.

4.2 Ротационная остеотомия

Эта операция выполняется, когда необходимо скорректировать не только угол торсии бедра, но и шеечно-диафизарный угол. Как и в случае с косой остеотомией, хирург определяет базовую плоскость, а также структуру деформаций бедра путем задания шести анатомических ориентиров (B, D, A, C, E, K на рисунке 2б). Положение плоскости разреза определяется ориентиром A. Вычисление ориентации этой плоскости может быть сформулировано в виде оптимизационной задачи. Она вычисляется таким образом, чтобы после двух вращений на углы $\hat{\alpha}$ (вокруг оси шейки бедра) и $\hat{\beta}$ (вокруг нормали плоскости разреза) разрезанной части бедра (FGHI на рисунке 2б), модель перешла в наиболее приемлемое состояние, описываемое шеечно-диафизарным углом (угол между прямыми AC и BD на рисунке 2б) и углом торсии шейки бедра, а также, чтобы верхняя часть головки бедра имела поверхность, наиболее совпадающую с поверхностью вертлужной впадины таза, в которой она анатомически расположена. Плоскость разреза большого вертела бедренной кости (HI на рисунке 2б) проходит параллельно плоскости разреза FG и необходима для того, чтобы выполнить вращение вокруг оси шейки бедра. Ее положение определяется вручную заданием анатомического ориентира K в основании большого вертела. Угол $\hat{\alpha}$ вычисляется таким образом, чтобы верхняя часть головки бедра (направление «верх» совпадает с осью Oz системы координат бедра) имела бы форму поверхности, наиболее совпадающую с соприкасающейся с ней частью поверхности вертлужной впадины таза.

На рисунке 3а показаны результаты планирования ротационной остеотомии и плоскость разрезания. На рисунке 3б показана повернутая часть кости на 90o в заднем направлении с уменьшением шеечно-диафизарного угла на 10^o с использованием вышеприведенной оптимизационной процедуры.



а) б)

Рисунок 3: Результат планирования (а) и проведения (б) виртуальной остеотомии.

5. ЗАКЛЮЧЕНИЕ

Одной из проблем планирования хирургических операций является точность определения их количественных параметров. При идеальном расчете хирург должен получить цифровую информацию, например, в градусах или миллиметрах, о степени необходимого перемещения вертлужной впадины в каждом из трех направлений. Исход такой операции во многом зависит от степени коррекции, то есть степени и направления перемещения объекта оперирования, в каждой из трех плоскостей. Разработанное программное обеспечение позволяет вычислять параметры коррекции деформаций в рамках планирования операций косой и ротационной остеотомий бедра с учетом трехмерной структуры суставов и, следовательно, минимизировать возможные ошибки на этапе планирования.

6. ЛИТЕРАТУРНЫЕ ИСТОЧНИКИ

[1] Krivonos O., Hesser J., Maenner R., Keppler P., Gebhard F., Kinzl L., Sakalouski A.A., Sakalouski A.M. Precise computer aided correction of bone deformities // Proc. of the World Congress on Medical Physics and Biomedical Engineering, Sydney, Australia, 2003.

[2] Hancharenka V., Tuzikov A., Arkhipau V., Kryvanos A. Preoperative planning of pelvic and lower limbs surgery by CT image processing // Proceedings of 8th International Conference on Pattern Recognition and Image Analysis (PRIA'2007), Yoshkar-Ola, Russia, 2007, pp. 270-274.

3D ELECTRONIC MODELING OF ARCHITECTURAL PARK COMPLEX OF COASTAL AREA IN *SARAPUL*

Sergey N. Zykov, Elena V. Ovchinnikova
Department of Design of Industrial Products
The Udmurtian State University, Izhevsk, Russia
E-mail: luka_sz@inbox.ru, evladi@list.ru

Abstract: In the article herein the application of cultural and historic heritage as image in environmental design was considered, as well as a usage of up-to-date tools for 3D computer-aided modeling while designing electronic model of architectural park complex was reviewed.

Keywords: *Russian wood architecture, architectural park complex, CAD systems, 3D geometrical modeling.*

Ever-growing development pressure is one of the main factors having an adverse effect on mind of modern urban man. Animate nature areas such as public gardens, avenues, sports-grounds, etc. arranged in the city assist in cushioning of adverse impact [2]. Aesthetic design solution of this environmental space may be found in cultural and historical heritage of the cities having wood architecture traditions. Natural materials and warm true colors of wood constructions create an atmosphere favoring calm repose. In addition, it should be noted that preservation of historical and cultural image of wood Russia within space designed contributes to the matter of great concern, i.e. spiritual renewal of Russia.

Long before stone constructions appeared, a lot of architectural forms were designed of wood as the most available construction material (Figure 1). It was wood architecture where many building and composite techniques, which satisfied natural and climatic conditions and artistic and aesthetic sense of people, were worked out [1, 4]. In order to determine dimensions of future constructions Russian carpenters developed a system of measurements which was closely related to average dimensions of human body. Speaking a language of modern design we may say that all architectural forms of Russian wood constructions met ergonomics requirements. Later on methods of proportions worked out in wood architecture were passed into stone one [5].

Modern procedures of wood architecture research and use of the research results in contemporary environmental design is unconceivable without active application of 3D geometrical modeling software. High-grade 3D electronic rendering of environmental objects provides at any stage means for making an assessment and introducing timely changes in space arrangement with due account of unique color and historical distinction of its location. Moreover, it is a matter of no little significance that electronic geometrical models of environmental objects may form primary initial data for issuance of technical documentation in practical implementation of the projects. Let us consider an example of design concept of a part of *Sarapul* quay.

Sarapul is one of the oldest towns at *Kama* river. In the description of *Sarapul*'s emblem there is the phrase as follows: "In silvery field upon a high mountain there is a wooden log town thanks to which this place is worthy of notice" [3]. Situated on the side of *Startseva* mountain *Sarapul* was built as defensive fortification at *Kama* river. These are quays that were a face of the town. Taking it into consideration a special concept was developed for arranging recreation area at quay using wood architectural composition.

Figure 1: Watchtower in *Belskoe* village (end of XVII century)

In this case selection of software for 3D electronic modeling was determined due to composition highly packed with elements of wood architecture. This assumes a great number of small structural elements such as trusses, bars with various section, boards, carved balusters, etc. that required to use software with good tool sets for geometrical construction. Hard CAD systems suit well these purposes, since they have powerful toolset for parametric modeling and photorealistic imaging of proper quality.

The concept of front-end engineering design consists of related set of environmental objects according to selected style of Russian wood architecture (Figure 2).

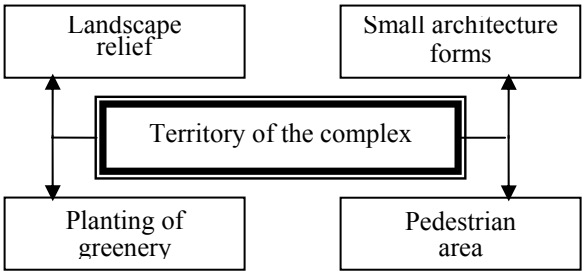


Figure 2: Concept of front-end engineering design

Expressiveness of overall environmental complex shaping is provided by stepped land form, alteration of high and low volumes, roof of various constructions and arrangement of the objects to be designed at different height.

Landscape relief.

The landscape features artificially created relief with three-level difference reinforced by retention walls with radial-wise lines. Between them there is an artificial pond with curved

wood deck, benches, carved bridges and steps at the relief difference.

Pedestrian area.

Away from entrance lobby there are two parallel avenues paved with cubes. Paths are curvilinear; they skirt a group of trees providing continuous change in distant views of watery waste.

Planting of greenery.

Trees are arranged by groups by two to three plants of different age. This structure of planting does not interfere with a view of water pool; moreover, it enchases and accentuates opening-up views and forms open glades as well.

Small architecture forms.

Small architecture forms in design concept as proposed are open space patterns of different species of wood: pier, light shaded shelter, above-water watchtowers and arbors (Figure 3).

Entrance lobby is arranged in the form of open framework with carved fence.

After 3D electronic model of landscape and architectural park complex had been completely developed, some photorealistic images were generated. Application of specialized software such as 3D-MAX for these purposes in the project was handicapped by great number of small modular elements. Therefore, the images were generated within CAD system software environment that considerably sped up the project to be worked upon (Figure 4). Moreover, a photorealistic image of satisfactory quality was made.

Figure 3: Small architecture forms of the project

The project of architectural park complex of coastal area in *Sarapul* as proposed above is an example of build up design concept which assumes combination of historical and ethnocultual features of the *Kama* region, wood architecture traditions and modern computer aids for 3D virtual modeling.

Figure 4: Photorealistic image of design concept of architectural park complex in *Sarapul*. View from water pool

References

1. Bubnov E.N. Russian wood architecture of Ural region. – Moscow: Stroiizdat, 1988. – p. 188.
2. Gorokhov V.A. Urban greenery. – Moscow: Architecture-C, 2005. – p. 589.
3. History of Udmurtia within XV - XX centuries. – The Udmurtian Institute of History, Language and Literature, Ural branch of RAS Izhevsk, 2007. – p. 544.
4. Koshaev V.B. Traditional dwelling of Western Transural region nations. Cultural genesis. Classification. Art. Izhevsk: The Udmurtian University, 2001. – p. 370.
5. Krasovski M. Encyclopedia of Russian architecture. Wood architecture. – Saint-Petersburg: Satis, 2005. – p. 382.
6. Sarapul within 1596 - 1985. Documentation and materials. – Izhevsk, 1987. – p. 378.

Cognitive Computer Graphics Based on n-m Multiterminal Networks for Pattern Recognition in Applied Intelligent Systems

Anna Yankovskaya
Laboratory of Intelligent Systems, Tomsk State University of Architecture and Building, Russia
ayyankov@gmail.com
Dmitry Galkin
Department of Humanities and Informatics, Tomsk State University, Russia
gdv_t@mail.ru

Abstract

Authors introduce original approach to design of cognitive computer graphics for applied intelligent systems (IS). Concept of "n-m multiterminal network" (N-MMN) is applied in novel way to knowledge structuring and cognitive representation.

Keywords: *cognitive computer graphics, intelligent systems, pattern recognition, n-m multiterminal networks.*

1. INTRODUCTION

Use of cognitive computer graphics (CCG) for intelligent systems (IS) is one of the key tools for object localization and recognition. For our purposes we accept definition of CCG as method to visualize data and represent knowledge that allows generating new decision or finding a way (prompting) to the new one [Зенкин]. CCG image is a tool that gives understanding of task or situation (image-decision), representation of IS-generated decision (decision-image), visualization of objects that helps to set task (image-task) [1]. CCG considered as crucial component for representation of knowledge as well as operation of it in IS, including decision-making justification. In this paper authors present CCG development for current work in progress Intelligent Instrumental Software (IIS) IMSLOG [2], which is later development of CCG for IS described in [3] and belongs to tradition of pattern recognition research (pattern recognition is a set of principals and rules for designing systems able to determine if given object belongs to the one of pre-defined classes). In particular, we address very complex problem of IS based medical diagnostic, where, for example, symptoms of decease would be an object and diagnosis would be a class.

This particular article is inspired by Global Risk Report 2009 prepared and published for World Economic Forum [4]. In this document experts use cognitive graphics as an instrument for analysis and representation of complex knowledge. They identified 36 six global risks under five categories – economic, geopolitical, environmental, societal, technological. All risks are interconnected and resulting image gives representation of global risks as interconnected network (see Figure 1). This model gives sense of structure, complexity, visual coherence, conceptual approach (network dynamic interconnection), concentration on content, easy focusing on key elements, generalized and unified complex data, analytical convenience and usability. However, for computer based IS we need adequate principles of structuring and multivariate analysis of cognitive maps similar to one presented at Figure 1.

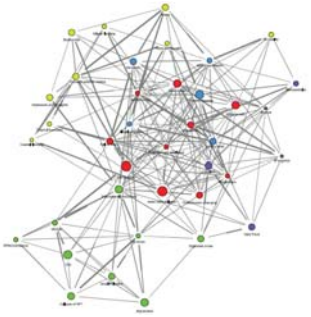


Figure 1: *Risk Interconnection Map (RIM) 2009 (Source: World Economic Forum).*

We propose original approach based on N-M multiterminal networks (N-MMN).

2. N-M MULTITERMINAL NETWORKS

"N-m multiterminal networks" (N-MMT) concept comes from radiotechnics and logic devices design. It means element of electric chain with several connections with other chains. As a general principal it is conveniently described as N-inputs and M-outputs for the object given as "black box". Any type of objects can be represented as n-m multiterminal network and analyzed with multivariate combinations of inputs and outputs. In our current project IIS IMSLOG we approach to use N-MMT for designing typical structural templates of applied IS (scenario, typical decision based on different algorithms) that appears for user as a separate object presented as N-MMT (Figure 2).

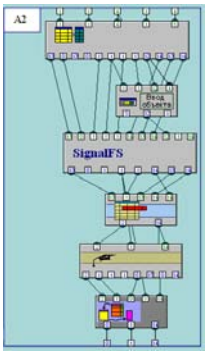


Figure 2: *Cognitive visualization of structural template (typical decision for algorithm of signal features revealing) as N_MMT for IIS IMSLOG.*

Our research and graphic modeling shows that we can easily find N-MMN equivalent representation for different typical decisions,

such as sequential recoding of features, signal features revealing algorithm, implication matrix construction etc. Decision-making is based on original logic-combinatorial and logical-combinatorial-probabilistic test methods of pattern recognition [5]. In case of medical diagnostic, inputs are symptoms and syndromes; output are syndrome or diagnosis (if output of N-MMN is not an input for another one). CCG for diagnostic decision-making will be looking like following (Figure 3).

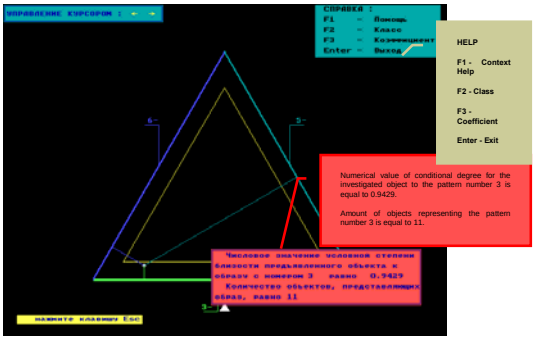


Figure 3: *CCG for medical diagnostic decision-making (ISS IMSLOG).*

Illustration reflects dimensional arrangement of object (patient own diagnosis) with respect to any 3 patterns (differential diagnosis). Side of triangle is one of the patterns. Position of white point (patient diagnosis) informs about diagnosis nearness to distinct disease (side) and error of decision made (distance to internal triangle). Thus, CCG for N-MMN provides us with knowledge representation for decision-making process itself, and triangle model gives us a structure of final decision. If one of the inputs (symptom or syndrome) changes, then general N-MMN structure changes too and final decision could be different. Again, CCG - both for N-MMN and triangle - immediately shows us all dynamic changes. Thus, cognitive graphics becomes an instrument for operations with IS knowledge base. It gives us unique opportunity to use knowledge and its cognitive representation for recognition of complex situations (patterns) with localized objects, influences and multiple variables. In the Global Risk model (Figure 1) it will let us to see what happens in the model if some variables of the risks are rapidly changing or some interconnection of risks become weaker/stronger. What kind of interconnections and influences are activated? What variables for different objects (risks) are to get changed? How can we localize the most dangerous impacts?

CCG tools in the N-MMN model allow us to apply this approach to recognition of different types of objects and situations: economic structure of the city or region, marketplace dynamics, scientific theoretical framework, military logistics and battlefield operation, governmental structures and processes, structure of course program and curriculum in educational etc. Do we face market stagnation or growth? What are the reasons of that? How effective governmental agency's operation is? Answers for the questions are to be given via elements of the network localized by means of CCG. For the purpose of the recognition of situation with high level of uncertainty, CCG for applied N-MMN based IS should be functional in the following ways: 1) represent objects by categories with key variables and interconnections, 2) activate any object of the network and any interconnection, 3) localize influences and vectors, including complex "chains" of influences, 4) localize segment of the network by assigned criteria.

3. CONCLUSION

CCG for applied knowledge based IS can be integrated with knowledge structure and used as means for pattern recognition through N-MMN. This approach gives high level of knowledge representation and dynamic recognition of complex situations. N-MMN based CCG tools can be applied for analysis, modeling, forecasting and decision making in different practical areas: decision making support in business and government, medicine, education, science and culture.

At the current stage of our work we consider following tasks of further development: 1) design complete set of CCG tools for IIS IMSLOG able to effectively operate with N-MMN based complex networks; 2) user friendly IS Interface design that gives full functional opportunities to operate with N-MMN graphic models and clear defined recognition and object localization; 3) adaptation of IS logical and mathematical apparatus to N-MMN based operation, including learning samples and validation.

4. ACKNOWLEDGMENTS

This paper prepared with support of Russian Foundation of Base Research, project № 07-01-00452a.

5. REFERENCES

[1] D.A. POSPELOV Cognitive Graphics – a window into the new world // Software products and systems, 1992 – p.4-6 (in Russian)

[2] A.E.YANKOVSKAYA, A.I.GEDIKE, R.V.AMETOV, A.M.BLEIKHER IMSLOG-2002 Software Tool for Supporting Information Technologies of Test Pattern Recognition // Pattern Recognition and Image Analysis. - 2003. - Vol.13. - No.4. - pp.650-657

[3] A.Ye. YANKOVSKAYA, GEDIKE A.I. Integrated Intelligent System EXAPRAS and its Application// Journal of Intelligent Control, Neurocomputing and Fuzzy Logic. – USA, Nova Science Publishers, Inc. – 1995. – Vol. 1. – pp. 243-269.

[4] GLOBAL RISKS 2009. A Global Risk Network Report // [On-line recourse] open access

http://www.weforum.org/pdf/globalrisk/globalrisks09/global_risks_2009.pdf

[5] A.Ye. YANKOVSKAYA Logical Tests and Tools of Cognitive Computer Graphics for Intelligent Systems // New Information Technology in Research of Discrete Structures: Proceedings of the 3-ed All Russia Conference with International Participation, Tomsk: Siberian Branch of RAS Publishing - 2000 - p.163-168

About the authors

Yankovskaya Anna Efimovna is a professor at Tomsk State University of Architecture and Building, Head of Laboratory of Intelligent Systems. Her contact email is yank@tsuab.ru, ayyankov@gmail.com.

Galkin D mitriy V ladimirovich is a candidate of science and docent of Tomsk State University, Department of Humanities and Informatics. His contact email is gdv_t@mail.ru.

Efficient acceleration structures layout for rendering on 32- and 64-bit many-core architectures

Maxim Shevtsov, Alexei Soupikov and Alexander Reshetov
Intel Corporation
{maxim.y.shevtsov, alexei.soupikov,alexander.reshetov}@intel.com

a) Asian Dragon model, 7.2M triangles, 64-bit extension consumes only 2Mb of 1.3Gb acceleration structure, extension processing time is <0.5% of rendering time

b) Thai Statue model, 2.2M triangles, 64-bit extension consumes only 2.1Mb of 1.4Gb acceleration structure, extension processing time is <0.5% of rendering time

d) Thai Statue model replicated 10 times (72M triangles total) 64-bit extension consumes only 4Mb of 7Gb acceleration structure, extension processing time is <0.5% of rendering time

Figure 1. 64-bit extension overhead for large models. Models are ray-traced with shadows at 1024x1024 on a 2-way 3GHz Intel ®Core™2 Duo machine (4 threads for construction/rendering) with 8 GB RAM

ABSTRACT

Any rendering solution needs fast acceleration structures to reduce the complexity of solving search problems. We address the following problems of constructing acceleration structures for the large number of primitives: compact memory layout, efficient traversal capability, memory address space independence, parallel construction capability and 32/64 bit efficiency.
This proposed kd-tree layout solution solves the above problems completely with highest possible efficiency. It is easily applied to other hierarchical acceleration structure types as well.

KEYWORDS: Rendering, acceleration structure, kd-tree, ray-tracing, proximity search.

1. Introduction

Rendering algorithms use acceleration structures to reduce the complexity of solving search problems [1]. Making their usage practical for high-speed parallel requires addressing of the following problems:

- Efficient traversal capability – compact representation should not slow down the traversal step
- Memory address space independence –an acceleration structure could be saved/loaded/transferred easily
- Parallel construction capability - the data format of acceleration structure should support creation in multiple parallel threads
- 32 and 64 bit efficiency – the acceleration structure size should not explode on 64 bit architectures. The 32 bit mode acceleration structure mode should have exactly the same binary representation on 64 bit architectures.

In this paper we propose specific memory layout solving the above problems. We use kd-tree as example, but the solution we proposed is also applicable to a wide range of partitioning hierarchies.

2. Previous work

The kd-tree is basically a binary tree in which each node corresponds to a spatial cell. Inner node of kd-tree represents splitting plane and refers to the two child nodes. Each leaf node, in contrast, stores primitives counter and refers to a corresponding list of primitives.

A compact kd-tree layout has been proposed in [3]. It uses only eight bytes per node. Storing offsets instead of pointers makes the data structure independent of the base address changes:

```
/* basic 8-byte layout for a kd-tree node */
struct KDTreeNode {
      union{
          //position of axis-aligned split plane
          float split_position;
          // or number of leaf primitives
          unsigned int items;
      }
      unsigned int dim_offset_flag;
//'dim_offset_flag' bits encode multiple data:
// bits[0..1]: encode the split plane dimension
// bits[2..30]: encode an unsigned address offset
// bit[31]: encodes whether node is an inner node or
leaf
};
// macros for extracting node information
#define DIMENSION(n) (n->dim_offset_flag & 0x3)
#define ISLEAF(n) (n->dim_offset_flag & (UINT)(1<<31))
#define OFFSET(n) (n->dim_offset_flag & 0x7FFFFFFC)
```

3. Solution

Efficient leaf/node test

During traversal the leaf/internal node test is executed at each traversal step and the branch depends on its results, so its performance is critical. Having 0 as a leaf indicator allows reducing the test to exactly 1 instruction before branch:

```
and Node, 0x03
jz  ProcessLeaf
```

```
/* 8-byte layout for a kd-tree node */
struct KDTreeNode {
       union{
            float split_position;
            unsigned int items;
       }
       int dim_offset_flag;
// 'dim_offset_flag' bits encode data in a new way
// bits[0..1] : indicate either
// • a leaf(if set to 0)
    // if 'items' field is >=0 it is true leaf
     // otherwise it is 64-bit extension
// • an inner node with split plane dimension
//   (if set to 1,2,3 for x,y,z axis corresp.)
// bits[2..31] : encode a signed address offset
};
```

32 and 64 bit efficiency

To handle an unpredictability of resulting tree size a construction algorithm usually allocates memory by continuous regions. The number of links between those sub-trees is relatively small (<<1% of total number of total number of kd-tree nodes).

The typical region size is way smaller than 4GBs. So the nodes can use 32-bit offsets to reference children within the same region. The only nodes that need 64-bit offsets are the nodes having children located in another memory region. *Since the number of such nodes is small they are encoded as extensions of 32-bit nodes. We use negative values of the primitive counter to indicate that the leaf is special, and it is a 64-bit extended node.*

Multiple threads construction

When the tree is constructed in multiple threads each thread builds some sub-tree [2]. Thus different threads may create a parent node and its children nodes. So when a parent is created the offset to children nodes may be unknown. That fact prevents from allocating 64-bit offset data next to a node. The 64-bit extended node data (actual 64-bit offset, actual leaf/axis and actual counter/split fields) is stored in a special per-thread table.

64-bit extension node is a special type of leaf with the following values of its fields:

a) *-(entry+1),* where *entry* is a table entry number, is stored in **items** field;

a) *(tbl)<<2* where *tbl* is a table number, is stored in **dim_offset_flag** field; note that 2 least-significant bits are zeroed, indicating a leaf.

Each construction thread creates its own 64-bit node table to avoid contention. If a table becomes full the correspondent thread just increases its size by re-allocation and data copy. Since the table is small it does not affect construction performance. The tests on models with up to 70M polygons demonstrated that 256-entry per-thread tables were never full. Storage or transmission of the tree located in multiple memory regions requires only adjustment of cross-region offsets in the table.

```
// relocation table's entries
struct TableEntry{
      //actual leaf/node but with offset==0
        KDTreeNode node;
      //true offset
        __int64 offset;
};
#define ISLEAF(n) (!(n.dim_offset_flag&0x3))
#define DIMENSION(n) ((n.dim_offset_flag&0x3)-1)
#define IS_64BIT_EXT(n) (n.items<0)
#define HASITEMS(n) (n.items)

#define MAKELEAF(n,its,ofs) n.items = its; \
  n.dim_offset_flag = ofs;
#define ENCODE64BIT_EXT(n,table_id,entry_id) \
      MAKELEAF(n,-(entry_id+1),table_id<<2)

#define DECODE64BIT_EXT(node, newadr) \
   int tab_id = (node.dim_offset_flag)>>2; \
   int entry_id = -node.items-1; \
   TableEntry e = m_tables[tab_id][entry_id]; \
   newadr = &node + entry.offset; \
   node =  entry.node;
```

Modifications of traversal algorithm

The tree constructed by 32-bit code can be rendered by 64-bit code without any modifications. To avoid testing 64-bit extension indicator at each traversal step the extended nodes are stored as leaves. Since the probability of traversing leaf is very small comparing to probability of traversing internal node the additional 64-bit extension test is performed at a very small fraction of traversal steps.

```
register KDTreeNode node = m_root;
// ADRINT is  int or __int64 (32/64-bit architectures)
ADRINT newadr = &node;
traverse_loop:
while (!ISLEAF(node)){
       //get dimension, traversal order, etc
         const ADRINT adr0 = newadr+…;//front child
         const ADRINT adr1 = newadr+…;//back child
        //traverse of either back/front child or both
        //…
}
//processing leaves
const int nitems = HASITEMS(node);
if(nitems>0){
       //...
}
#ifndef _M_X64
else if(IS_64BIT_EXT(node)){
//64-bit extensions processing:
//newadr is patched using relocation table
        DECODE64BIT_EXT(node, newadr);
      goto traverse_loop;
 }
#endif// _M_X64
//popping from stack etc
```

4. Memory/Performance considerations

Our experiments with variety of models proved that memory overhead from using proposed 64-bit extensions is negligible (refer to Figure 1 for examples models).

Also managing per-thread 64-bit node tables doesn't affect construction performance. Our measurements show that construction slowdown is <1% and thus is negligible too. We also performed tests for 1-128 construction threads with wide range of models (1-100M polygons). They demonstrated that 128-entry per-thread tables are far most than sufficient to connect portions of a tree constructed with different threads.

The performance of rendering using new layout supporting 64-bit extensions is the same as of rendering the efficient layout supporting 32-bits only (see Figure 1 for details). Even on complex models and high memory regions granularity the slowdown using the proposed layout was less than 0.5% comparing with 32-bit only offsets and one continuous memory region for the whole tree.

REFERENCES

[1] V. Havran: "*Heuristic Ray Shooting Algorithms*". PhD thesis, Czech Technical University in Prague, 2001.

[2] M. Shevtsov, A. Soupikov, and A. Kapustin.: "*Highly parallel fast kd-tree construction for interactive ray tracing of dynamic scenes*". In Proceedings of Eurographics 2007.

[3] I. Wald, "*Realtime ray tracing and interactive global illumination*", PhD thesis, Saarland University, 2004.

Modified Progressive Switched Median Filter for Image Enhancement

Denis Kuykin, Vladimir Khryashchev, Ilya Apalkov
Digital Circuit and Signal Laboratory
Yaroslavl State University, Yaroslavl, Russia
connect@piclab.ru

Abstract

Modified progressive switching median filter is proposed. It is constructed using introduction of the impulse noise detector from adaptive switching median filter into the progressive switching median filter. The results of comparative analysis of new algorithm with other filters are presented. The results show that the proposed algorithm is more efficient in the case of random-valued impulse noise removing from digital images.

Keywords: *impulse noise, impulse detection, nonlinear filter, order statistics, face detection, image preprocessing*

1. INTRODUCTION

Images corrupted by impulse noise often occur in practical applications. Impulse noise may appear in digital images because of channel decoder damages, dyeing down of signal in communication links, communication subscriber's moving, video sensor's noises and other [1].

In a variety of impulse noise models for images, corrupted pixels are often replaced with values equal to or near the maximum or minimum of the allowable dynamic range. For 8-bit images, this typically corresponds to fixed values near 0 or 255. We consider here a more general noise model in which a noisy pixel can take on arbitrary values in the dynamic range according to some underlying probability distribution. Let $v(n)$ and $x(n)$ denote the luminance values of the original image and the noisy image, respectively, at pixel location $n=[n_1,n_2]$. Then, for an impulse noise model with error probability p, we have

$$x(n)=\begin{cases} v(n), & with \quad probability \quad 1-p, \\ \eta(n), & with \quad probability \quad p, \end{cases} \qquad (1)$$

where $\eta(n)$ is an identically distributed, independent random process with an arbitrary underlying probability density function. If noise model describes fixed-valued impulse noise (salt-and-pepper noise) then $\eta(n)$ can only take on values 0 or 255 with equal probabilities. For random-valued impulse noise $\eta(n)$ is described by a uniform distribution from 0 to 255.

There are many researches who are trying to solve the problem of salt-and-pepper impulse noise removal, and they had achieved very good results [1, 2, 6, 8]. The removal of random-valued noise is more difficult problem, because the differences in gray levels between a noisy pixel and its noise-free neighbour most of the times are less significant [3]. Some impulse noise removal algorithms have demonstrated sufficient effectiveness in restoration of random-valued impulse noise corrupted images. Modern examples of them are: directed weighted median filter (DWM) [4], adaptive central weighted median filter (ACWM) [5] and signal-depended rank ordered mean filter (SDROM) [6].

In this work noise impulses detection and removing problems are considered separately and improved algorithm for random-valued noise removal is proposed.

The following metrics of algorithms effectiveness are used to provide the comparative analysis of these algorithms:

- peak signal-to-noise ratio (PSNR) [1];
- information fidelity criteria (IFC) [7];
- visually perceived quality of restored images.

2. MODIFIED PROGRESSIVE SWITCHING MEDIAN FILTER

The proposed algorithm (MPSM) is based on the known progressive switching median filter (PSM) [8] and adaptive central weighted median filter [2]. It uses switching schema which includes two stages of noisy image processing:

1. Preliminarily detection of noise corrupted pixels of digital image.

2. Filtering of noise impulses which have been detected in first stage of processing using gathered information about image properties.

At first stage ACWM-based impulse detector is used for detect noise corrupted pixels and at the second stage PSM filtering procedure is used to replace such pixels with approximately correct values. At the next section some results of image enhancement are depicted.

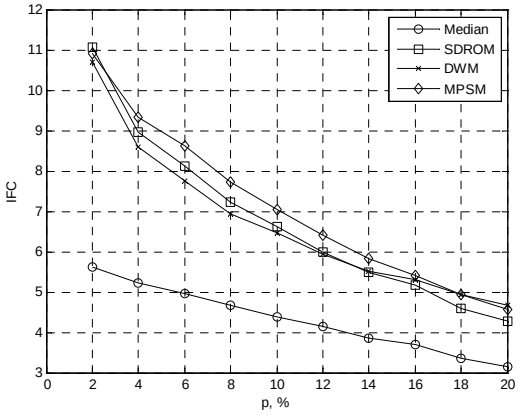


Figure 1. The dependence of IFC of restored image "Lena" from the random-valued impulse noise density

3. IMAGE ENHANCEMENT RESULTS

Above in Figure 1 IFC values of restored test image “Lena” for the set of filters are depicted. Results show that the proposed MPSM algorithm provides better IFC values than other considered filters if impulse noise density is relatively low (p ~ 10-20%). Its IFC value is about 5% more then ones for other considered filters.

Also it’s noticeable from analysis of restored images that the visual quality which is provided by MPSM filter is higher that one for other considered algorithms.

4. USING MPSM FILTER IN FACE DETECTION PREPROCESSING

Images containing faces are essential to intelligent vision-based human computer interaction, and research efforts in face processing include face recognition, face tracking, pose estimation, and expression recognition. Often it’s needed to detect the faces on noisy images. Below in Table 1 the face detection errors count in dependence of preprocessing algorithm is depicted. The Viola-Jones face detection algorithm [9] was used. Two types of errors are calculated: the missed faces (I) and the false-hits (II).

It can be noticed that using the MPSM filter on preprocessing stage provides better results in the set of situations especially if we would take into consideration only missed faces errors.

Table 1

Preprocessing influence on the face detection results
(10% random-valued impulse noise)

Image index	Count of faces	Detection errors							
		Original image		Corrupted image		Median filter (5×5)		MPSM	
		I	II	I	II	I	II	I	II
1	5	0	0	0	0	0	1	0	0
2	17	8	1	12	1	12	1	7	1
3	17	7	1	13	1	11	1	8	2
4	10	3	0	5	2	3	0	3	0
5	5	1	0	1	1	1	2	1	0
6	8	2	0	2	0	2	0	2	0
7	9	1	1	1	1	1	1	1	2
8	4	2	1	1	0	2	1	1	0
9	4	3	0	4	0	5	0	5	0
10	4	1	0	2	0	1	0	2	0
Total	83	28	4	41	6	38	7	30	5

5. CONCLUSIONS

Improved progressive switching median filter for random-valued impulse noise removal is proposed. The best random-valued noise detection algorithm and effective filtration algorithm were combined together and it produced new effective filter for random-valued impulse noise removal.

As it was shown by comparative analysis this MPSM filter provides better quality of restored images in comparison with compared to other considered algorithms. The numeric image quality indexes, visually perceived image quality and face detection results after MPSM processing were used in this analysis. MatLab implementations of MPSM filter are available on-line at www.piclab.ru/research/mpsm.html.

6. REFERENCES

[1] A. Bovik. *Handbook of Image and Video Processing (Communication, Networking and Multimedia). Academic Press. 2005.*

[2] H. Hwang, R. Haddad. *Adaptive median filters: new algorithms and results, IEEE Trans. Image Processing, vol. 4, pp. 499-502, Apr. 1995.*

[3] L. Yin, R. Yang, M. Gabbouj, Y. Neuvo. *Weighted median filters: a tutorial, IEEE Trans. Circuits Systems, vol. 43, pp. 157-192, Mar. 1996.*

[4] Y. Dong, S. Hu. *A new directional weighted median filter for removal of random-valued impulse noise, IEEE Signal Processing Letters, vol. 14, pp. 193-196, March 2003.*

[5] T. Chen, H. Wu. *Adaptive impulse detection using center-weighted median filters, IEEE Signal Processing Letters, vol. 8, pp. 1-3, Jan. 2001.*

[6] E. Abreu, Mitra S. *A signal-dependent rank ordered mean (SD-ROM) filter - a new approach for removal of impulses from highly corrupted images, in Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP’95), Detroit, USA, May 1995, vol. 4, pp. 2371 – 2374.*

[7] H.R. Sheikh, A.C. Bovik, G. de Veciana. *An Information Fidelity Criterion for Image Quality Assessment Using Natural Scene Statistics, IEEE Trans. Image Processing, vol.14, pp. 2117-2128, Dec. 2005.*

[8] Z. Wang, D. Zhang. *Progressive switching median filter for the removal of impulse noise from highly corrupted images, IEEE Trans. Circuits Systems – II, vol. 46, pp. 78-80, Jan. 1999.*

[9] P. Viola, M. Jones. *Rapid object detection using a boosted cascade of simple features // Proc. Int. Conf. on Computer Vision and Pattern Recognition. 2001. № 1. P. 511-518.*

About the authors

Denis K. Kuykin postgraduate student, Laboratory of Digital Circuits and Signals, Yaroslavl State University, Russia. His research interests are in area of nonlinear image processing. His e-mail address is denis.kuykin@gmail.com.

Vladimir V. Khryashchev Candidat of science, docent, Laboratory of Digital Circuits and Signals, Yaroslavl State University, Russia. His research interests are in areas of digital signal and image processing theory and applications. His e-mail address is vladimir@piclab.ru.

Ilya V. Apalkov Candidat of science, Laboratory of Digital Circuits and Signals, Russia. His research interests are in areas of nonlinear signal processing, image quality assessment, programming languages. His e-mail address is apalkoff@piclab.ru.

Motion Synthesis with Motion Splicing

Yusuke Takebayashi, Koji Nishio and Ken-ichi Kobori
Osaka Institue of Technology, Osaka, Japan
takebayasi@ggl.is.oit.ac.jp,{nishio, kobori}@is.oit.ac.jp

Abstract

In recent years, the reuse of motion capture data has become widespread. For example, many methods to generate new motion by synthesizing plural existing motions has been suggested. We can generate a composite motion such as "Walk while waving a hand" by using these methods. However, these methods often lose the feature of original data when they generate a new motion. Then, these methods generate an unnatural motion.
In this paper, we propose a motion synthesizing method with motion splicing. We generate a new motion from the existing motion based on relation of master and servant in a composite motion. Our method generates a spliced motion by splicing a human part. We splice five parts that is the lower body, arms, the torso and the head. Our method generates plural spliced motions, and generates a new motion by synthesizing those motions. In this way, our method can generate a new motion without losing the feature of original data.
Some experiment results show that our method generate a new motion from the existing motion.
Keywords: *Motion capture, Motion Synthesis, Motion Splicing.*

1. INTRODUCTION

In recent years, the reuse of motion capture data has become widespread. Motion capture data of short motion is on the market. However, the motion that a user needs may not be included in the existing motion capture data. The existing motion capture data have a limit to reuse. Therefore, many methods to generate a new motion synthesizing plural existing motions has been suggested. We can generate a composite motion such as "Walk while waving a hand" by using these methods. However, these methods tend to generate a new motion by blending plural motions on arbitrary parameters. Therefore, these methods often lose the feature of original data when they generate a new motion. Then, these methods generate an unnatural motion.
Therefore, the method to generate a new motion by splicing the motion of the upper body and that of the lower body has been suggested[1]. This method can generate a new motion without losing the feature of original data. However, splicing the motion of the upper body and that of the lower body is possible under a condition that the motion of the lower body of two motions must be similar.
In this paper, we propose a motion synthesizing method with motion splicing. Our method generates a spliced motion by splicing a human part based on relation of master and servant. Our method generates a new motion by synthesizing spliced motions. In addition, our method achieves flexible motion synthesis by splicing a human part.

2. GENERATION OF THE MOTION

Our method generates a new motion from the existing motion by using motion splicing and motion synthesis.
First, we explain relation of master and servant in a composite motion. When the human being performs plural motions at the same time, there is a dominant motion among plural motions. It is assumed that the human being performs "Walk while waving a hand". When the master motion is "Walk", the human being performs "Walk while waving a hand" by moving only the right arm. On the other hand, when the master motion is "Wave a hand", the human being performs "Walk while waving a hand" by moving the upper body greatly. Our method generates a motion by setting the master motion and the servant motion.

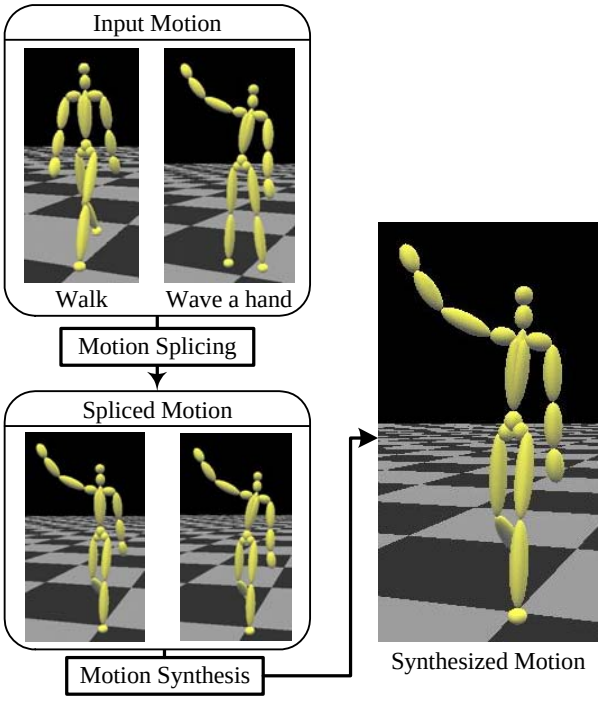


Figure 1: A flow of proposal method.

We show a flow of proposal method in the Figure 1. The flow of proposal method is as follows.

(1) One of input motions is set as the master motion. We generate a spliced motion by splicing the master motion "Walk" and the part of other input motion.
(2) We set "Wave a hand" as the master motion and generate another spliced motion.
(3) We generate a synthesized motion by synthesizing two spliced motions.

2.1 Motion splicing

Our method generates a spliced motion by splicing the part of the master motion and the part of other input motion. We splice five parts that is the lower body, arms, the torso and the head. In this paper, we define motion capture data $\mathbf{M}(t)$ in Equation (1).

$$\mathbf{M}(t) = \{\mathbf{p}(t), \mathbf{q}_1(t), \cdots, \mathbf{q}_N(t)\} \quad (1)$$

$\mathbf{p}(t)$ is the position of the root joint. $\mathbf{q}(t)$ is the rotate quaternion of the each joint. N is the total number of joints.
At first, our method generates the motion $\mathbf{M}_C(t)$ by splicing the master motion $\mathbf{M}_{Im}(t)$ and the part of the servant motion $\mathbf{M}_{Is}(t)$ in Equation (2).

$$\mathbf{M}_C(t) = \left\{\mathbf{p}^{Im}(t), \mathbf{q}_1^{Im}(t), \cdots, \mathbf{q}_i^{Is}(t), \cdots, \mathbf{q}_{i+n-1}^{Is}(t), \cdots, \mathbf{q}_N^{Im}(t)\right\} \quad (2)$$

i is the number of the root of the part to be spliced. n is the total number of joints of the part to be spliced.
However, there is a possibility that the motion $\mathbf{M}_C(t)$ becomes an unnatural motion when the part of the master motion is replaced with the part of the servant motion directly. The reason is because the movement of each joint depends on another joint when the human being acts. Therefore, we recalculate the rotate quaternion of the root joint of the part to be spliced and generate the motion $\mathbf{M}_S(t)$ in Equation (3).

$$\mathbf{M}_S(t) = \left\{ \mathbf{p}^{Im}(t), \mathbf{q}_1^{Im}(t), \cdots, \mathbf{q}_i^{S}(t), \mathbf{q}_{i+1}^{Is}(t), \cdots, \mathbf{q}_{i+n-1}^{Is}(t), \cdots, \mathbf{q}_N^{Im}(t) \right\} \quad (3)$$

We calculate the rotate quaternion $\mathbf{q}_i^S(t)$ of the root joint of the part to be spliced. At first, in order to keep the feature of the original motion, we calculate a feature of motion of the part from the motion $\mathbf{M}_{Is}(t)$ in Equation (4).

$$\mathbf{F}_j(t) = \mathbf{P}_j^{Is}(t) - \frac{1}{m}\sum_{k}^{\mathbf{B}} \mathbf{P}_k^{Is}(t) \qquad \left(j \in \{i, \cdots, i+n-1\}\right) \quad (4)$$

$\mathbf{B}$ is a set of the joint number of the chest. m is the total number of the chest. Our method defines a relative position from the chest to each part as the feature of motion of the part. The reason is because the chest moves a little when the human being acts. Secondly, when the master motion is spliced the part of the servant motion, we calculate the target position $\mathbf{P}_j^S(t)$ in Equation (5).

$$\mathbf{P}_j^S(t) = \mathbf{F}_j(t) + \frac{1}{m}\sum_{k}^{\mathbf{B}} \mathbf{P}_k^{Im}(t) \qquad \left(j \in \{i, \cdots, i+n-1\}\right) \quad (5)$$

Finally, we calculate the rotate quaternion that minimizes the sum of squared distances between the target position $\mathbf{P}_j^S(t)$ and the position of the part of the motion $\mathbf{M}_C(t)$ using the method of Horn[2]. And, we calculate $\mathbf{q}_i^S(t)$ by multiplying the minimum rotate quaternion by $\mathbf{q}_i^{Is}(t)$.

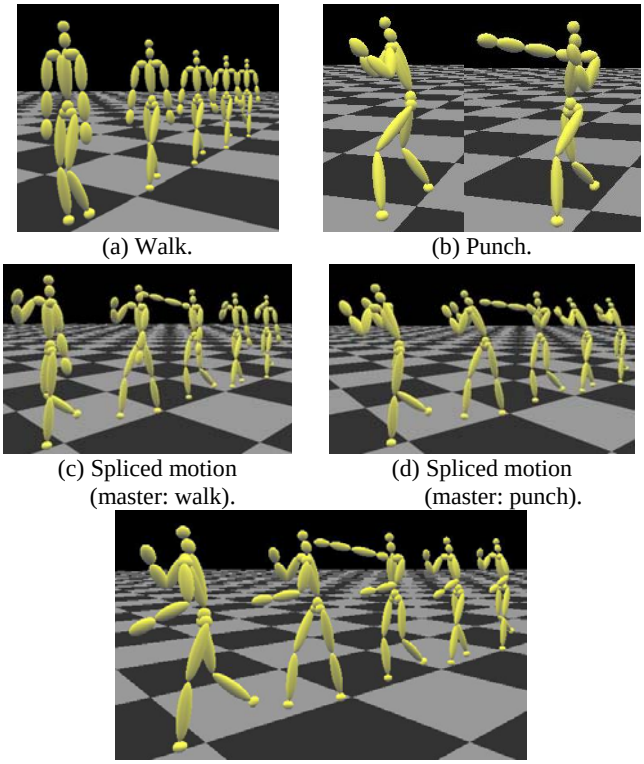
2.2 Motion synthesizing

Our method synthesizes two spliced motions by using linear interpolation. Because the motion capture data is constructed the position of the root joint and the rotate quaternion of the each joint, we can't apply linear interpolation to all data. Therefore, we calculate only the position of the root joint of the synthesize motion by using linear interpolation. On the other hand, the rotate quaternion of the each joint of the synthesized motion is calculated by using spherical linear interpolation. As a result, our method generates the synthesized motion $\mathbf{M}_O(t)$ in Equation (6).

$$\mathbf{M}_O(t) = \left\{ \mathbf{p}^{O}(t), \mathbf{q}_1^{O}(t), \cdots, \mathbf{q}_N^{O}(t) \right\} \quad (6)$$

3. EXPERIMENT

We had experimented to verify our method. We generated a new motion by synthesizing two spliced motions. A parameter of linear interpolation and spherical linear interpolation is set to 0.5 in the experiment.
We generate a new motion by using "Walk" and "Punch" in the experiment. Figure 2 shows the experiment result. Figure 2-(a) shows "Walk" and "Punch" that is input motions. Figure 2-(b) shows generated spliced motion when the master motion is "Walk", the servant motion is "Punch". In addition, Figure 2-(c) shows generated spliced motion when the master motion is "Punch", the servant motion is "Walk". Finally, Figure 2-(d) shows generated synthesized motion by synthesizing two spliced motions.
Our method can generate the spliced motion without losing the feature of the master motion. In addition, our method can generate realistic motion by synthesizing these spliced motions.



(a) Walk. (b) Punch.
(c) Spliced motion (master: walk). (d) Spliced motion (master: punch).
(e) Synthesized motion.

Figure 2: The experiment result.

4. CONCLUSION

In this paper, we have proposed the motion synthesizing method with motion splicing. Our idea is that two motion data are spliced based on relation of master and servant in a composite motion. Our method has generated a new motion by synthesizing two spliced motions. We have verified that our method could have generated a new motion that maintained the feature on each motion. In addition, we have verified that our method could have generated realistic motion by synthesizing two spliced motion.
In the future, we will generate a motion like the movement of the human being by synchronizing input motions. In addition, we will generate a motion like the movement of the human being by control of the speed of motion.

5. REFERENCES

[1] Rachel Heck, Lucas Kovar, Michael Gleicher, "Splicing Upper-Body Actions with Locomotion," EUROGRAPHICS 2006, No.3, Vol.5, pp.459-466, 2006
[2] Berthold K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," Journal of the Optical Society of America A, Vol.4, pp.629-642,1987.

About the author

Yusuke Takebayashi is a Ph.D. student at Osaka Institute of Technology, Graduate School of Information Science and Technology. His contact email is takebayasi@ggl.is.oit.ac.jp.
Koji Nishio is an associate professor at Osaka Institute of Technology, Department of Media Science. His contact email is nishio@.is.oit.ac.jp.
Ken-ichi Kobori is a professor at Osaka Institute of Technology, Department of Media Science. His contact email is kobori@.is.oit.ac.jp.

Object Extraction in Spatially Transformed Images

Boris Alpatov, Pavel Babayan
Department of Automation and Information Technology in Control
Ryazan State Radio Engineering University, Ryazan, Russia
atr_art@rgrta.ryazan.ru

Abstract

This work is addressed to the problem of object extraction in the images distorted by spatial transformations. The solution of this problem has been obtained using the method based on the Johnson distribution parameters estimation. The simplified real-time algorithm for an object extraction was developed.

Keywords: *object extraction, spatial transform, Johnson distribution.*

1. INTRODUCTION

Object extraction is defined as a partitioning of the image into object points and background points. The quality of object extraction determines the performance of solving of high-level image analysis problems [1]. One of the most popular approaches to object extraction is an estimation of a background image and comparison of this estimation and the observed image [2], [3].

In this work the main attention is given to the problem of object extraction under the effect of the random spatial transformations of images. The main reasons of them are the error of spatial transformation parameters estimation and the athmospheric ununiformity.

The particularity of the presented approach is the using of the asymmetric thresholding.

2. PROBLEM DEFINITION AND SOLUTION

Let $g(x,y)$ be a known background image. Fix the point of the current image with coordinates (x_0, y_0) and make some definitions for this point: r – the binary parameter corresponding to the existence of the object at this point, h – the luminance of the object, l – the observable luminance, $z_X \sim N(0,\sigma_z^2)$, $z_Y \sim N(0,\sigma_z^2)$ – the independent gaussian random values with the given variance defining random transformations of the image, ξ – the gaussian white noise of the image sensor: $\xi \sim N(0,\sigma_\xi^2)$.

Then the luminance of the observable image at the point (x_0, y_0) can be represented as:

$$l = g(x_0 - z_x, y_0 - z_y)(1-r) + hr + \xi . \quad (1)$$

The object luminance is assumed to be a random value with uniform distribution with boundaries $c_{\min}, c_{\max}$, $\sigma_\xi(x,y) << c_{\max} - c_{\min}$.

Thus, $g(x,y)$, l, σ_z^2, σ_ξ^2 are known. The problem is to find $\hat{r}$ to be the estimation for r.

The requirements for the estimation are defined with the Neumann-Pirson criteria:

$$P(\hat{r} = 1/r = 0) \le p_-,\ P(\hat{r} = 1/r = 1) = p_+ \to \max , \quad (2)$$

where p_- is the preliminary defined false extraction probability, p_+ is the true extraction probability.

The optimal decision rule for the problem is determined as:

$$\hat{r} = u^*(l) = \begin{cases} 1, \text{ if } \dfrac{p(l/r=1)}{p(l/r=0)} > \Lambda_0, \\ 0, \text{ otherwise,} \end{cases} \quad (3)$$

where Λ_0 is the threshold likelihood ratio, $p(l/r=1)$, $p(l/r=0)$ the probability distributions of the observable object luminance in the presence or absence of the object, respectively.

When $r=1$ the observed luminance is a composition of the variables h and ξ. Neglecting the noise ξ, $p(l/r=1)$ may be concerned as regular between $c_{\min}$ and $c_{\max}$.

Then the rule of the decision making is:

$$\hat{r} = u^*(l) = \begin{cases} 1, \text{ if } p(l/r=0) < (\Lambda_0(c_{\max} - c_{\min}))^{-1}; \\ 0, \text{ otherwise.} \end{cases} \quad (4)$$

In order to find the conditional distribution $p(l/r=0)$ it is possible to apply the Taylor decomposition of $g(x,y)$ at the point (x_0, y_0) restricting to 2-nd order members. Then

$$l = g(x_0, y_0) + f + \xi , \quad (5)$$

where

$$f = -\frac{\partial g}{\partial x} z_x - \frac{\partial g}{\partial y} z_y + \frac{1}{2}\frac{\partial^2 g}{\partial x^2} z_x^2 + \frac{1}{2}\frac{\partial^2 g}{\partial y^2} z_y^2 + \frac{\partial^2 g}{\partial x \partial y} z_x z_y . \quad (6)$$

The analytical derivation of the distribution of (5) is impossible. In order to find the approximation of this distribution we can represent (6) as quadratic form, diagonalize it, apply known expressions for the moments of normal distribution, calculate four moments for random value l and use them for the numeric calculation of the Johnson distribution parameters [4].

Thus the rule of the decision making is:

$$\hat{r} = u^*(l) = \begin{cases} 0, \text{ if } l_{\min} < l < l_{\max}, \\ 1, \text{ otherwise,} \end{cases} \quad (7)$$

where $l_{\min}$, $l_{\max}$ are the boundaries of $(1-p_-)100\%$ confidence interval for the Johnson distribution. In general this interval is asymmetrical relatively to the background luminance value; therefore the procedure defined by (7) is named here as the asymmetric thresholding. If the luminance in the neighborhood of

(x_0, y_0) variates linearly, then the algorithm comes to the proposed algorithm [1].

If the background image and the additive noise variance are unknown, they can be estimated with use of the procedures, explained in [1]. At the same time, the estimation of the variance of geometrical distortion σ_z^2 is a problem. In practice, a good result can be achieved with $\sigma_z = 0{,}1 \div 0{,}5$.

3. THE SIMPLIFIED OBJECT EXTRACTION ALGORITHM

The algorithm given with the expression (7) has sufficiently luck in computational complexity, especially in real-time applications.

The simplified object extraction algorithm concludes in the following operations for each pixel:

1. The calculation of the difference between the luminance of the current pixel and the luminance of its neighborhood. For the point (i, j) there must be calculated:

$$d_1 = \hat{g}(i,j) - \hat{g}(i-1,j),\ d_2 = \hat{g}(i,j) - \hat{g}(i+1,j),$$
$$d_3 = \hat{g}(i,j) - \hat{g}(i,j-1),\ d_4 = \hat{g}(i,j) - \hat{g}(i,j+1), \quad (8)$$

where $\hat{g}(i,j)$ – the estimation of the background image.

2. The finding of the estimations of low and high boundaries of the confidence interval of the centered asymmetric distribution:

$$L = \sigma_z \min\{d_1, d_2, d_3, d_4, 0\},\ R = \sigma_z \max\{d_1, d_2, d_3, d_4, 0\}. \quad (9)$$

3. The finding the estimations of low and high boundaries of the confidence interval

$$\hat{l}_{\min}(i,j) = g(i,j) + (L - \hat{\sigma}_\xi(i,j))T,$$
$$\hat{l}_{\max}(i,j) = g(i,j) + (R + \hat{\sigma}_\xi(i,j))T, \quad (10)$$

where T – half-width of $(1-p_-)100\%$ confidence interval for the normalized Gaussian random variable, $\hat{\sigma}_\xi(i,j)$ – the estimation of the standard deviation of the additive noise.

4. Making of the decision about the presence of the object

$$\hat{r} = u^*(l(i,j)) = \begin{cases} 0, \text{ if } \hat{l}_{\min}(i,j) < l(i,j) < \hat{l}_{\max}(i,j), \\ 1, \text{ otherwise.} \end{cases} \quad (11)$$

The described algorithm has a low computational complexity and can be effectively used in real-time image processing systems.

4. EXPERIMENTAL RESEARCH

During the experimental research three algorithms was compared:

I. The algorithm without using of background uniformity ($\sigma_z = 0$ in (9)).

II. The simplified algorithm.

III. The algorithm based on the Johnson distribution.

Two video sequences (50 frames long) were selected to conduct the quantitative experimental research. The image sequences contain the moving object observed in the cross-country background. The signal-to-noise ratio is about of 10. The size of the images is 300x300. The difference between the image sequences is in the spectral band of the observation (IR and TV).

The figure 1 illustrate operating curves for the IR sequence. The result for TV sequence is similar to the result for IR sequence.

An analysis shows that the algorithm (II) allows to improve the quality of object extraction. At the same time the algorithm (III) has no disadvantages in comparison with the algorithm (II).

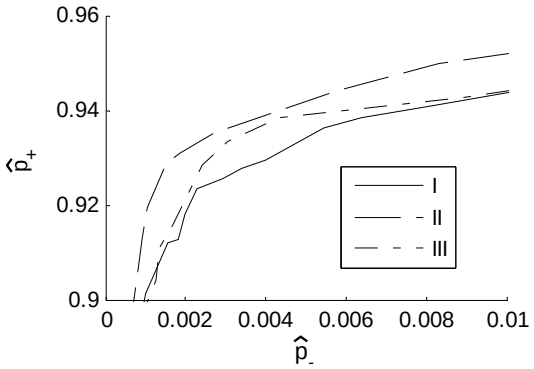


Figure 1: The operating curves for IR sequence

5. CONCLUSION

An improvement of the effectiveness of object extraction requires taking into the account the influence of spatial transforms. The research can be used to develop and enhance of real-time image processing and analysis systems. The research has been supported by the grant for the leading scientific school НШ-10.2008.10.

6. REFERENCES

[1] Алпатов Б.А., Бабаян П.В., Балашов О.Е., Степашкин А.И. Методы автоматического обнаружения и сопровождения объектов. Обработка изображений и управление. – М.: Радиотехника, 2008. – 176 с.: ил.

[2] Alper Yilmaz, Omar Javed, Mubarak Shah *Object Tracking: A Survey* ACM Computing Surveys, Vol. 38, No. 4, Article 13, Dec. 2006.

[3] Алпатов Б.А. Алгоритм обнаружения и выделения движущегося фрагмента изображения // Техника средств связи. Серия техника телевидения. – 1991. – №2. – С. 72-76.

[4] StatLib :: Data, Software and News from the Statistics Community http://lib.stat.cmu.edu – 01.11.2008

About the author

Boris Alpatov is a professor at Ryazan State Radio Engineering University, Department of Automation and Information Technology in Control. His contact email is aitu@rgrta.ryazan.ru.

Pavel Babayan is laboratory head at Ryazan State Radio Engineering University, Department of Automation and Information Technology in Control. His contact email is atr_art@rgrta.ryazan.ru.

Segmentation of chip microimages by contour tracking

Chochia P.A.
Institute for Information Transmission Problems of the Russian Academy of Sciences
(Kharkevich Institute), Moscow, Russia
chochia@iitp.ru

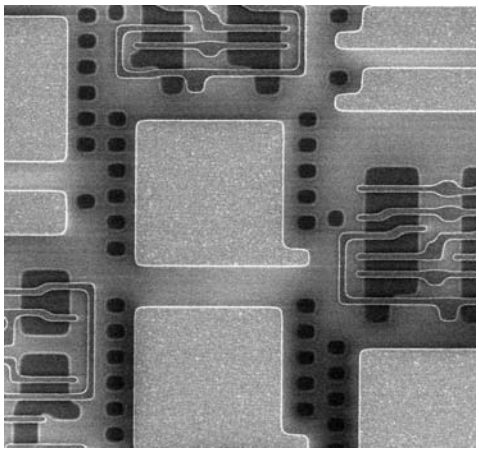
Abstract

The segmentation of chip microimages, obtaining by scanning electron microscope is considered. In addition to high noise level these images are characterized by the presence of two rather different types of borders between the patterns, where the width of the first–type borders varies in enough large range. The algorithm for transformation and segmentation of these images is proposed.

Keywords: *Image Segmentation, Image Processing, Contour Detection, Contour Tracking.*

1. INTRODUCTION

The developing of microelectronic devices needs the waffle layer studying. One of the ways is to analyze the waffle surface picture, captured by Scanning Electron Microscope (SEM). The task we consider is the detection of all patterns in the scanned data, i.e. the separation of the image to the set of segments. The image, obtained by SEM, is interpreted as two-dimensional monochrome signal; the example is represented in the picture. The approach to segmentation is based on detecting and tracking contour lines at the borders of patterns; the peculiar property of source images is the presence of two border types: light contrast ridges of different width (the first type), and brightness edges (the second one).



To detect the border between patterns we follow contour–tracking approach [1, 2]. Contrary to filter-based algorithms [3, 4] it has some advantages for our task, because it allows to apply the analyze window of various size and shape, and to obtain continuous and connected lines of minimum thickness.

Two types of pattern borders need different approaches to their detection and tracking. Basically on two-component image model [5], appropriate image conversion transforms the second border type to the first one and allows using only one contour tracking algorithm. Then this half-stuff is subjected to filtration and correction, and the final contour map is obtained.

2. SOURCE IMAGE PREPROCESSING

Under the preprocessing source image is transformed to the set of values, reflecting the contour existence probability in image pixel, that is near zero values in inner pattern areas and near ones at the ridges between them; we will name it *zero-aligned image*.

2.1 Aligned image with first–type contours

Zero–aligned image may be obtained by unsharp masking method [3, 5] as the difference between an image and its smoothed copy: $M_v(x) = x - S_v(x)$. Here x is the pixel of the source image X, $S_v(x)$ is smoothing operator response (mean value of X over $v \times v$ pixels fragment, centered at x point), and $M_v(x)$ is the result of filtration. The ridges width and contrast have large variety. To reduce some failures and to improve the result, the operator $S_v(x)$ is applied to the image with previously removed light ridges. This can be done with the following clipping procedure: $x_c = x$, if $M_a(x) \le t$, and $x_c = M_b(x)$, if $M_a(x) > t$. To reduce the image noise level, the median filtering [3] procedure $x_m = \mathrm{med}\{x_c \in \mu\}$ over 3×3 pixels vicinity μ was used. The image background alignment is carried out by subtracting smoothed clipped and filtered image X_m from the source image X: $A(x) = x - S_v(\mathrm{med}(x_m))$. If the difference is less then zero, $A(x)$ becomes equal to 0.

2.2 Aligned image with second–type contours

The gradient filter is applied to clipped and filtered image X_m to obtain zero–aligned image with second–type contours. This filter response equals to pixel values span over 3×3 vicinity, i.e. to the difference $x_g = \max\{x_m \in \mu\} - \min\{x_m \in \mu\}$. The fluency of the first–type contours is suppressed by using the clipped image x_m.

2.3 Dire ctional map

The directional map contains the information about contour trend in every image point. These data are used later for contour tracking. As known, the pixel values variance has its minimum in the direction along the contour, and maximum — in the perpendicular one. Therefore contour direction at a point can be defined as the direction of minimum pixel variation. In the algorithm we chose 4 directions for analysis: 0º, 45º, 90º и 135º.

3. CONTOUR TRACKING ALGORITHM

Contour tracking is carried out over zero–aligned image together with the directional map. After finding the contour threshold and detecting the set of initial contour tracking points, the contour lines tracking is performed. It includes: finding the (next) initial contour tracking point and interpret it as the current tracking point, choosing the contour direction at this point, finding the coordinates of the next point, defining ridge profile parameters, drawing the line segment to the next tracking point, calculating

the line profile parameters, and jumping to the next tracking point.

3.1 C ontour threshold

Contour tracking may be started from any contour point, therefore it is not necessary to detect all or the majority of contour points; one point is enough. Initial contour tracking points set is found by thresholding previously prepared zero–aligned image. This threshold is found by analysis of the pixel values distribution.

3.2 Contour direction and the next tracking point

The first step in finding sequent contour tracking point is the choice of the contour direction. Mean pixel values are calculated inside 3 sectors of 45º: in the sector that extends the current contour direction, and in two adjacent. The direction with the maximum of calculated values is chosen as new contour direction.

The width of the real ridges may vary in large range, and possible direction set is also enough coarse. Therefore the next tracking point coordinates need clarification realized by finding the central point of the ridge. This is done by finding the point, maximizing the correlation between ridge profile and rectangular aperture. Simultaneously with the line tracking the parameters of profile asymmetry A_{aver} and convexity C_{aver} are calculated.

3.3 Contour line leg drawing

The contour line leg is drawn from the previous point (i_n, j_n) to the next one (i_{n+1}, j_{n+1}), simultaneously extra initial points are cleared in some "corridor" near the line. The corridor width d is chosen $d = 2u' + 4$, where u' is the half–width of the ridge. The line drawing process finishes when current line leg touches another contour line or the image border. In such a case the algorithm starts new line tracking process beginning from the next initial point. Otherwise tracking point coordinates (i_n, j_n) are replaced by new ones (i_{n+1}, j_{n+1}), and the algorithm repeats tracking procedure.

4. UNITED CONTOUR TRACKING

For united tracking of the first– and the second–type borders, the intermediate contour map X_1 with first–type contours is integrated with aligned image X_2 containing second–type contours. Simultaneously the "corridor" of the same width $d = 2u' + 4$ near the tracked lines of X_1 is cleared for removing the parasitic lines in the image X_2. It may be written as: $x = x_1$, if $x_1 > 0$; $x = 0$, if x is in u'-vicinity of the contour; and $x = x_2$ otherwise. Then the tracking algorithm, discussed above, is applied to new united image X.

5. CONTOUR MAP PROCESSING AND FILTERING

5.1 Open lines prolongation and closing

Open line prolongation up to its closing is performed by the same contour tracking algorithm, discussed in Section 3. The algorithm is applied to new synthesized image X, being combined as the weighted sum of pixels of the first– and second–type aligned images (z_1 and z_2) with the map of already found contours (m) by the following formula: $x = k_1 z_1 + k_2 z_2 + k_3 m$. Here k_1, k_2, k_3 are the weight coefficients, and x is the synthesized image pixel value.

5.2 Segme nt numbering

Segment numbering is carried out by attributing some index (segment number) to every pixel of obtained contour map image. Simultaneously the numbering and verification of the resulting segments (that is the inner areas, separated by contour lines) is executed, and residuary open lines, if any, are removed.

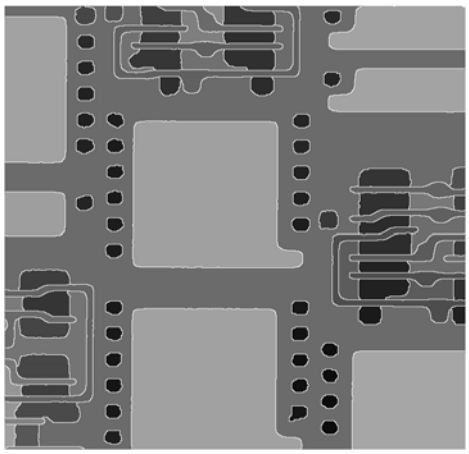
5.3 False lines and segments removing

Two types of false contour lines are possible. First of them (open line) has at least one non-connected end. Second one is correct from the connection point of view, but topologically it not constitutes the border and entirely places inside one segment. The algorithm verifies the segment numbers at both sides of every contour line. If the numbers are equivalent, the line is interpreted as false one and should be removes. This is realized simply by replacing its pixels by the value of neighbor segment pixel.

Due to the presence of noise, some false segments may appear as formally correct contour lines. For their detection and removing the contour line profile parameters of asymmetry A_{aver} and convexity C_{aver}, discussed in the subsection 3.2, are used. These values are compared with some thresholds T_A and T_C. If $C_{aver} < T_C$ or $A_{aver} < T_A$, the line is considered as false one and removes.

5.4 Final outcome

The segmentation result is represented in the picture below. The segments are displayed as constant value areas, filled by mean area brightness, and separated by thin white contour lines. The overall accuracy of detecting the segments is about 98%.



6. REFERENCES

[1] Roberts J.M. *Attentive Visual Tracking and Trajectory Estimation for Dynamic Scene Segmentation*. Ph.D thesis., Univ. of Southampton, 1994.

[2] Smith P., Drummond T., Cipolla R. *Edge tracking for motion segmentation and depth ordering* // Proc. 10th British Machine Vision Conf., V. 2, Nottingham, 1999, pp. 369–378.

[3] Gonzalez, R.C., Woods R. E. *Digital Image Processing*. 2nd ed., Prentice Hall, New Jersey, 2002.

[4] Фурман Я.А., Кревецкий А.В., Передреев А.К., и др. *Введение в контурный анализ и его приложения к обработке изображений и сигналов*. Физматлит, 2003.

[5] Чочиа П.А. *Двухмасштабная модель изображения* // Кодиров. и обработка изображений. М.: Наука, 1988, с. 69-87.

Space objects localization and recognition using an adaptive optical observation system

Vladimir Aleshin, Stanislav Klimenko, Dmitry Novgorodcev
Department of General and Applied Physics

Moscow Institute of Physics and Technology, Moscow, Russia
aleshin_vl@mail.ru, stanislav.klimenko@gmail.com, nov-dmitrij@yandex.ru

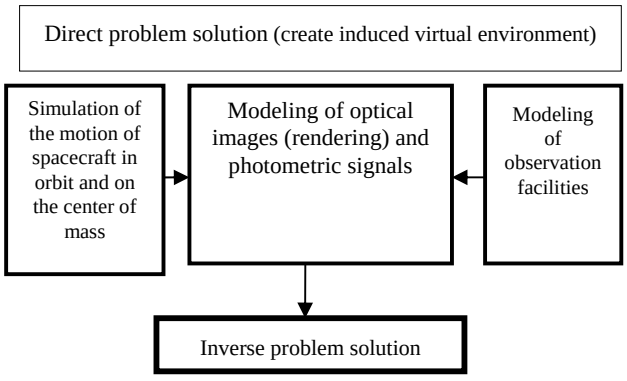
Abstract

The complex space systems are exposed to unforeseen failures and accidents. Given the high cost of development and maintenance of these systems, it is necessary to analyze the causes and consequences of accidents. Specificity of the given problems limits the publications on this theme. It is necessary to note materials of the conference Advanced Maui Optical and Space Surveillance System (AMOS) in which, unfortunately, are published only abstracts of articles. Great distance to the space objects (hundreds and even thousands of km) and essential influence of atmosphere makes quality of received images rather low, that does not allow to use the traditional methods of computer vision (offered, for example, in [11]). For the solving of inverse problems of localization and orientation estimation in our work we use methods of induced virtual environment, which principles are described in [3].

Keywords: *Space object, adaptive optics, localization, recognition*

1. INTRODUCTION

The task the space object state automatic monitoring is confined primarily to the determination of the stabilization and orientation estimation (localization in 3D space). Another challenge is to recognize the 3D form of an object using the database of space objects 3D models. The development of the physically adequate rendering technology[4,5,8] allows to use the corresponding algorithms not only for displaying the various peripheral devices [3], but also as a tool for simulation of direct problems [1,2]. Here is a schematic diagram of using the induced virtual environment in this task.



2. IMAGES PROCESSING OF ADAPTIVE OPTICAL SYSTEM

The problem of imaging is one of the crucial problems in the ground-based observations through a turbulent atmosphere. The most effective tool for compensation of phase distortions caused by fluctuations of refractive index of atmosphere is an adaptive optical system. The adaptive optical system consists of the wave front sensor and the active adaptive mirrors compensating the phase distortion measured by the sensor. The sensor and adaptive mirrors are connected by control feedback.

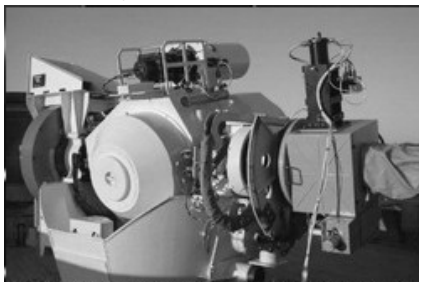


Figure 1: Russian telescope with adaptive optical system

In this paper, we use real images of space objects obtained through domestically developed adaptive optics system [9]. Despite significant improvement in the quality of images obtained, it is not possible to use traditional methods of computer vision. In the considered algorithms, we used the method of recognition based on the direct problem solving (model-based recognition).

The method for localization of the space objects implements the following [6,7,8]:

1. Hypothesis of stabilization selection (one of the prevalent hypothesis - focus on subsatellite point).
2. Set the starting rotation in the coordinate system and, if necessary, the rotation axis (this data is variation parameters).
3. According to orbital parameters the virtual scene is induced, calculating transformation matrices according the ballistic information of the stabilization system of space object and orientation of the telescope, as well as using a direction to the Sun vector and a local vertical vector (taking into account the suspension of the telescope). These matrices and vectors are determined for each frame of simulation video that provides calibration of images. In the ballistic data calculation, the modern astroballistic forecast software is used.
4. The calculated matrices and vectors is used in physically adequate rendering subsystem, replicating a sequence of ideal images (without atmosphere) This sequence of images is the aprioristic information for

elimination of ill-posedness issue. If there is the realization of photometric signal of this space object, images are converted into the photometric signal.

5. Using the obtained sequence of induced virtual images (taking into account the average residual dispersion function of the adaptive system) the variation criterion of an orientation estimation is formed. The criterion bases on sequences of the real measured and artificial induced images.
6. By the solution of the given variation problem the localisation and orientation state of space object with the maximum value of criterion is determined. The example of orientation estimation of spacecraft UARS (the unit vector of orientation, which corresponding to the maximum value of variation criterion is equal to (0.98, 0.03, 0.19) in orbital coordinate system) is shown below:

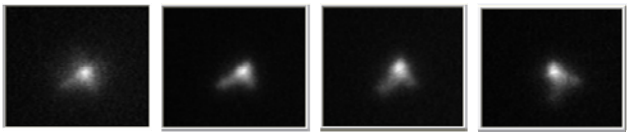


Figure 2: The sequence of real images of spacecraft UARS

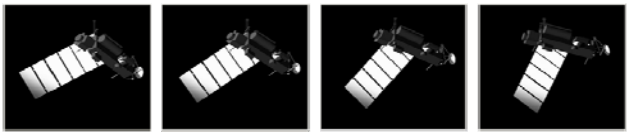


Figure 3: The sequence of artificial induced images of spacecraft UARS (orientation vector 0.98, 0.03, 0.19)

3. CONCLUSION AND RESULTS

In this article the results of researches are presented and the real optical observations obtained through domestically developed adaptive optics system are analysed. These measurements are made at the diffraction limit of the resolution and at the quantum limit of sensitivity. In addition to the problems of poor conditionality and ill-posedness according to Tikhonov, typical for linear inverse problems, there is a degeneration problem - presence of non-uniqueness of the solution. Software programs of space objects localization and orientation estimation are implemented on the basis of the induced virtual environment methods. The induced virtual environment is created using modern astroballistic forecasts and physically-adequate rendering. Wide experimental material (more than 100000 images) is processed using these programs and it is shown that the accuracy of space object localization may be several degrees at observation range from 300 to 1400 kilometers [9].

4. REFERENCES

[1] A. Tarantola, *Inverse Problem. Theory and Methods for Model Parameter Estimation – SIAM, 2005*.

[2] A.N. Tikhonov, V.Y. Arsenin, *Methods for solving ill-posed problems - M.: Nauka, 1986.*

[3] Aleshin V.P., Afanasev V.O. Baygozin D.A., Baturin Y.M., Klimenko S.V. etc. *The system visualization of induced virtual environment: the status of the project / / The collection of 14 th Int. Conf. «GraphiCon-2004», MSU, 2004, - 318 pp., 12-15 pp.*

[4] V.A. Galaktionov *Software technology of synthesis of realistic images. Abstract. Dis. e. f.-m. Mr. IPM, M. 2006*

[5] A.G. Voloboy A.G., Galaktionov V.A., K.A. Dmitriev, E.A. Kopylov, *Bidirectional ray tracing using quasi-Monte Carlo method for integrating the illumination / / "Software", № 5, 2004 - pp. 25-34.*

[6] Aleshin V.P., Afanasev V.O., Baygozin D.A., Baturin Y.M., Klimenko S.V., Lavrov V.V., Novgorodtsev D.D., Ryndin J.G. *The methodology and results of the application of virtual environment technology to solve the problems of recognition of the spacecrafts and evaluation their parameters using ground-based observations. // Proceedings of 6th International Workshop on Virtual Environment on PC Cluster, VEonPC 2006, Protvino-Altai, published by ICPT, ISBN 5-88835-018-4, – pp. 22-28.*

[7] Aleshin V.P., Afanasiev V.O., Klimenko S.V., Lavrov V.V., Novgorodians D.D., Ryndin J.G., *Methods of computer graphics and induced virtual environment in problems of non coordinate data processing / / Questions of Radio Electronics. - № 4, 2007. - pp. 52-72.*

[8] Aleshin V.P., Klimenko S.V., Lavrov V.V., Novgorodtsev D.D., *Modeling of optical images and photometric signals using a physically-accurate rendering and induced virtual environment technology / / Questions of Radio Electronics. - № 4, 2007. - pp. 73-90.*

[9] Aleshin V. P., Grishin E.A., Inshin P.P., Novgorodtsev D.D., Shargorodsky V.D., *Capabilities estimation of low-orbiting spacecrafts observations by the adaptive optics system of the Altay optic-laser Center. International conference "Near-Earth Astronomy-2009", Kazan, 2009*

[10] 2006**,** 2007, 2008 Amos Conference Abstracts Of Technical Papers

[11] Harry Voorhees, Robert Radke, Conrad Poelman., *Automatic Reconstruction of Spacecraft 3D Shape from Imagery, 2008 Amos Conference, Abstracts of Technical Papers*

About the author

Vladimir Aleshin is a docent at Moscow Institute of Physics and Technology, Department of General and Applied Physics. His contact email is aleshin_vl@mail.ru.

Stanislav Klimenko is a professor at Moscow Institute of Physics and Technology, Department of General and Applied Physics. His contact email is stanislav.klimenko@gmail.com.

Dmitry Novgorodtsev is a post-graduate student at Moscow Institute of Physics and Technology, Department of General and Applied Physics. His contact email is nov-dmitrij@yandex.ru.

THE WAY OF AUTOMATION OF GRAPHIC METHOD OF THE SOLUTION OF MATHEMATICAL MODELING PROBLEMS

Tolok A.V.
Laboratory of the Computer graphics, specialized technical and program means
Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
atol@ipu.rssi.ru

Abstract

The offered approach is the further computer development of graphic methods of the solution of mathematical modeling (MM) problems. It will be a question of ways of computer graphic information's organization in the form of special images-models, which reflect geometrical sense of an object that was analytically set. Such way of image graphic representation of complex analytical statements allows to simplify algorithms of problem's solution. Thus the breadth of MM application in the fields of researches allows speaking about prospects of the use of such approach to various classes of applied problems.

Keywords: *Image-model, normal field, recursive algorithm, geometrical modeling, voxel data representation, mathematical programming, optimization problems, gradient method.*

1. INTRODUCTION

The process of visualization of graphic information is unequivocally important, but is far not its unique application. Computer graphic information as the organized image of graphic data can and should be actively applied as analyzing model of investigated object. Illustrativeness of graphic representation in existing graphic means of solution of mathematical modeling problems allows to find result quickly not demanding construction of complex analytical expressions. Thus graphic data should display geometrical properties of investigated function adequately. It is possible to consider such image as a model of function. We organize a process of the computer analysis of mathematical models in a graphic way on the basis of application of graphic images-models.

One of requirements to *computer graphic image-model* should be its *dimensional commensurableness with object-prototype*. In this case only it is possible to speak about adequacy of an *image-model* to the prototype.

The characteristic of change of normal inclination in neighbor points of an object is the invariant, so it's a geometrical property which characterizes a constancy of an object form in various transformations. We allocate this property and model it using graphic images on the basis of changes of color intensity. In such graphic image we organize the local geometrical information, in the form of some set of the images, allowing to automate possibilities of the differential analysis of object-prototype. Such graphic image we will define as *image geometrical model*. As such graphic image can contain not only the information on geometrical properties of a prototype it is possible in general case to speak about some *image-model (M-image)* [1].

In a stage of development of analytical design systems the special role is played by process of formation and the analysis of spatial objects. In this case voxel graphic representation causes special interest for developers of analytical CAD because of orderliness, regularity and index *commensurableness* of elements of an image with the object.

One of the basic differences of *M-images* from a traditional *graphic representation* of projected object is the width of its application in the further automated processes (for example, rendering, optimizing and engineering calculations, etc.). *The M-image* is dimensionally equal with object, contains local geometrical characteristics of an object, and allows to generate new *M-images.* Voxel representation is considered as a set of cubic neighborhoods of the points which fill the orthogonal space of geometrical object so, contains possibility of definition of normal's components in these points. The offered way of voxel graphic representation of such information allows to form the *image geometrical information* about analytical object as a *voxel graphic structure*. Under the voxel graphic structure of an image geometrical model here is offered to consider structural organization of M-images as integer scalar fields. Base images of structure display components of a normal field, normalized on the value of a colour palette. Other M-images as structure elements are generated on the basis of base M-images and display differential properties of an object. Base images represent the relational change of components of the normal for the analytically set object (U).

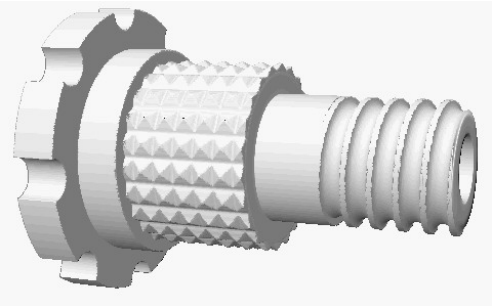


Figure 1: Thread bush, described by a mathematical apparatus of R-functions

The quantity of base images coincides with the dimension of the object (for example, in 3D-case A_u, B_u, C_u are coefficients of tangent plane in the differential neighborhood of a point of the object (U)). At next level images of partial derivatives are automatically determined in plane xOy $(\frac{\partial u}{\partial x} = \frac{A_u}{C_u}; \frac{\partial u}{\partial y} = \frac{B_u}{C_u})$. On the base of the images of partial derivatives we can get images of components of the normal fields $(A_{\frac{\partial u}{\partial x}}, B_{\frac{\partial u}{\partial x}}, C_{\frac{\partial u}{\partial x}})$ and $(A_{\frac{\partial u}{\partial y}}, B_{\frac{\partial u}{\partial y}}, C_{\frac{\partial u}{\partial y}})$, enabling acquisition

of images of higher derivatives etc. Thus a graphic structure representing geometrical properties of object (U) is generated.

2. THE "RANOK" SYSTEM

The system of analytical designing RANOK (Recursive ANalysis on image Components) is based on the presented principles. It recursively synthesizes the base graphic M-images. Graphic information in base M-images allows rendering the object (Fig. 1)) [2]. Due to this information the RANOK system allows to expand the possibilities in the field of researches of object. A gradient method, allowing to solve optimization problems, is automated. On a Fig. 2 the result of gradient motion for function, describing a «cup» is shown.

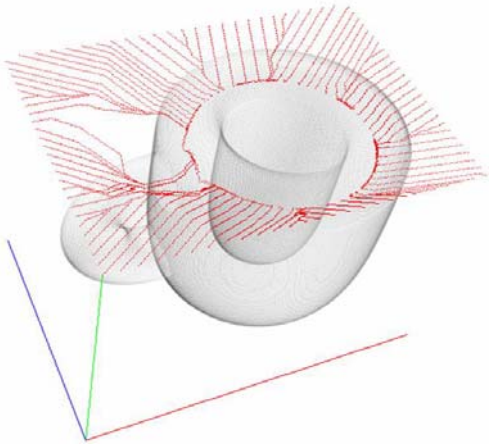


Figure 2: Movement on the gradient descent from the perimeter of a clearance verge

Figure 3 depicts the curve of gradient development w of the considered system enabling one to carry out the long-term planning with allowance for gradual increase of the flows under study. At that, the values of the three flows can be determined at each point of the gradient motion.

3. FORMULATION OF THE OPTIMIZATION PROBLEM OF MATHEMATICAL PROGRAMMING BASED ON THE APPARATUS OF R-FUNCTIONS

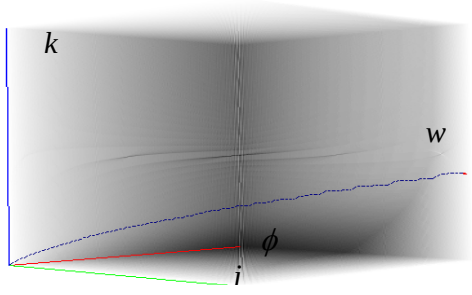


Figure 3: Example for demonstration of a gradient development of the 3-D system.

The equation system with boundary conditions of mathematical programming problem we replace by its logic intersection w

$$w = \bigcap_{i=1}^{n} w_i = w_1 \wedge w_2 \wedge \ldots \wedge w_n .$$

where w_i - equations of boundary conditions.

The objective function F is applied in the geometrical model F_w which is ready to research by gradient method in the RANOK system as follows:

$$\begin{cases} F_w = F + w_0 \cdot (1 + \nabla F) \to \max \\ F_w = F + \left| w_0 \cdot (1 + \nabla F) \right| \to \min \end{cases},$$

where $w_0 = w - |w|$ – is the zeroed domain of permissible plans.

Illustrations of the Examples of Solution of Problems of Nonlinear Programming in the RANOK System are presented in the proceedings [3].

4. CONCLUSION

Computer technologies of graphic methods of solution of tasks is just begins to emerge, but even now it is possible to state that it offers a tool for dealing with a wide range of computer-assisted mathematical applications One of the methods of using the vector fields represented in the raster and voxel images for solution of the gradient problems allows one to reconsider application of the field theory in the automation systems. This approach is characterized by the reduction of the density of the field under consideration to a representation as vivid as possible. The listed possibilities of the system do not describe the entire class of problems solved on the regular voxel structures of the cognitive M-images. Application of R-operations enables one to describe piecewise the domain of permissible solutions and objective functions of any complexity, and the cognitive model images used in the context of the RANOK system would enable one to work with spaces of any dimensionality.

5. REFERENCES

[1] Tolok, A.V., *Design of Computer Images of Geometric Characteristics for Estimation of the Relief of the Surface of a Function of Two Variables*, Zbirnik dopovidei NAN Ukraini, Mat.,prirodoznavstvo, tekhnichni nauki, 2004, no. 4, pp. 63-69.

[2] Maksimenko-Sheiko, K.V., Matsevityi, A.M., Tolok, A.V., and Sheiko, T.I., *R-functions and Inverse Problem of Analytic Geometry in the Three-dimensional Space*, Ezhemes. teoret. prikl. nauch.-tekhn. zhurn. 2007, vol.~10, Moscow: Novye tekhnologii. pp. 23-32.

[3] Korogod V.L., Myl'tcev A..M., Tolok A.V. *The solution of the mathematical programming problems in the system if analytical design* RANOK // Management of the large-scale development systems (MLSD'2008), (Upravlenie razvitiem krupnomasshtabnyh system (MLSD'2008) Materialy vtoroy megduharodnoy konferencij (1-3 October 2008, Moscow, Russia). Volume 2. M.: Trapeznikov Institute of Control Sciences, Russian Academy of Sciences 2008. – pp.122-125

.

Using Contours and Colour Region Boundaries of Photographs in Sculptural Portrait Design

Alexandra Solovyova, Alexander Kuchuganov
CAD Systems Department
Izhevsk State Technical University, Izhevsk, Russia
kuchuganov@istu.ru, alexandra-solovyova@yandex.ru

1. INTRODUCTION

Creating sculptural portraits is one of developing trends of 3D graphics. The development of multimedia and means of communication opened up new fields of their use in video conferencing, robotics, design, advertising actions. A 3D portrait can be created with the view of both achieving photographic resemblance and creating a caricature. However, achieving the desired resemblance degree requires certain artistic skills and time consumption connected with laborious geometric constructions.

Different methods for 3D head model design automation include laser scanning techniques, methods for 3D head model reconstruction from video clips, stereopairs, sets of images acquired at different view angles, methods that represent 3D head models by combining standard models stored in a database. A group of methods is based on the modification of generic models. These methods use less memory, but require paying more attention to specifying individual features of the depicted person.

In the article the method for standard wireframe parametrization is proposed which is based on the modification of a 3D model according to the location of contours and colour region boundaries of a photographic image.

2. STANDARD WIREFRAME DESIGN

We create 3D models using a kinematic method which represents the model as a forming contour and its motion law defined by guide curves [1]. Among its advantages are the simplicity of converting sketches and drafts into a 3D model and a small number of contours required for the model construction. A 3D wireframe consists of the basic section, which is the original forming contour; two basic profiles which define the projections of the object on coordinate planes (in the case of head model construction they correspond to the full-face and half-face photographs); additional sections which define the transformations the forming contour undergoes while moving along its path; 3D profiles which lie on the surface of the model and define its shape.

To form the standard wireframe the sections of maximally different shape were left. The 3D profiles were placed in such a way as to define the shape transformations of the sections and pass through the feature points of the face. The contours were designed with respect to the location of facial muscles and in compliance of proportions based on the Golden Ratio.

3. PROCESSING PHOTOGRAPHS TO OBTAIN INFORMATION ON 3D SHAPE

Photographs can serve as an available source of information on the depicted person. To make the standard model more precise we use full-face and half-face photographs defining the basic profiles of the wireframe. The photographs are taken with uniform illumination coming from the camera, which makes it possible to use the shape from shading approach.

Processing is applied to the images to obtain detailed information on the shape of the face. This processing includes dividing the photographs into gradations of brightness and colour clustering using several parameters [2]. The contours approximating the regions of lower and higher brightness are extracted. The contours and boundaries of the selected regions are represented in vector form.

Using brightness gradations is not always convenient to obtain unambiguous information on the shape. It is necessary to introduce additional clustering parameters to provide a more adequate 3D shape description. An angle between the light source direction **l** and the normal vector **n** to the facial surface can be used as one of such parameters. If the surface is considered an ideal diffuser, then, according to Lambert's cosine law, the cosine of this angle is proportional to intensity reflected by the surface [3]: $I_d = I_s \cos(\mathbf{n}\hat{}\mathbf{l})$, where I_s is source intensity. Adding a number of further simplifications, we can estimate the cosine of the angle between the light source direction and the surface normal. The results of clustering using this parameter provide us with a closer approximation of the spatial characteristics of the facial surface. The way the cluster boundaries are located in the forehead, cheeks, cheek-bones and chin areas is similar to that of the profiles forming the wireframe (Fig. 1).

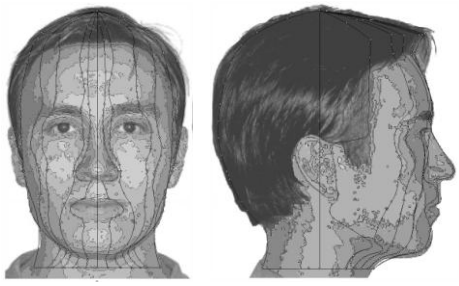


Fig. 1: "Stretching" a wireframe to cluster boundaries

The facial feature points we use include extreme vertical points of the head (vertex and gnathion), paired (eye corners, mouth corners etc.) and unpaired (nose tip, subnasal point, bridge of nose) points. We perform the search for the facial feature points using contours approximating the brightness edges of the images [4]. Some of the feature points in both projections can be found as points of contours with extremal properties. The location of the nose tip, the subnasal point and the bridge of the nose can be successfully determined using contours extracted from the half-face photograph (Fig. 2). It is possible to make up for a deficiency of information on the location of the feature points using the set of equations based on the facial proportions which controls their relative position.

The feature point detection method basing on the extracted contours appears to be more applicable in the case of insufficient

illumination quality and when processing images with arbitrary head pose.

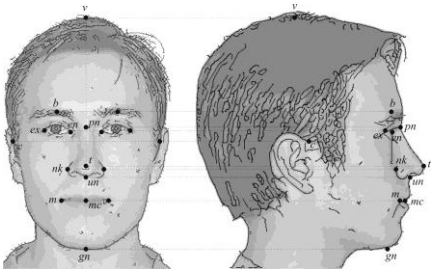


Fig. 2: Feature point detection basing on the extracted contours

4. STANDARD WIREFRAME PARAMETRIZATION METHOD

We use facial feature points to establish the connection between the photographs and the 3D wireframe. Their location on the photographs is determined using the extracted contours, the corresponding wireframe points are also known. Basing on the feature points a simplified face scheme is designed which defines basic 3D head shape elements (Fig. 3). The points forming the simplified model define the location of the 3D wireframe profiles. Their coordinates are stipulated by algebraic relations.

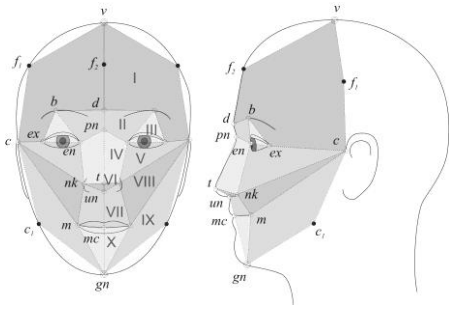


Fig. 3: Simplified face scheme

The first step of standard portrait individualization is matching the wireframe feature points with their projections on the photographs. The simplified face scheme parameters are calculated using the coordinates of the feature point projections. Basing on the scheme the 3D wireframe is designed. Its profiles are constructed by means of interpolating splines. Thus, the standard wireframe is "stretched" to the given points.

Further refinements are made to the model by performing the synthesis of wireframe fragments using contours and colour region boundaries previously extracted from the images. Each of the profiles stretched to the photograph should be transformed into a new one, passing through feature points, both similar in shape to the original one and taking into account the nearby fragments of colour region boundaries. We use smoothing splines to reach this effect [5].

Smoothing splines make it possible to construct a smoothly varying curve passing near the nodes of a specified grid within the given limits of error. The set of the ends of line segments and circular arcs approximating the profile being processed and colour region boundaries located within its neighbourhood acts as a grid in our situation. Let the grid formed by their vertical coordinates be of the form $z_0 < z_1 < \ldots < z_m$.

As the 3D profiles of the wireframe define the way the forming contour transforms while moving along the vertical axis, each vertical coordinate value z can correspond to a unique point of each profile alone. We seek the modified profile in the form of function $S(z)$ which is a smoothing spline defined on a grid $z_0 < z_1 < \ldots < z_m$ minimizing the functional $J(f) = \int_a^b \left(f''(z)\right)^2 dz + \sum_{i=0}^{m} \frac{1}{\rho_i}\left(f(z_i) - y_i\right)^2$ (this condition specifies the compromise between smoothing and approximating), where y_i are the grid node values (in our situation the horizontal coordinate values of the points forming the profile and the colour region boundaries), ρ_i are the weighting factors controlling the spline properties.

Altering the weighing factors we can control the degree of approximation between the resulting contour and each of the original points. We use this feature to keep the feature points of the wireframe from changing their previously defined positions and regulate the wireframe modification degree as well.

The sections of the standard wireframe are corrected automatically. To do this we fix the section positions in relation to the points of the profiles and set the correspondence between the section points and 3D profiles crossing them.

An example of a sculptural portrait designed basing on the given photographs is shown in Fig. 4.

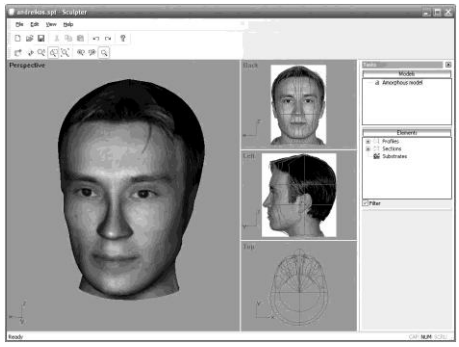


Fig. 4: An example of a sculptural portrait

5. CONCLUSION

The proposed method for standard wireframe parametrization based on the contours and colour region boundaries of the photographs is universal and can be used for creating other geometrically complicated 3D models.

6. REFERENCES

[1] Kuchuganov V.N., Kharin V.V. *Kinematic geometric models in conceptual design* // International Conference GraphiCon 2003.

[2] Gonsales R.C., Woods R.E., 2002. *Digital image processing*. Pearson Education, Inc, publishing as Prentice Hall.

[3] Shapiro L.G., Stockman G.G., 2001. *Computer vision*. Pearson Education, Inc, publishing as Prentice Hall.

[4] Kuchuganov A.V. *Recursions in tasks of images analysis* // Pattern Recognition and Image Analysis: New Information Technologies (PRIA-9-2008). Vol. 1, pp. 365-36.

[5] Shikin E.V., Plis L.I. *Curves and Surfaces on the Computer Screen. Spline User's Guide.* – M.: DIALOG-MIFI, 1996. – 240p.

About the authors

Alexandra Solovyova is a Master's degree student at Izhevsk State Technical University, CAD Systems Department. Her contact email is alexandra-solovyova@yandex.ru.

Alexander Kuchuganov is an associate professor at Izhevsk State Technical University, CAD Systems Department. His contact email is kuchuganov@istu.ru.

Анализ точности моделирования глобального освещения

А.Г. Волобой*, С.В. Ершов*, Д.Д. Жданов**, И.С. Потемин*, Л.З. Шапиро*
*Институт прикладной математики им. М.В.Келдыша РАН, Москва
**Государственный Оптический Институт им. С.И. Вавилова, Санкт-Петербург

Аннотация

В данной работе рассматриваются вопросы точности моделирования освещения сложных сцен и построения их фотореалистических изображений. В работе был проведен анализ ошибок моделирования. Были приведены оценки прогнозирования времени вычислений, необходимого для достижения заданной точности моделирования, как критерия, определяющего окончание процесса моделирования.

Ключевые слова: *реалистичные изображения, глобальная освещенность, карта освещенности, оценка точности.*

1. ВВЕДЕНИЕ

Физически аккуратное моделирование освещения находит широкое применение в архитектуре, при проектировании светотехнических устройств, а также при построении реалистических изображений сцен виртуальной реальности. Результатом такого моделирования могут быть либо карты распределения световых характеристик (освещенности, яркости или интенсивности света) на объектах сцены, либо фотореалистическое изображение сцены.

В задачах, связанных с построением фотореалистичных изображений, эффективным является сочетание методов прямой трассировки лучей, рассчитывающих распределение глобальной (или вторичной) освещенности или яркости на объектах сцены, с методами обратной трассировки, суммирующих яркости объектов сцены с прямыми яркостями от источников света [1, 2]. Идея прямой трассировки лучей методом Монте-Карло состоит в статистическом воспроизведении механизма распространения света путем моделирования всевозможных траекторий лучей от источников света до объектов сцены. Траектории световых частиц (фотонов) прослеживаются на всех этапах их существования, от момента их генерации источниками света до поглощения или выхода из сцены. Метод естественным образом поддерживает все типы поверхностей, включая сочетания диффузных и зеркальных свойств, а также поверхности, описываемые сложными функциями отражения (преломления) света. Поскольку в нашей реализации расчета глобальной освещенности каждый фотон переносит элементарный поток, то объекты сцены могут накапливать падающий на них поток и, в зависимости от типа объекта, сохранять его в виде карт освещенности или яркости.

Данная работа посвящается методам оценки погрешностей моделирования глобальной освещенности (яркости) объектов сцены и прогнозированию времени моделирования, необходимого для достижения заданной точности.

2. ОСНОВНЫЕ ИСТОЧНИКИ ОШИБОК

Вычисление световых характеристик осуществляется методом Монте-Карло и содержит стохастические ошибки вычисления. Кроме того, световые характеристики вычисляются на элементах объектов сцены конечных размеров (в вершинах треугольной сетки или в текстурных ячейках). Поэтому точная оценка световых характеристик в заданных точках сцены невозможна. Таким образом, погрешность вычисления состоит из двух основных ошибок – ошибки «дискретизации» сцены, связанной с конечными размерами элементов сцены, и ошибки «сходимости», связанной со случайным процессом вычислений.

Ошибка «дискретизации» — это ошибка интерполяции освещенности в пределах элемента сцены, накапливающего световой поток. Каждый элемент сцены накапливает единственную величину, и, очевидно, чем больше элемент сцены, тем более проблематично осуществлять точное распределение этой величины в пределах элемента сцены. Наиболее оптимальным решением является адаптивное разбиение сцены, критерием которого будет сведение ошибок «дискретизации» и «сходимости» к одному значению.

Для метода Монте-Карло ошибка «сходимости» может быть определена как дистанция между идеально правильной картой освещенности (яркости) и реально вычисленной картой. Принимая во внимание, что размеры и форма треугольников, на которых представлены карты, меняются по сцене, необходимо выполнить усреднение вычисленной дистанции по размеру сцены (суммарной площади всех треугольников сцены). Данная дистанция есть среднеквадратическое отклонение распределения глобальной освещенности (яркости) от идеальной карты и может служить оценкой ошибки моделирования. Выражение среднеквадратического отклонения имеет следующий вид:

$$\sigma = \sqrt{\frac{\int \left(E_0(x,y) - E(x,y)\right)^2}{S}},$$

где: E_0 – идеальная карта освещенности,

E – карта освещенности, выч. методом Монте-Карло,

S – суммарная площадь поверхности всех треугольников, на которых была рассчитана карта освещенности.

Основным недостатком вычисления выражения среднеквадратичного отклонения является невозможность применения его на практике. Идеальная карта освещенности (яркости) неизвестна, она есть результат моделирования при бесконечном времени вычисления. Поэтому в условиях реального моделирования в качестве оценки идеальной карты освещенности (яркости) используется карта, вычисленная при других (независимых) начальных значениях датчиков случайных чисел. Было принято решение для получения двух независимых карт освещенности (яркости) – сохранять

результаты трассировки четных лучей в «четной» карте, а результаты трассировки нечетных лучей – в «нечетной».

Выражение среднеквадратичного отклонения для оценки погрешности вычисления глобальной освещенности (яркости) примет следующий вид:

$$\sigma = \sqrt{\frac{\int \left(E_{нечет}(x,y) - E_{чет}(x,y)\right)^2}{S}},$$

где: $E_{нечет}$ – карта яркости, вычисленная для нечетных лучей,

$E_{чет}$ – карта яркости, вычисленная для четных лучей.

Глобальная карта освещенности (яркости) есть карта, сформированная нечетными и четными лучами.

3. ПРОГНОЗИРОВАНИЕ ВРЕМЕНИ РАСЧЕТА

Оценка точности моделирования является необходимым условием физически аккуратного оптического моделирования. Достаточным же условием является прогнозирование времени расчета, необходимого для достижения заданной точности. Для оценки необходимого времени моделирования можно использовать хорошо известную теоретическую зависимость:

$$\sigma_{rel} \approx \frac{1}{\sqrt{t}}.$$

Предполагая, что количество испущенных фотонов достаточно велико ($N \to \infty$) и элементы сцены накапливают независимые фотоны с единичной энергией, то выражение сводится к:

$$\sigma_{rel} = \frac{1}{\sqrt{n}},$$

где n – количество лучей, аккумулированных элементом сцены.

Принимая во внимание, что в среднем время на трассировку одного луча (фотона), приходящего на элемент сцены, меняется незначительно, можно заключить, что количество лучей, аккумулированных элементом, пропорционально времени моделирования. Очевидно, что высокая точность прогноза времени моделирования возможна при значительном количестве протрассированных лучей.

Алгоритмы оценки точности и прогнозирования времени моделирования не являются трудоемкими, поэтому оценка точности и прогноз могут вычисляться интерактивно, без видимой задержки времени моделирования. Графически анализ точности вычислений включает в себя график зависимости точности от времени вычислений (рис. 1, до настоящего времени показывается реальная зависимость, после настоящего времени – прогноз). На нем отображается текущее значение точности (цифра 2 на рис. 1), прошедшее время и количество протрассированных лучей (цифра 1), установленная пользователем желаемая точность (цифра 3) и прогнозируемое время для достижения заданной точности с прогнозируемым количеством лучей (цифра 4).

График прогнозируемого времени вычислений обновляется каждый раз по мере накопления новой порции данных. Начальный прогноз, сделанный через 17 секунд расчета (показанный на рис. 1), хотя и не является абсолютно точным (однопроцентное значение ошибки реально было достигнуто через 12 минут и 5 секунд, а не 11 минут и 49 секунд, как предсказано на рис. 1), он достаточно достоверен. Надо отметить, что прогноз времени вычисления достаточно точен с первых секунд расчета (он находится в пределах 2-3% от общего времени расчета), и по мере продолжения вычислений прогноз становится более достоверным.

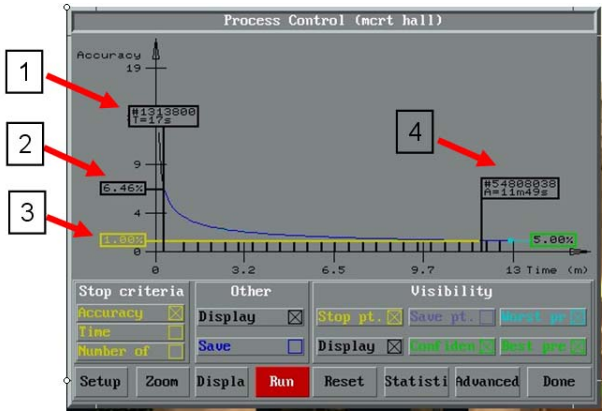


Рис. 1. График зависимости точности от времени вычислений с прогнозом времени для достижения заданной точности.

Алгоритмические и программные решения оценки и прогнозирования точности оптического моделирования были реализованы в программном комплексе Inspirer [9].

Работа поддержана грантами РФФИ № 07-01-00450 и № 08-01-00649, а также фирмой Integra Inc. (Япония).

4. ЛИТЕРАТУРА

[1] E.P.Lafortune, Y.D.Willems. Bidirectional path tracing. Computer Graphics Proceedings, Portugal, pp.145-153,1993.

[2] S.N.Pattanaik, S.P.Mudur. Adjoint equations and random walks for illumination computation, ACM Transactions on Graphics 14: 77-102, 1995.

[3] Inspirer: Simulation software for lighting design and analysis. http://www.integra.jp/en/inspirer/index.html

Abstract

The accuracy of physically based lighting simulation and realistic rendering is considered in the paper. The stochastic error is analyzed for the task of global illumination simulation. Also the issue of prediction of calculation time necessary to obtain requested simulation accuracy is discussed and practical approach to solve the issue is suggested.

Keywords: *realistic images, global illumination, illumination maps, accuracy estimation.*

Authors:

Alexey G. Voloboy, PhD, senior researcher, Keldysh Institute for Applied Mathematics RAS (KIAM RAS). E-mail: voloboy@gin.keldysh.ru

Sergey V. Ershov, PhD, senior researcher, KIAM RAS.

Dmitry D. Zhdanov, PhD, senior researcher, Vavilov State Optical Institute

Igor S. Potemin, researcher, KIAM RAS

Lev Z. Shapiro, PhD, senior researcher, KIAM RAS.

Каркасно-Сеточная Модель Тентовой Тканевой Конструкции в Системе К3-ТЕНТ

Попов Евгений Владимирович, Шалимов Владимир Николаевич, Шалимова Ксения Валентиновна
Нижегородский Государственный Архитектурно-Строительный Университет, Нижний Новгород, Россия
popov@sandy.ru; vsh11@mail.ru; kvz21@mail.ru

Тентовые тканевые конструкции (ТТК) за последние 20 лет получили весьма широкое распространение, т.к. имеют неоспоримые преимущества при перекрытии больших площадей, предназначенных для размещения торговых центров, выставочных павильонов, а также спортивных сооружений. В условиях острого дефицита городских площадей они также являются незаменимыми. ТТК состоят из комбинаций разнообразных сложных поверхностей двоякой кривизны, из-за чего имеют весьма эстетичные формы. Однако именно поэтому их проектирование представляет собой чрезвычайно сложную задачу. Сложность также заключается в том, что форма ТТК при заданном каркасе неизвестна заранее, в силу специфики тканевых материалов, представляющих собой тонкие пространственные мембраны с нулевой изгибной жесткостью и работающие только на растяжение. В мировой практике проектирования ТТК сложилась следующая технологическая последовательность: определение формы тканевой конструкции, анализ ее нагружения и построение карты раскроя конструкции [1].

В данной работе представлено описание каркасно-сеточной модели ТТК, принятой в системе автоматизированного проектирования К3-Тент. Система К3-Тент является дальнейшим развитием системы FABRIC CAD, описанной в работе [3]. Первоначально, в системе FABRIC CAD была принята математическая модель ТТК, основанная на возможностях NURBS аппроксимации поверхностей. Сочетание математического аппарата МНС с аппаратом NURBS аппроксимации показало высокую эффективность [3]. Дальнейшая модернизация модели ТТК выявила то, что в большинстве случаев реального проектирования данного типа конструкций принятая в FABRIC CAD идеология построения математической модели не обладает достаточной общностью. В частности, использование NURBS существенным образом сужает класс проектируемых конструкций, накладывая определенные ограничения на их форму. Преодоление данных ограничений приводит к заметному усложнению инструментария моделирования системы и, как результат, замедлению ее работы. Кроме того, проектирование ТТК заключается также в конструировании металлоконструкций, поддерживающих тентовое сооружение, что вызывает необходимость совместного использования системы FABRIC CAD с другими CAD системами. Это существенным образом удорожает работы, связанные с проектированием и производством. По этой причине было принято решение о создании нового типа математической модели ТТК, получившей название ***Каркасно-сеточной***. Данный тип математической модели является чрезвычайно простым и легко реализуемым на базе любой CAD-системы, использующей граничное представление моделей объектов. В качестве подобной системы была выбрана система К3, хорошо известная отечественным пользователям рядом своих эффективных приложений. Система К3-Тент является расширением системы К3.

Структурно система К3-Тент состоит из двух модулей. Первый модуль ***«Формообразование»*** предназначен для создания пространственной модели тента. Он позволяет задавать пространственные координаты закрепленных узлов и ребер тента, задавать граничные условия. На основании этой информации модуль в автоматическом режиме находит форму оболочки Методом натянутых сеток (см. [1], [2]). Второй модуль ***«Подготовка производства»*** предназначен для выполнения конструкторских работ, связанных с изготовлением тента. Модуль позволяет нанести линии разреза и линии маркировки на поверхность, разрезать поверхность на лоскуты, определить площади и длины сторон лоскутов, сформировать выкройки, т.е. развернуть лоскуты на плоскость, передать данные во внешние системы через графический формат.

Для создания ТТК, как одна из составных частей каркасно-сеточной модели, в систему К3-Тент введен специальный объект ***Каркас***. Каркас представляет собой набор связанных друг с другом пространственных линий, в качестве которых могут выступать отрезки прямых линий и дуги окружностей. Узлы каркаса, как правило, либо отвечают точкам крепления полотнища тента, либо являются точками, в которых полотнище изменяет свою геометрию. Объект Каркас, с математической точки зрения, представляет собой граф с заданным набором циклов. Граф в системе К3-Тент удовлетворяет следующим требованиям:

1. Ребра графа, в качестве которых выступают отрезки прямых линий и дуги окружностей, создаются стандартными средствами системы К3.
2. Граф, как объект типа "Каркас", создается командой "Создать", в процессе выполнения которой выделяются и классифицируются его вершины всех типов, вычисляются и запоминаются все характеристики вершин и ребер, включая звезды вершин. Характеристиками ребер Каркаса являются начальная и конечная вершины ребра и его тип (отрезок прямой, дуга окружности).
3. Замкнутые топологические треугольники и четырехугольники каркаса автоматически воспринимаются системой в качестве фрагментов полотнища. Однако при создании объекта Каркас система запрашивает Пользователя о необходимости включения или не включения каждого цикла графа треугольной или четырехугольной формы в состав объекта. Это делается для того, чтобы топологические треугольники или четырехугольники не формировались на тех участках конструкции, где полотнище конструктивно отсутствует.
4. В процессе моделирования конструкции может возникнуть необходимость изменения топологии графа (добавлении или исключении некоторых ребер и/или вершин). В этом случае предусмотрен режим перерегистрации Каркаса, позволяющий модифицировать каркас, изменяя или не изменяя его топологию.
5. Вершины графа могут подвергаться редактированию, то есть изменению их характеристик (координат). Изменение

координат вершин может производиться как с сохранением топологической связи ребер в вершине (сохранение звезды), так и без сохранения.

Каждому ребру и каждой вершине Каркаса приписывается соответствующее кинематическое граничное условие. Оно представляет собой характеристику ребра или вершины, отражающую ее поведение в процессе геометрического преобразования модели. Вершины Каркаса так же имеют граничные условия трех типов: вершина может быть свободной, а так же упруго или жестко закрепленной. Упругое закрепление вершины позволяет моделировать тросовые растяжки тканевого полотнища. Если конструкция имеет одну или несколько плоскостей симметрии, Каркас может быть составлен с учетом этих плоскостей, а соответствующие его ребра могут иметь граничные условия, учитывающие отсутствие перемещений, ортогональных той или иной плоскости симметрии. Данная возможность позволяет существенно сократить время на формирование итоговой каркасно-сеточной модели.

Объект ***Сеть*** представляет собой сеть, состоящую из треугольных ячеек, построенную на топологических треугольниках и четырехугольниках объекта Каркас. Треугольная сеть строится в системе К3-Тент в автоматическом режиме с последующей ее регуляризацией с использованием МНС, как это описано в работе [1]. Объект Сеть в системе К3-Тент представлен тремя структурами, каждая из которых содержит информацию о топологии и геометрии вершин, ребер и ячеек сети. С формальной точки зрения представление объекта Сеть соответствует классическому Граничному представлению (Boundary Representation, см. [5]). Свойства объекта Сеть наследуются из свойств объекта Каркас и служат основой для геометрического преобразования Сети на базе МНС.

Завершающим этапом формирования итоговой каркасно-сеточной модели ТТК является геометрическое преобразование объекта Сеть с использованием Метода натянутых сеток как это описано в работах [1], [2], [3]. Данное преобразование, имеющее в системе К3-Тент название «релаксация», направлено на отыскание новых координат узлов объекта Сеть, отвечающих условиям минимума ее энергетического функционала. Как это показано в [2], данная процедура эквивалентна отысканию такой псевдо-регулярной сети, узлы которой лежат на поверхности минимальной площади. Как отмечалось некоторыми исследователями (см., например, [6]) для проектирования ТТК в большинстве случаев является предпочтительным использование фрагментов поверхностей минимальной площади (так называемых минимальных поверхностей). Релаксация осуществляется на основе обобщенной формулировки МНС, предполагающей наличие внешних воздействий на систему в виде наложенных упругих узловых связей и узловых псевдонагрузок [4]. При необходимости данная модель может быть легко отредактирована путем изменения геометрии и/или топологии объекта Каркас. На рисунке приведен пример итоговой каркасно-сеточной модели тентовой тканевой конструкции.

Таким образом, в работе представлено описание каркасно-сеточной модели ТТК, принятой в системе автоматизированного проектирования К3-Тент, предназначенной для отыскания формы данного вида конструкций. Модель строится на базе пространственного каркаса, формируемого с помощью стандартного набора инструментов моделирования системы К3. Формируемый объект типа «каркас», состоящий из набора отрезков прямых линий и дуг окружностей, служит основой для построения предварительной сети с треугольными ячейками. Построенная сеть подвергается релаксации с помощью Метода натянутых сеток (МНС) совместно с заданными кинематическими граничными условиями, что позволяет получить модель тканевого полотнища. Разработанная каркасно-сеточная модель позволяет проектировщику быстро модифицировать каркас конструкции и переопределять в автоматическом режиме форму сети, моделирующей полотнище тента. Особенности данного типа модели, в сочетании с высокоэффективным инструментарием системы К3, позволяет осуществлять проектирование и производство тентовых тканевых конструкций практически любой сложности, как серийно, так и по индивидуальным проектам.

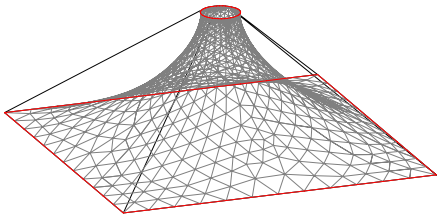


Рисунок: Итоговая Каркасно-сеточная модель тентовой тканевой конструкции.

СПИСОК ЛИТЕРАТУРЫ

[1] Popov, E.V. *On Some Variation Formulations for Minimum Surface. The Transactions of the Canadian Society for Mechanical Engineering, vol.20, N 4, 1996.*

[2] Попов Е.В. *Построение поверхностей минимальной площади с помощью метода натянутых сеток. // Международный межвузовский сб. трудов кафедр графических дисциплин, Вып. 5, Н.Новгород, 2000.*

[3] Попов Е.В., Тарасов А.И. *FABRIC CAD система проектирования Тентовых Конструкций // Proceedings of the 11th International Conference on Computer Graphics & Vision GRAPHICON'2001, UNN, Nizhny Novgorod, 2001.*

[4] Попов Е.В. *Метод натянутых сеток в задачах геометрического моделирования. Дисс. на соискание ученой степени доктора технических наук. //ННГАСУ. Н. Новгород. 2001.*

[5] Роджерс Д., Адамс Дж. *Математические основы машинной графики. М, изд-во «МИР», 2001.*

[6] Fujikake M. *Analysis of fabric tension structures. // Computers and Structures, N 32, 1982.*

FRAME AND GRID MODEL OF TENSILE FABRIC STRUCTURE IN K3-TENT SYSTEM

Abstract

The frame and grid model of tensile fabric structure is described in this paper. The model is based on spatial framework that can be developed by means of standard K3 CAD system toolkit. The Framework object consists of a set of straight lines and circle arcs and is a basis of a preliminary triangle grid. Such grid supplied with cinematic boundary conditions is a subject for relaxation by Stretched Grid Method. The frame and grid model allows the Designer to modify quickly structure frame and to redefine automatically the shape of grid that models tent surface. The model allows one to take into account the structure symmetry conditions.

Keywords

Tensile fabric structure, Framework, Stretched Grid Method.

Методы метапрограммирования в компьютерном зрении: 7-точечный алгоритм и автокалибровка

О.С.Сидоркина, Д.В.Юрин
Факультет вычислительной математики и кибернетики
Московского государственного университета им. М.В.Ломоносова, Москва, Россия
sidorkina_olga@mail.ru, yurin_d@inbox.ru

Аннотация

Неотъемлемой частью практически всех систем восстановления трехмерных сцен является 7-точечный алгоритм, линейная задача автокалибровки также часто используется в таких системах. Эти задачи включают в себя вычисление коэффициентов полинома (3 или 4 степени) задаваемого как детерминант линейной комбинации двух матриц. Показано, как с помощью методов метапрограммирования на C++, для этой специфичной для области компьютерного зрения технической задачи построить решение с помощью компактного и ясного кода, значительная часть которого выполняется в момент компиляции программы до ее исполнения. Результатом компиляции является близкий к оптимальному ассемблерный код без циклов и условных переходов, реализующий вычисления по явной формуле, построенной компилятором. С детальным пояснением методов и алгоритмов приведен полный и переносимый исходный код, готовый к использованию при решении практических задач.

Keywords: *7-ми точечный алгоритм, автокалибровка, метапрограммирование, C++*

1. ВВЕДЕНИЕ

Традиционным языком программирования в задачах компьютерного зрения и компьютерной графики является C++. Этот выбор обусловлен высокой вычислительной сложностью характерных задач. Однако, в настоящее время, наблюдается разрыв между огромными возможностями современных C++ и компиляторов и используемыми техниками программирования, зачастую берущими начало от языка C или «C с классами». Тем не менее, постепенно современные методы программирования проникают и в область компьютерного зрения, иногда – неявно, через используемые библиотеки, например матричных вычислений, таких как uBlas из библиотеки Boost [1], а в ряде случаев появляются разработки, специализированные для задач компьютерного зрения. В качестве таких положительных примеров можно отметить библиотеку VIGRA [2,3] и алгоритмы на основе минимизации энергии на графах [4,5], включенные в настоящее время в библиотеку Boost.

Настоящая работа посвящена некоторым техническим аспектам проблемы восстановления трехмерных сцен, а именно задачам вычисления фундаментальной матрицы по междукадровым точечным соответствиям [6, pp. 281, 291] с помощью 7-точечного алгоритма и автокалибровке с помощью абсолютной дуальной квадрики [6, p. 465]. В обоих случаях математически задача формулируется следующим образом. Ищется решение системы линейных уравнений

$$\mathbf{C}\vec{\mathbf{x}} = 0 \qquad (1)$$

с помощью сингулярного разложения как правые собственные векторы, принадлежащие нуль-пространству матрицы $\mathbf{C}$. Количество уравнений на два меньше, чем число переменных, поэтому в нуль-пространстве оказываются два собственных вектора и общее решение выражается их линейной комбинацией. Компоненты искомого вектора $\vec{\mathbf{x}}$ - суть элементы фундаментальной матрицы $\mathbf{F}$ для первой задачи и элементы матрицы абсолютной дуальной квадрики $\mathbf{Q}_\infty^*$ для второй. Для того чтобы свести задачу к линейной, в системе уравнений (1) наложены не все ограничения. Матрица $\mathbf{F}$ имеет размер 3x3, определена с точностью до множителя и имеет ранг 2, а матрица $\mathbf{Q}_\infty^*$ имеет размер 4x4, симметрична (это требование учтено при построении системы (1)) и удовлетворяет условию $\det \mathbf{Q}_\infty^* = 0$. Формируя из двух векторов нуль-пространства матрицы $\mathbf{C}$ две матрицы $\mathbf{X}_1$ и $\mathbf{X}_2$, решение, удовлетворяющее всем ограничениям, ищется как $\mathbf{X} = (1-\lambda)\mathbf{X}_1 + \lambda\mathbf{X}_2$, где λ удовлетворяет уравнению $\det((1-\lambda)\mathbf{X}_1 + \lambda\mathbf{X}_2) = 0$. Это алгебраическое уравнение 3 или 4 степени (для $\mathbf{F}$ или $\mathbf{Q}_\infty^*$ соответственно) можно переписать в виде:

$$\det(\mathbf{A} + \lambda\mathbf{B}) = 0 \qquad (2)$$

где $\mathbf{A} = \mathbf{X}_1$, $\mathbf{B} = \mathbf{X}_2 - \mathbf{X}_1$.

Однако здесь возникает техническая сложность: по известным матрицам $\mathbf{A}$ и $\mathbf{B}$ вычислить коэффициенты полинома (2). Обычно используется один из двух подходов. 1) написать численный алгоритм, 2) получить явные выражения для коэффициентов через элементы матриц с помощью программ аналитических вычислений (Maple, Mathematica). Первый подход обычно связан с выполнением «лишних» вычислений, что неприемлемо при погружении 7-точечного алгоритма внутрь процедуры RANSAC. Второй подход требует использования дорогостоящего программного обеспечения и ведет к объемному, трудночитаемому и некрасивому коду на C. В настоящей работе предлагается третий подход вычисления коэффициентов таких полиномов (2), основанный на технике метапрограммирования в C++ [7]. Суть подхода можно сформулировать как «***заставить компилятор построить явные выражения для коэффициентов полинома во время компиляции***».

2. АЛГОРИТМ

Объем публикации в формате POSTER не позволяет сколько-нибудь подробно представить материал. Однако полный 8-страничный вариант статьи может быть найден:

1. На диске с трудами конференции в электронной форме.
2. Будет выложен в трудах конференции на сайте http://www.graphicon.ru
3. На сайте лаборатории Математических методов обработки изображений по адресу http://imaging.cs.msu.su/~yurin/notes_on_CVision.html

Полный вариант статьи содержит исходные коды на C++ с детальными пояснениями всех алгоритмов, способами модификаций, применения и тестовыми примерами.

Приведенные листинги являются ***полными***, будучи собранными (Copy-Paste из текста статьи) вместе, компилируются, верно работают и могут быть использованы при решении практических задач, даже без детального понимания кода.

Просмотр дизассемблированного кода показывает отсутствие вызовов функций, циклов, условных и безусловных переходов. Кроме считывания значений элементов матриц, все вычисления проходят в регистрах SSE.

3. БЛАГОДАРНОСТИ

Работа выполнена при поддержке грантов РФФИ 09-01-92470-МНКС_а, 09-07-92000-ННС_а. Авторы выражают благодарность инженеру фирмы Sensor-IC (Москва, Зеленоград) Осипенко А.С. за плодотворные обсуждения методов сигнализации об ошибках во время компиляции и помощь в тестировании кода на UNIX-платформах.

4. ЛИТЕРАТУРА

[1] *Boost libraries*. http://ww.boost.org

[2] U.Köthe. *STL-Style Generic Programming with Images // C++ Report Magazine 12(1), pp. 24-30, January 2000.* http://kogs-www.informatik.uni-hamburg.de/~koethe/ .

[3] U.Köthe, K.Weihe. *The STL Model in the Geometric Domain // in: M. Jazayeri, R. Loos, D. Musser (Eds.): Generic Programming, Proc. of a Dagstuhl Seminar, Lecture Notes in Computer Science 1766, pp. 232-248, Berlin: Springer, 2000*

[4] Y.Boykov, V.Kolmogorov. *An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision // In IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 9, pp. 1124-1137, Sept. 2004* http://www.csd.uwo.ca/faculty/yuri/Abstracts/pami04-abs.html .

[5] http://www.boost.org/doc/libs/1_39_0/libs/graph/doc/kolmogorov_max_flow.html

[6] R.Hartley, A.Zisserman. *Multiple View Geometry in Computer Vision // Cambridge University Press, 2004. - 672 p.*

[7] А. Александреску. *Современное проектирование на C++ // Серия C++ In-Depth, т.3.:Пер. с англ. -М.:Издательский дом "Вильямс",2002. -336 с. ил. -Парал. тит. англ.*

[8] Г.Уоррен, мл. *Алгоритмические трюки для программистов // Пер. с англ. –М.:Изд. Дом «Вильямс», 2003. -288 с.:ил. –Парал. тит. англ.*

Об авторах

Сидоркина Ольга Станиславовна– аспирантка лаборатории Математических методов обработки изображений факультета Вычислительной математики и кибернетики Московского государственного университета им. М.В.Ломоносова. sidorkina_olga@mail.ru

Юрин Дмитрий Владимирович, к.ф.-м.н., с.н.с. лаборатории Математических методов обработки изображений факультета Вычислительной математики и кибернетики Московского государственного университета им. М.В.Ломоносова. yurin_d@inbox.ru

Metaprogramming Techniques in Computer Vision: 7-point Algorithm and Auto-calibration

Olga S. Sidorkina, Dmitry V. Yurin
Department of Computational Mathematics and Cybernetics
Moscow State University, Moscow, Russia
sidorkina_olga@mail.ru, yurin_d@inbox.ru

7-point algorithm is an essential part of almost all 3D reconstruction systems based on images set. Frequently, a linear auto-calibration algorithm is a part of such systems too. Both algorithms include coefficients calculation of 3-d or 4-th order polynomial defined as determinant of linear combination of two matrices. We show how with metaprogramming techniques in C++, this specific for computer vision problem can be solved with compact and clear code. A significant part of this code executes in the compile time. The result of the code compilation is near optimal assembler code without loops and jumps, implementing calculation with direct formula, constructed by compiler. Ready-to-use complete C++ code is given with detailed methods and algorithms description.

Keywords: *7-point algorithm, auto-calibration, metaprogramming, C++*

About the authors

Olga S. Sidorkina is a PhD student at Laboratory of Mathematical Methods of Image Processing, Chair of Mathematical Physics, Faculty of Computational Mathematics and Cybernetics, Moscow Lomonosov State University. Her contact email is sidorkina_olga@mail.ru

Dmitry V. Yurin, PhD, is a senior scientist at Laboratory of Mathematical Methods of Image Processing, Chair of Mathematical Physics, Faculty of Computational Mathematics and Cybernetics, Moscow Lomonosov State University. His contact email is yurin_d@inbox.ru

Создание модели системы однородных объектов и её описание средствами XML

Вячеслав Архипов, Василий Гончаренко
Объединённый институт проблем информатики НАН Беларуси

Резюме

В статье предлагается методика выделения на полутоновых изображениях рентгеновской компьютерной томографии однородных объектов и описания их системы с использованием языка XML.

Ключевые слова: *Сегментация изображений, система объектов, описание системы, XML.*

1. ВВЕДЕНИЕ

Задача данной статьи – описание процесса выделения на изображении рентгеновской компьютерной томографии костных объектов, с последующим созданием модели их системы, которая предполагает как описание геометрии каждого из объектов, так и их взаимосвязей.

Такая методика оптимальна для решения задач построения моделей систем однородных объектов, на основании данных томографического обследования для их дальнейшего использования в каких-либо расчетах или моделировании.

Также предлагаемая методика может успешно применяться для передачи данных между разными приложениями. Особенно, когда заранее не известно, какое приложение будет использовать создаваемую модель, и поэтому она должна быть описана в простом и универсальном виде.

Для описания модели используется язык описания структурированных данных XML

2. СЕГМЕНТАЦИЯ ИЗОБРАЖЕНИЙ РКТ

Задача выделения костей выполняется в два этапа [1]. Первый этап – пороговая сегментация: из исходного изображения создаётся бинарное. Порог, который используется на этом шаге, определяет минимальную плотность костной ткани.

Второй этап – дополнительная сегментация. Этот этап использует сегментацию, основанную на преобразовании водораздела, и маркировку объектов (в ручном режиме или автоматически). При этом используются значения только тех вокселов исходного изображения, которые на предыдущем этапе определились как принадлежащие костям. Преобразование водораздела это один из наиболее эффективных алгоритмов сегментации изображений, основанный на математической морфологии.

3. КОНТУРНОЕ ПРЕДСТАВЛЕНИЕ БИНАРНЫХ ОБЪЕКТОВ

Так как мы рассматриваем однородные объекты, то можем задавать их как замкнутую поверхность, которая, в свою очередь, может быть представлена как совокупность замкнутых контуров.

Контура лежат в плоскостях аксиальных срезов и разделяют вокселы принадлежащие объекты и фону в этих срезах. Контура могут быть внешними или внутренними. Пример выделения контурного представления слоя бинарного объекта показан на Рисунке 1.

Рисунок 1: Контурное представление аксиального среза бедренной кости.

4. ОТ МНОЖЕСТВА КОНТУРОВ – К МОДЕЛИ СИСТЕМЫ ОБЪЕКТОВ

4.1 Свойства системы контуров и объектов

Выделенные контура не произвольно разбросаны в пространстве, а подчинёнными определённым правилам:

- контура лежат в определённых параллельных плоскостях;
- контура вложены друг в друга;
- контура принадлежат объектам и определяют их геометрию;
- каждый контур представляет собой список точек, которые задаются своими координатами.

Для удобства описания системы объектов, откажемся от использования внутренних контуров. Будем их интерпретировать как внешние контура других объектов. При этом полость внутри объекта также будет интерпретироваться как объект.

Объекты также обладают набором свойств:

- объекты состоят из контуров, каждый объект можно интерпретировать как список контуров;
- объекты могут быть вложенными в другие объекты;
- все объекты обладают одинаковым набором свойств, которые могут иметь разное количественное значение у разных объектов. Например, для моделирования механических свойств системы необходимо задать значения плотности и коэффициента Пуассона каждого объекта.

4.2 Язык разметки XML

Описываемая система объектов может использоваться не только в контексте одного приложения, но также может экспортироваться между разными программами. Кроме того,

желательна возможность просмотра, проверки правильности и изменения состава системы объектов без использования сложного программного обеспечения.

Для этих целей идеально подходит язык разметки XML (Etensible Markup Language) – рекомендованный Косорциумом Всемирной паутины [2] язык разметки, фактически представляющий собой свод общих синтаксических правил [3, 4]. XML – текстовый формат, предназначенный для описания структурированных данных. Он предполагает использование тегов, для описания составных частей системы и имеет древовидную структуру.

4.3 Описание системы объектов на языке XML

Описание тегов XML задаётся по определённым правилам как описание типа документа или как схема. Для описания формата системы объектов мы используем правила описания типа документа DTD (Document Type Definition).

В Листинге 1 задан формат описания системы объектов, обладающих механическими свойствами. Вместо механических свойств в теге *material* могут описываться любые другие свойства (например, оптические, электрические и т.д.).

Листинг 1: Описание формата модели системы объектов.

```
MechObjSyst.dtd:
<!ELEMENT mechanical-object-system  (material-type, object-list,  layer-list,  contour_list)>

<!- описание типов материала-->
<!ELEMENT material-type (material+)>
<!ELEMENT material (puasson, density)>
<!ATTLIST    material id ID #REQUIRED>
<!ELEMENT puasson  (#PCDATA)>
<!ELEMENT density (#PCDATA)>

<!- описание объектов-->
<!ELEMENT object-list (object+) >
<!ELEMENT object (material_id, contour_list, neib_object_id*)>
<!ATTLIST    object id ID #REQUIRED>
<!ELEMENT material_id  (#PCDATA)>
<!ELEMENT contour_list (contour_id+) >
<!ELEMENT contour_id  ((#PCDATA)>
<!ELEMENT neib_object_id (#PCDATA)>

<!- описание слоёв-->
<!ELEMENT layer-list (layer+) >
<!ELEMENT layer (layer_depth, contour_id *)>
<!ATTLIST    layer id ID #REQUIRED>
<!ELEMENT layer_depth  (#PCDATA)>
<!ELEMENT contour_id  (#PCDATA)>

<!ELEMENT contour_list  (contour+) >
<!ELEMENT contour (object_id, layer_id, int_contour_id*,   ext_contour_id*, vertex_numb,
                                vertex_list+)>
<!- описание контуров-->
<!ATTLIST   contour id ID #REQUIRED>
<!ELEMENT object_id (#PCDATA)>
<!ELEMENT layer_id (#PCDATA)>
<!ELEMENT int_contour_id (#PCDATA)>
<!ELEMENT ext_contour_id (#PCDATA)>
<!ELEMENT vertex_numb (#PCDATA)>
<!ELEMENT vertex_list  (point+)>
<!ELEMENT point  (x, y)>
<!ATTLIST   point id ID #IMPLED>
<!ELEMENT x  (#PCDATA)>
<!ELEMENT y  (#PCDATA)>
```

Таким образом, мы описываем типы материала, из которого «сделаны» объекты, каждый тип идентифицируется своим уникальным номером. такие же уникальные номера имеют все элементы модели: объекты, слои и контура вершины контуров могут не обладать уникальным идентификатором. Затем мы описываем объекты, для каждого из которых указываем номер соответствующего материала, список номеров контуров, а также можем указать список номеров соседних объектов. Затем описываем слои, каждый из них содержит список контуров.

В последнюю очередь описываются контура, каждому из них соответствует номер слоя и номер объекта в который они входят. Контур может содержать также списки номеров внутренних и внешних относительно него контуров. Последний уровень описания: список вершин, каждая из которых описывается парой объектов.

5. ЗАКЛЮЧЕНИЕ

Предложенное описание систем трёхмерных однородных объектов удобно при экспорте и хранении данных. Использование языка XML позволяет проверять правильность структуры файла с помощью стандартного программного обеспечения для работы с XML. С помощью этого же программного обеспечения можно изменить состав системы, например, выделить из неё один объект.

Не смотря на всю «избыточность» описания, размеры файла, с описанием системы объектов не превышает, а скорее наоборот, – уступает, DICOM файлам. Но, в отличие от них, содержит не только данные о локальных свойствах точек, но и обо всех взаимосвязях элементов системы.

Если размеры файлов критичны, и нужно более компактное описание, можно сократить имена тегов.

Подобное (но немного упрощённое) описание системы объектов успешно использовалось при выполнении научно-исследовательского проекта ИНТАС [5], в рамках которого разрабатывалась система обработки изображений РКТ. Из исходного изображения выделялись кости, а затем эти данные экспортировались в другую систему, которая моделировала разрушение костей под воздействием динамических нагрузок.

6. ЛИТЕРАТУРНЫЕ ИСТОЧНИКИ

[1] V. Goncharenko, A. Tuzikov. Watershed Segmentation with Chamfer Metric. // Lecture Notes in Computer Science, vol. 4245/2006, Springer Berlin/Heidelberg, 2006, pp.518-529. (Proceedings of the 13th International Conference on Discrete Geometry for Computer Imagery, October 25-27, 2006, Szeged, Hungary).

[2] http://www.w3.org/

[3] http://www.w3.org/TR/REC-xml

[4] http://www.w3.org/XML/

[5] V. Hancharenka, V. Arkhipau, O. Krivonos, A. Tuzikov. A system for preoperative planning of pelvic and lower limbs surgery, // Proceedings of the 21st International Congress on Computer Assisted Radiology and Surgery (CARS 2007), vol. 2., suppl. 1, 2007, p. 494.

Тестирование вычислительных возможностей метода SPH для моделирования физических эффектов в компьютерной графике

Светлана Александровна Кукаева, Вадим Евгеньевич Турлапов
Нижегородский государственный университет им.Н.И. Лобачевского, Н.Новгород, Россия
kukaeva_svetlana@mail.ru, vadim.turlapov@cs.vmk.unn.ru

Abstract

Modeling of liquid and heat exchange using method SPH is considered. Structure for acceleration was realized (uniform grid). Few effects: Falling of liquid column and solid body to liquid, heat exchange in body and melting, are numerically solved. Speed of work on CPU dependence from number of particles was investigated. Results of numeric computations are presented. Results allow to modeling about 2000-3000 particles in real-time on CPU.
Key Words: *SPH, modeling of liquid body, heat exchange, ice melting*

1. ВВЕДЕНИЕ

Существует ряд задач, который имеет большое прикладное значение для компьютерной графики: моделирование жидкости, моделирование взаимодействия двух тел, исследование процесса теплопередачи. Известно большое количество методов, используемых для этой цели. В данной работе за основу взят метод Гидродинамики Сглаженных Частиц (SPH – Smoothed Particle Hydrodynamics), как единственный метод решения таких задач. Его применяют для исследования больших деформаций эластичных тел [1], процессов теплопередачи и отвердевания [2], моделирования жидкостей и газов [3], [4], [5], [6], и др. Все это применяется в игровой индустрии, спецэффектах кино и т.д. Цель настоящей работы – реализовать для многоядерных CPU и протестировать вычислительные возможности метода SPH для моделирования перечисленных выше физических эффектов в компьютерной графике.

2. ТЕСТИРОВАНИЕ МЕТОДА SPH

Метод SPH (разработан Люси в 1977, и независимо Джинголдом и Монаганом в 1977) - лагранжевый метод, в котором для моделирования используются частицы, для каждой из которых в любой момент времени известны масса, положение, скорость, а также плотность и давление.

Итак, пусть у нас есть частица x_i с массой m_i и некоторым свойством A_i. Тогда, если известны значения этого свойства для частиц, находящихся в некоторой окрестности x_i (в пределах так называемых двух сглаживающих длин), то A_i будет вычисляться как

$$(1)\quad A(x) = \sum_j m_j \frac{A_j}{\rho_j} W(x - x_j, h)$$

Где ρ_j - плотность частицы, W – сглаживающая функция, h – длина сглаживания. В формуле (1) и далее предполагается, что суммирование ведется по всем частицам, находящихся в пределах двух сглаженных длин.

Сглаживающая функция такова, что

$$\int_{x-x_i} W(x - x_i, h)dx = 1$$

$$W(x - x_i, h) = 0, |x - x_i| > 2h$$

В качестве тестовых в данной работе приняты задачи падения столба жидкости и тела (более подробно алгоритм решения описан в [4], [5]), теплопередача, таяние льда [2]. В рамках решения данных задач решаются уравнения Навье-Стокса (2) и теплопередачи (3).

$$(2)\quad \frac{dv}{dt} = -\frac{\nabla p}{\rho} + \mu\frac{\nabla^2 v}{\rho} + \frac{F}{\rho}$$

$$(3)\quad c_p \frac{dT}{dt} = \frac{1}{\rho}\nabla(k\nabla T)$$

В классической реализации данные алгоритмы имеют оценку сложности $O(n^2)$.

Таблица 1. Результаты тестирования классического метода SPH

Количество частиц	FPS*		
	Падение столба жидкости	Теплопередача	Таяние льда
1000	20	16	14
1500	14	10	9
2000	7	6	4
2500	6	4	3
3000	4	3	2

Таблица 2. Результаты тестирования метода SPH с использованием OpenMP

Количество частиц	FPS**		
	Падение столба жидкости	Теплопередача	Таяние льда
1000	47	46	47
1500	31	24	33
2000	25	21	25
2500	20	15	14
3000	15	13	11

Для ускорения вычислений использовался алгоритм, основанный на равномерной сетке. Пусть моделируемая сцена разбита на k ячеек сетки. В этом случае для каждой частицы i необходимо просмотреть все частицы, находящиеся в одной и в соседних ячейках с ней

Таблица 3. Зависимость скорости работы программы от среднего числа частиц в одной ячейке сетки*.

1000	500	100	50	20
20	20,5	21	22	24

Пусть в среднем в каждой ячейке находится c частиц. Тогда для расчета всех сил, действующих на каждую

* на процессоре Intel(R) Celeron(R) CPU 2.6GHz 768 Mb RAM
** на процессоре Intel(R) Core(TM)2 E6600 2.4GHZ, 3.25 Gb RAM

частицу, необходимо затратить время порядка *O(nc)*.Чем больше величина *c*, тем хуже эта оценка. Следовательно, лучших (по вычислительным затратам и по точности) результатов можно добиться в том случае, когда сторона одной ячейки сетки принимается равной *2h*. Для данной реализации алгоритма так же были протестированы результаты.

Таблица 4. Результаты тестирования метода SPH с использованием равномерной сетки

Количество частиц	FPS*		
	Падение столба жидкости	Теплопередача	Таяние льда
1000	24	19	21
1500	14	11	12
2000	9	8	7
2500	6	6	6
3000	5	5	4

Таблица 5. Результаты тестирования метода SPH с использованием равномерной сетки и OpenMP

Количество частиц	FPS**		
	Падение столба жидкости	Теплопередача	Таяние льда
1000	47	47	45
1500	33	31	28
2000	28	25	23
2500	22	20	17
3000	17	18	16

В следующей таблице приведены обобщенные результаты тестирования метода SPH.

Таблица 6. Средняя оценка вычислительных возможностей метода SPH

Количество частиц	FPS**			
	Без использования OpenMP		С использованием OpenMP	
	Без сетки	С сеткой	Без сетки	С сеткой
1000	26	29	46,7	46,3
1500	16	19	29,3	30,7
2000	12,3	16	23,7	25,3
2500	9	11,9	16,3	19,7
3000	7,3	8,7	13	17

2.1.Результаты моделирования

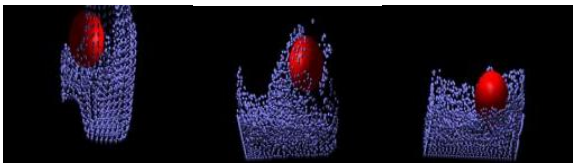


Рисунок 1. Падение тела, столба жидкости, 1800 частиц (см. табл.5)

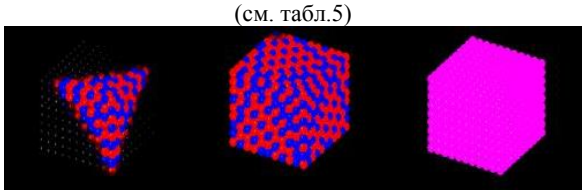


Рисунок 2. Теплопередача в теле,1089 частиц (см. табл.5). Температура тела показана цветом, синим цветом обозначены участки с более низкой температурой, красным – с более высокой. Нагревание ведется от точечного источника тепла

Рисунок 3. Таяние льда, 1089 частиц (см. табл.5). Белым цветом показан лед, синим – образовавшаяся в ходе таяния вода. Нагревание происходит в результате контакта с воздухом.

3. ЗАКЛЮЧЕНИЕ

Реализован алгоритм SPH, позволяющий также использовать ускоряющую структуру (равномерную сетку), для решения задач падения столба жидкости, исследования процессов теплопередачи, таяния льда. Протестированы вычислительные возможности реализованного алгоритма на CPU в зависимости от исследуемой проблемы. Для визуализации использовалась простейшие средства OpenGL. Представлены результаты вычислительных экспериментов. Показано, что в реализации на CPU с использованием многоядерности, метод SPH может обеспечить реальное времени моделирования во всех рассмотренных задачах примерно для 2000-3000 частиц. Результаты немного превосходят аналогичные исследования [7]. Результаты работы [3] показывают, что реализация SPH метода на GPU способна увеличить производительность в несколько раз.

4. СПИСОК ЛИТЕРАТУРЫ

[1] Yoichi Kawashima and Yuzuru Sakai. Large Deformation Analysis of Hyperelastic Materials Using SPH Method, e-Journal of Soft Materials, Vol. 3(2007), 21–28

[2] Joseph J. Monaghan, Herbert E. Huppert , M. Grae Worster. Solidification using smoothed particle hydrodynamics, Journal of Computational Physics 206 (2005), 684–705

[3] Yanci Zhang, Barbara Solenthaler, Renato Pajarola. Adaptive Sampling and Rendering of Fluids on the GPU. IEEE/ EG Symposium on Volume and Point-Based Graphics, 2008

[4] Marcus Vesterlund. Simulation and Rendering of a Viscous Fluid using Smoothed Particle Hydrodynamics, 3rd December, 2004

[5] Matthias Müller, Barbara Solenthaler, Richard Keiser, Markus Gross. Particle-Based Fluid-Fluid Interaction. Eurographics/ACM SIGGRAPH Symposium on Computer Animation (2005), pp. 1–7

[6] Frank Losasso, Jerry Talton, Nipun Kwatra, Ronald Fedkiw**.** Two-Way Coupled SPH and Particle Level Set Fluid Simulation. IEEE Transactions on Visualization and Computer Graphics, Issue 4 (July 2008) Volume 14 , Pages 797-804,2008

[7] Distributed SPH Fluid Simulator (for Hewlett-Packard Scalable Visualization Array). Budapest University of Technology and Economics. Department of Control Engineering and Information Technology. 31p. 2008. http://www.iit.bme.hu.

* на процессоре Intel(R) Celeron(R) CPU 2.6GHz 768 Mb RAM

** на процессоре Intel(R) Core(TM)2 E6600 2.4GHZ, 3.25 Gb RAM

Detection of iris in image by interrelated maxima of brightness gradient projections

Ivan Alexeevich Matveev, Dorodnicyn Computing Centre of RAS, matveev@ccas.ru

Abstract

A method is proposed to detect a human iris location and size in digital image given some point lying inside the pupil. Method is based on construction of histogram projections of local brightness gradients and interrelating local maxima of these histograms as probable positions of pupil and iris borders. Method is characterized by low calculation cost and high stability against noise.

Keywords: *iris identification, image processing, brightness gradient, image projection*

1. INTRODUCTION

Recognition of human by iris is one of the most demanded biometric technologies. Algorithm of iris size and position estimation is an essential part of iris registration systems. Since outer borders for both pupil and iris can be approximated by circles with good precision, circle detection is a central element of any system of iris detection in image. There are plenty of methods of circle (or circumference) detection implemented and tested for this task: detecting of mass center of an object selected by thresholding function [1], detection of a point most remote from borders of such selected object [2], maximizing of integro-differential circular symmetric operator [3], generalized [4] and split [6] Hough transform, Hough transform using brightness gradient [8], brightness gradient projection method [9], paired gradient vectors [10], circular shortest path construction [7], restoring centers of circles passing through randomly selected points [5]. However, so far the fact was not employed that iris border contains two circles (pupil-iris and iris-sclera borders) with interrelated parameters. Synchronous detection of two circles with parameters subject to certain mutual restrictions, implied by nature of iris allows substantially enhance algorithm characteristics in comparison with search of single circle. A proposed algorithm of iris location is based on construction of histograms (local brightness gradient circular projections) and comparisons of their maxima as possible positions of iris borders.

2. CONSTRUCTION OF CIRCULAR PROJECTIONS OF BRIGHTNESS GRADIENT

As in majority of recognition tasks the problem of iris location can be treated as a problem of selection of best positions of the two circles from a set of alternatives. These alternatives are given by positions of maxima of circular projections of brightness gradient. These projections are constructed relative to an approximate iris center, as detected in [9].

Input data for the algorithm are monochrome eye image and an approximate position of eye center. Irises with diameter not exceeding image size can be detected.

Denote: $\mathbf{c} = \begin{pmatrix} c_x & c_y \end{pmatrix}^T$ be a point of approximate center position. Method [9] guarantees its distance to real pupil center is not greater than half of pupil's radius. For simplicity consider this point as a coordinate origin. $\mathbf{x} = \begin{pmatrix} x & y \end{pmatrix}^T$ is a point vector, $b(\mathbf{x})$ is a brightness (intensity) in this point, $\mathbf{g}(\mathbf{x}) = \nabla b(\mathbf{x})$ is a brightness gradient.

Only points with certain gradient value and direction can belong to iris border. This set is described by indicator function:

$$v_U(\mathbf{x}) = \begin{cases} 1, \ \|\mathbf{g}\| > T_1, \ T_2 < \dfrac{\mathbf{x} \cdot \mathbf{g}}{\|\mathbf{x}\| \ \|\mathbf{g}\|} < T_3, \ U \\ 0, \ otherwise \end{cases},$$

where T_1, T_2, T_3 are thresholds established according to characteristics of input image, U is an additional condition selecting a sector (quadrant) of the coordinate plane. For instance, right quadrant is given by condition $U \equiv R : |x| > |y|, x > 0$. Taking one of such conditions for $v_U(\mathbf{x})$ one can obtain a histogram of number of points satisfying the conditions as a function of radius. For example the following is a histogram of right quadrant normalized to radius:

$$\Pi_R(r) = \frac{1}{2\pi r} \sum_{r-0.5 < \|\mathbf{x}\| < r+0.5} v_R(\mathbf{x}).$$

Fig.1. represents eye image and its right quadrant histogram $\Pi_R(r)$. Eight local maxima positions $\arg loc \max_{n,r} \Pi_R(r)$, n=1…8 are outlined in the histogram there.

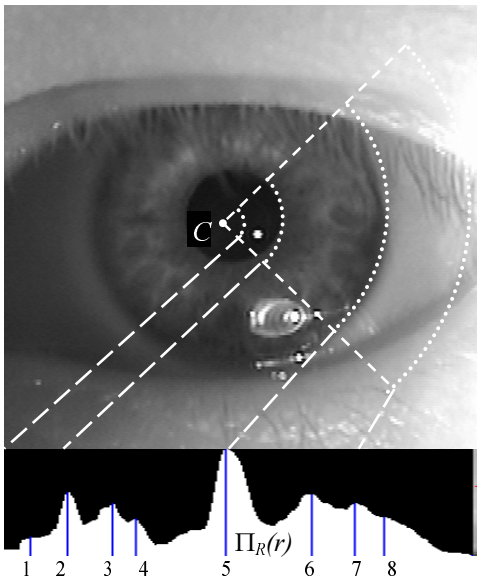


Fig.1. **Sample of circular projection and local maxima positions**

After detecting local maxima positions for all four quadrants one can obtain distances to hypothetic circle borders from central point in appropriate direction.

Combining these values one can obtain coordinates of centers $\mathbf{q} = \begin{pmatrix} q_x & q_y \end{pmatrix}^T$ and radii ρ of these circles:

$$q_x^{n,m} = \frac{1}{2}\left(\arg loc \max_{n,r} \Pi_R(r) - \arg loc \max_{m,r} \Pi_L(r)\right)$$

$$q_y^{u,v} = \frac{1}{2}\left(\arg loc \max_{u,r} \Pi_T(r) - \arg loc \max_{v,r} \Pi_B(r)\right)$$

$$\rho^{n,m,u,v} = \frac{1}{4}\begin{pmatrix} \arg loc \max_{n,r} \Pi_R(r) + \arg loc \max_{m,r} \Pi_L(r) + \\ \arg loc \max_{u,r} \Pi_T(r) + \arg loc \max_{v,r} \Pi_B(r) \end{pmatrix}$$

Quality of circle obtained for four given positions of local maxima (n, m, u, v) may be estimated as sum of projection function values in these positions.

3. SELECTION OF INTERRELATED HISTOGRAM MAXIMA

So, various hypothetic circles are constructed by a method of circular projections. The circles can be hypothetic pupils (index P is used further) or irises (index I). Circles can be defined by their parameters, center position and radius: $(\mathbf{q}_P, r_P)$ и $(\mathbf{q}_I, r_I)$. If two circles are the borders of an iris the following limitations due to human iris nature are necessarily true:

1) $r_P > \frac{1}{6} r_I$ (iris radius cannot exceed pupil more than six times)

2) $r_P < \frac{3}{4} r_I$ (pupil radius cannot be bigger than 75% of iris)

3) $d < r_P$, $d = \|\mathbf{q}_P \quad \mathbf{q}_I\|$ iris center lies inside pupil circle

4) $2(r_I - r_p - d) > r_I - r_P + d$, or after reduction $d < \frac{r_I - r_P}{3}$, (Lengths of segments between pupil and iris borders cut by a line passing through pupil and iris centers do not differ by more than two times).

From all pairs of circles satisfying (1-4) the one is selected with maximum sum of quality values.

Thus algorithm in whole consists of four steps:

- calculation of local gradients in image
- building circular projections (histograms) for four quadrants
- selecting local maxima in histograms
- enumeration of circle combinations searching most likely (with biggest quality) pair.

4. EXPERIMENTS

The following databases from public domain were used for performing experimental study: UBIRIS (http://www.di.ubi.pt/~hugomcp/doc/ubiris.pdf , 1207 images), CASIA Iris Image Database (http://www.sinobiometrics.com , 16213 images) and Iris Challenge Evaluation (http://iris.nist.gov/ice/ , 2954 images). Size of these images is 640*480 pixels, iris radii vary from 50 to 200 pixels.

Testing method. Eye images were reviewed by human expert who indicated pupil and iris borders in each of them. These data were then considered as true and were used for method verification. Then images were processed automatically. The approximate eye center was detected by method [9] (this point rarely matches true pupil center or true iris center, but is always close to them). With the help of method proposed here pupil and iris were detected. Their parameters were compared with those indicated by human operator. Table below gives numbers of rude errors (difference in any one of center coordinates or radii exceeds 10 pixels) and moderate (difference in any one of center coordinates or radii exceeds 5 pixels) errors. If all parameter values differ from true one not more than by five pixels, detection is considered correct.

Table. Results of algorithm for test databases.

Data base	Image count	Number of moderate errors in pupil	Number of rude errors in pupil	Number of moderate errors in iris	Number of rude errors in iris
UBIRIS	1207	315	3	32	1
CASIA	16213	1916	31	212	20
ICE	2954	112	7	9	2

Execution of the algorithm takes not more than 0.01 second in PC with P-IV 3GHz CPU for an image of 640*480 pixels. Main share of calculation time is taken by Sobel gradient estimation.

Proposed method of iris location may be applied for preliminary determination of pupil (with precision up to 5 pixels) and iris (with precision up to 10 pixels) positions if a point lying inside pupil is known. Method is useful for real-time applications.

5. LITERATURE

1. Maenpaa T. An Iterative Algorithm for Fast Iris Detection // IWBRS 2005. Beijing, China: Beijing Press, P. 127-134.
2. Lipinski B. Iris Recognition: Detecting the pupil // http://cnx.org/content/m12487/latest/
3. Daugman J. High confidence personal identification by rapid video analysis of iris texture // Proc. IEEE Internat. Carnahan conf. on security technology, 1992. P. 50-60.
4. Wildes R.P., Asmuth J.C., Green G.L. et al. A system for automated iris recognition // Proc. of the 2nd IEEE Workshop on Applications of Computer Vision. 1994. P. 121-128.
5. T.-C.Chen, K.-L.Chung An Efficient Randomized Algorithm for Detecting Circles // Computer Vision and Image Understanding 83, 172–191 (2001)
6. Benn D.E., Nixon M.S., Carter J.N. Robust eye centre extraction using the Hough Transform // Proc. 1st Int. Conf. on Audio- and Video-Based Biometric Person Authentication, 1997
7. C.Sun, S.Pallottino Circular Shortest Path in Images // Pattern Recognition, Vol. 36, No.3, pp.709-719, March 2003
8. Davies E.R. A high speed algorithm for circular object location // Pattern Recognition Letters. 1987. V.6. P. 323-333.
9. Matveev I.A. Method of detection of circle with known internal point. // Transactions of Institute of System Analysis of RAS on Dynamics of Heterogeneous Systems, 2007. V. 31(1) P. 288-293 (in Russian).
10. A.A.Rad, K.Faez, N.Qaragozlou Fast Circle Detection Using Gradient Pair Vectors // Proc. VIIth Digital Image Computing: Techniques and Applications, Sun C., Talbot H., Ourselin S. and Adriaansen T. (Eds.), 10-12 Dec. 2003, Sydney

Исследование точности оценки углов поворота лица по монокулярному цифровому изображению

Сергей Каратеев, Владимир Князь, Юрий Визильтер, Ирина Бекетова, Сергей Желтов
Государственный научный центр РФ
Государственный научно – исследовательский институт авиационных систем (ГосНИИАС)

АННОТАЦИЯ

Исследуется точность оценки углов поворота лица по монокулярному цифровому изображению. Разработан подход к оценке точности, основанный на сравнении синтезированных изображений лица с реальными трехмерными моделями. Создано программное обеспечение для моделирования и измерений, собран специализированный комплекс для бесконтактных измерений, и с использованием данного комплекса получены трехмерные модели и фотореалистические текстуры необходимого числа тестовых лиц. Осуществлены первичные измерения точности оцениваемых параметров с использованием предложенной методики.

Keywords: *Biometrics, face 3D reconstruction, facial imagery modeling, facial imagery analysis.*

1. ВВЕДЕНИЕ

Задача определения ориентации объекта и в частности лица на монокулярном цифровом изображении часто возникает при анализе цифровых изображений. В данном случае необходимость определения ориентации лица связанна с задачей формирования фронтального изображения лица для биометрических персональных документов.

Как следует из требований ГОСТ ИСО/МЭК 19794-5-2006, при получении фотографии лица на биометрические документы допустимый угол поворота изображения лица (головы) вокруг любой из осей координатной системы не должен превышать 5 градусов.

Вообще говоря, построение связанной трёхмерной системы координат на основании двухмерного изображения лица является некорректной обратной задачей. Поэтому в системах подготовки и контроля цифровых фотографий для биометрических документов для определения углов поворота относительно пространственных осей используются косвенные методы оценки углов поворота головы по изменению пропорций лица.

2. СИСТЕМЫ КООРДИНАТ

Для определения истинных углов поворота изображения необходимо ввести две системы координат – связанную с головой и опорную. Связанная с головой система координат XsYsZs определяется двумя ортогональными плоскостями: франкфуртской (или глазнично-ушной) горизонталью– плоскостью XsZs проходящей через верхние края отверстий наружного слухового прохода и нижнюю точку нижнего края левой орбиты и плоскостью YsZs –перпендикулярной плоскости XsZs и проходящей через ось симметрии лица (центр переносицы, центр губ). Ось Zs совпадает с линией пересечения плоскостей и направлена от поверхности лица

Опорная система координат XYZ определяет виртуальную камеру, формирующую 2D изображение и представляется следующим образом ось Y – параллельна вертикальной оси плоскости изображения, ось X – параллельна горизонтальной оси плоскости изображения, ось Z дополняет до правой системы координат и направлена от модели лица, центр расположен в середине 2D изображения. Изображения считается фронтальным, когда оси опорной и связанной систем координат колинеарны.

3. МЕТОДИКА МОДЕЛИРОВАНИЯ

Для оценки точности определения углов поворота изображения лица предлагается использовать трёхмерные модели лиц, получаемые путем трехмерного сканирования.

Первым шагом в оценке точности является получение трёхмерной модели лица человека. Для определения ошибок определения угловых координат модель поворачивается вокруг осей опорной системы координат на заданные углы, и строится ее проекция на фокальную плоскость виртуальной камеры, ориентированной относительно опорной системы координат. Эта проекция считается изображением повёрнутого лица, по которому осуществляются оценки углов поворотов изображения с помощью косвенных методов измерения, реализованных в программе оценки соответствия изображения лица требованиям ГОСТ.

4. МЕТОДИКА ИЗМЕРЕНИЙ

Для получения трёхмерной модели лица используется специализированный комплекс бесконтактных измерений, построенный на фотограмметрических принципах, позволяющих рассчитать координаты заданной точки объекта по двум разноракурсным изображениям объекта, наблюдаемых стереосистемой видеокамер.

Система бесконтактных измерений включает:

- Две камеры для технического зрения, предназначенные для захвата черно-белых изображений человека в структурированном свете и расчета трехмерных координат поверхности лица.
- Цветную фотокамеру высокого разрешения для получения цветного изображения и фотореалистичного текстурирования трехмерной модели
- Портативный DLP проектор для создания ПК-управляемого подсвета, обеспечивающего автоматизацию решения задачи стереоотождествления
- Персональный компьютер

Предварительным этапом работы с фотограмметрическим комплексом бесконтактных измерений является калибровка

[1,2], то есть оценка параметров модели камер, учитывающих нелинейные искажения, возникающие при формировании изображений камерой. Калибровка системы позволяет обеспечить высокую точность измерений трехмерных координат объекта.

Внешнее ориентирование системы выполняется для оценки положения и ориентации камер в системе координат, задаваемой специальным тестовым объектом. В результате процедуры ориентирования, выполняемой по снимкам тестового сюжета, определяются координаты и углы поворота камер в заданной системе координат. В дальнейшем координаты точек поверхности объекта рассчитываются в системе координат, заданной ориентированием системы.

Для расчета трехмерных координат поверхности объекта и построения его трехмерной модели необходимо для каждой видимой точки объекта (A) найти ее координаты на левом (a_l) и правом (a_r) изображениях (решить задачу стереоотождествления точек левого и правого изображений). Тогда, с использованием результатов ориентирования стереосистемы (положение камер), рассчитываются трехмерные координаты точки A.

Для автоматизированного решения задачи стерео-отождествления соответственных точек изображений с левой и правой камеры в системе применяется оригинальный кодированный подсвет объекта, минимизирующий количество кадров, использующихся для расчета трехмерных координат поверхности объекта, при сохранении высокой плотности измерений.

Основные технические характеристики системы:

- Время сканирования – ~0.5 сек
- Время расчета трехмерной модели – 5 сек
- Плотность измерения координат – 10-25 точек на мм^2
- Точность измерения пространственных координат – 0,5 мм

Система выполняет следующие функции:

• Сканирование и получение необходимого количества снимков лица для последующего использования при построении трехмерной модели лица

- Построение высокоточной трехмерной модели лица
- Текстурирование полученной трехмерной модели

Для расчета трехмерных координат поверхности используются последовательности снимков с черно-белых камер, а в качестве текстуры служит цветной снимок высокого разрешения, получаемый с цифрового фотоаппарата. Текстурирование трехмерной модели выполняется автоматически на основе данных ориентирования цифрового фотоаппарата.

5. ЗАКЛЮЧЕНИЕ

Данная работа находится на завершающей стадии исследований. К настоящему моменту разработан подход к моделированию и измерениям, создано программное обеспечение для моделирования и измерений, собран специализированный комплекс для бесконтактных измерений, и с использованием данного комплекса получены трехмерные модели и фотореалистические текстуры необходимого числа тестовых лиц. Осуществлены первичные измерения точности оцениваемых параметров с использованием предложенной методики. Предварительные оценки точности составляют порядка 1° для углов поворота относительно оси Z и Y, и порядка 5° относительно оси X. Худшая оценка точности при определении угла относительно оси X связана с высокой вариабельностью вертикальных пропорций лица у различных людей.

Финальная статистическая обработка полученных результатов позволит уточнить эти предварительные результаты, а также выбрать наилучшие методы оценки углов поворота лица на изображении и на основании оценки точности этих методов принять решение о пригодности их для использования в системе формирования цифровых фотографий для биометрических документов.

6. ЛИТЕРАТУРА

[1] Schenk T.F. Digital Photogrammetry, TerraScience, 1999, ISBN 0967765307.

[2] Luhmann T., Robson S., Kyle S., Harley I., Close Range Photogrammetry, Principles, Methods and Applications. Whittles, 2006, ISBN 1870325508, 510 pp.

Об авторах

Каратеев Сергей Львович, начальник сектора лаборатории компьютерного машинного зрения ГосНИИАС. e mail goga@gosniias.ru

Князь Владимир Александрович начальник лаборатории цифровых видеометрических методов, трехмерных измерений и виртуальной реальности ГосНИИАС, e mail knyaz@gosniias.ru.

Визильтер Юрий Валентинович, кандидат технических наук, начальник лаборатории компьютерного машинного зрения ГосНИИАС e mail viz@gosniias.ru

Бекетова Ирина Валентиновна ведущий инженер лаборатории компьютерного машинного зрения ГосНИИАС. e mail irus@gosniias.ru.

Желтов Сергей Юрьевич, доктор технических наук, профессор, член-корреспондент РАН, Генеральный директор ФГУП ГосНИИ Авиационных систем (ГосНИИАС

Exploration of Precision of Face Orientation Estimation Based on Monocular Digital Imagery

ABSTRACT

The precision of face orientation estimation based on monocular digital imagery is addressed. The approach for precision estimation is developed based on comparison of synthesized facial 2D images and scanned face 3D model. The software for modeling and measurement is developed. The special system for non-contact measurements is created. Required set of 3D real face models and colored facial textures are obtained using this system. Some preliminary precision estimation results are obtained.

Keywords: *Biometrics, face 3D reconstruction, facial imagery modeling, facial imagery analysis.*

Young Scientists School

GraphiCon'2009

October 5-9, 2009
Moscow, Russia

Улучшение качества изображения с микроскопа при помощи технологии HDRI в интерактивном режиме

Матросов Михаил, Гаганов Виктор, Игнатенко Алексей
Лаборатория компьютерной графики и мультимедиа факультета ВМиК, Московский государственный университет им. М.В. Ломоносова, Москва, Россия
E-mail: fsgs2k@gmail.com, {vgaganov, ignatenko}@graphics.cs.msu.ru

Сивоволенко Сергей
OctoNus Software Ltd.
E-mail:
sivovolenko@octonus.com

Аннотация

В данной работе[1] описывается метод применения технологии HDRI (high dynamic range imaging) для интерактивного улучшения изображения, полученного при помощи оптического микроскопа. Современные цифровые камеры и дисплеи могут записать и отобразить очень ограниченный динамический диапазон яркости, значительно меньший, чем воспринимаемый человеческим глазом. В связи с этим, выведенное на монитор изображение выглядит значительно менее контрастным, чем наблюдаемое непосредственно через окуляры микроскопа. Контрастность можно повысить за счет использования технологии HDRI. Однако для ее применения необходимо получение набора снимков с разной экспозицией, что требует значительного времени и не может быть сделано интерактивно. Нами предлагается подход для реализации интерактивного улучшения изображения. В наблюдаемой сцене постоянно фиксируются изменения. Если сцена не меняется в течение достаточного времени, осуществляется улучшение изображения с помощью технологии HDRI, иначе на монитор выводится изображение в исходном виде. Предложенный подход был реализован в виде программной системы и были проведены эксперименты, подтверждающие высокую степень его интерактивности.

Ключевые слова: *интерактивное улучшение изображений, детектор движений, HDRI, tone-mapping, цифровой микроскоп.*

1 ВВЕДЕНИЕ

В работе рассматривается система цифрового микроскопа, в ней полученное с помощью оптического микроскопа изображение выводится на монитор оператора. Такие системы находят широкое применение в различных областях промышленности и науки, таких как медицина, биология и контроль качества на производстве. Однако цифровые микроскопы имеют один серьезный недостаток. Современные устройства ввода (такие как цифровые камеры) и вывода (такие как LCD-дисплеи) могут записать и отобразить лишь небольшой динамический диапазон яркости (dynamic range of luminance). Для современных мониторов он составляет порядка $100 : 1$. Существуют разночтения в определении динамического диапазона для цифровых камер (см. [1, p. 16]), однако можно указать его значение приблизительно равным $400 : 1$. В то же время, динамический диапазон воспринимаемой человеческим глазом яркости составляет порядка $10^9 : 1$. Если не принимать в расчет возможность аккомодации глаза, то динамический диапазон воспринимаемой в рамках одной сцены яркости составляет порядка $10000 : 1$ (данные взяты из [2, p. 191]). Таким образом, при наблюдении объекта в окуляры микроскопа доступен динамический диапазон яркости на два порядка больший, нежели при наблюдении посредством монитора.

Рис. 1: Изображение, полученное напрямую с камеры (сверху), и подготовленное для вывода HDR-изображение (снизу)

Проблема ограниченности динамического диапазона яркости устройств ввода и вывода не нова, и существует подход к ее решению. Он включает в себя два этапа.

[1]Изображения из статьи в цвете и полном разрешении доступны по адресу http://graphics.cs.msu.ru/ru/science/research/3dreconstruction/dmc

Первым делом необходимо получить изображение сцены с расширенным динамическим диапазоном яркости, ее HDR-изображение (high dynamic range). Для этого выполняются несколько снимков с разной экспозицией. Подробнее о HDR-изображениях см. [2, ch. 3]. Затем необходимо подготовить HDR-изображение для вывода на монитор — устройство с низким динамическим диапазоном яркости, на LDR-устройство (low dynamic range). Процедура подготовки HDR-изображения к выводу на LDR-устройство заключается в интеллектуальном сжатии динамического диапазона с сохранением контрастности деталей на изображении. Эта процедура носит название tone-mapping, подробнее о ней см. [2, ch. 6]. В результате получается приятное на глаз изображение с хорошей контрастностью. На рисунке 1 сверху приведен пример изображения сцены, полученного напрямую с камеры, а на рисунке 1 снизу показан пример изображения той же сцены, полученного в результате применения описанного выше подхода.

Построение HDR-изображения и подготовка его для вывода на LDR-устройство требует существенных вычислительных затрат. Кроме того, для получения HDR-изображения необходимо выполнить съемку кадров с длительными экспозициями. Таким образом, невозможно выполнять улучшение изображения в интерактивном режиме. В данной работе предлагается подход, позволяющий осуществлять улучшение изображения с микроскопа при помощи технологии HDRI с сохранением интерактивности за счет использования детектора движения.

В разделе 2 подробно описан предлагаемый метод — как идея организации интерактивности, так и технические детали ее реализации. Результаты тестирования и итоговые показатели скорости работы указаны в разделе 3. В разделе 4 приведено заключение по поводу проделанной работы, а также указаны усовершенствования, которые могут быть добавлены в будущем.

2 ПРЕДЛАГАЕМЫЙ МЕТОД

Основной идеей предлагаемого метода является его интерактивность, т. е. быстрая и автоматическая реакция на изменения в сцене. Для вычисления улучшенного изображения требуется значительное время, нам же необходимо в каждый момент времени выводить на монитор оператора актуальную картинку.

Для этой цели постоянно отслеживаются изменения в наблюдаемой сцене. При наличии в ней изменений на монитор выводится изображение без улучшений. Если же сцена остается статичной в течение достаточного времени, на монитор выводится улучшенное изображение. Таким образом изображение на мониторе всегда поддерживается в актуальном состоянии с минимальным временем отклика на изменения в сцене.

2.1 Стратегия вывода изображения

В каждый момент времени программа находится в одном из состояний “motion”, “synthesis” или “display”. В процессе работы мы постоянно анализируем полученные с камеры изображения, на основании чего делаем вывод, изменяется ли сцена, и переходим в то или иное состояние. Изменения могут быть вызваны как движением объекта (если оператор передвинул образец или сместил окуляр микроскопа), так и изменением степени увеличения или положения фокусировки микроскопа, или же изменением освещения сцены.

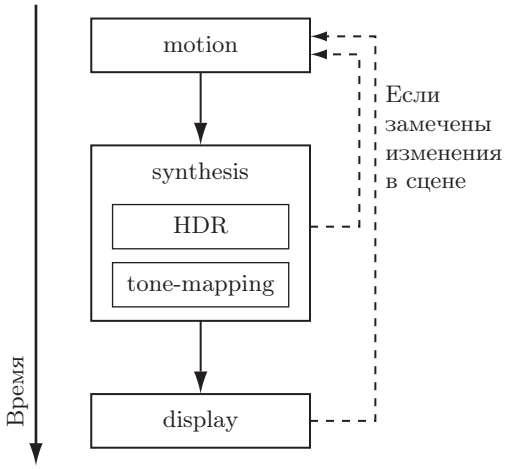


Рис. 2: Схема перехода между состояниями программы

На рисунке 2 представлена схема перехода между состояниями программы. Если сцена неизменна с течением времени, переход осуществляется по стрелкам, направленным вниз. Если в сцене фиксируются изменения, переход происходит по пунктирным стрелкам, направленным вверх.

Пока мы фиксируем изменения в сцене, программа находится в состоянии “motion”, и на экран монитора выводится изображение, полученное напрямую с камеры (далее будем называть его LDR-изображением). Как только сцена перестает меняться, программа переходит в состояние “synthesis”. В нем сначала строится HDR-изображение (на схеме это блок с надписью “HDR”), а затем оно подготавливается к выводу на монитор (блок с надписью “tone-mapping”). Для построения HDR-изображения используется несколько изображений с камеры, выполненных с различной экспозицией.

В процессе синтеза HDR-изображения мы также анализируем сцену на наличие изменений, в течение этого времени на мониторе отображается последнее полученное с камеры LDR-изображение. Как только в сцене были замечены изменения, мы избавляемся от полученных изображений как от неактуальных и возвращаемся в состояние “motion”.

Если же в процессе синтеза HDR-изображения не было зафиксировано изменений в сцене, выполняем tone-mapping и получаем подготовленное для вывода TM-изображение (предполагается, что tone-mapping выполняется достаточно быстро). После этого переходим в состояние “display”, в котором на монитор выводится TM-изображение. В процессе вывода мы в фоновом режиме продолжаем анализировать сцену на наличие изменений. Если таковые были замечены, избавляемся от полученного TM-изображения как от неактуального и возвращаемся в состояние “motion”.

Чтобы сгладить эффект перепадов яркости выводимого изображения при переходе в состояние “display” и из него в “motion”, мы использовали небольшую задержку, в течении которой изображение плавно меняется из LDR в TM и наоборот с помощью линейного усреднения. Этот простой прием позволяет сделать перепады яркости между TM и LDR изображениями приятными на глаз.

2.2 Детектор движения

Алгоритм должен уметь фиксировать движения в двух различных ситуациях. Во-первых, для состояний "motion" и "display", ему необходимо сравнивать изображения, сделанные с одной экспозицией. Во-вторых, для состояния "synthesis" в процессе построения HDR-изображения, ему нужно сравнивать изображения, выполненные с различными экспозициями. Ниже рассмотрены оба случая.

2.2.1 Случай одинаковых экспозиций

В этом случае мы выполняем простое попиксельное сравнение изображений с некоторой заданной заранее точностью.

Будем считать, что изображение I_1 отличается от изображения I_2 в пикселе (x, y), если значения пикселей отличаются больше, чем на ε. Будем считать, что на изображении I_2 зафиксированы изменения относительно изображения I_1, если отношение количества пикселей, в которых они отличаются, к общему количеству пикселей больше, чем δ:

$$\frac{\left|\{(x,y) \in P : |I_1(x,y) - I_2(x,y)| > \varepsilon\}\right|}{\left|P\right|} > \delta,$$

где P есть множество пикселей изображений.

Величины ε и δ можно варьировать для изменения уровня чувствительности. Необходимый уровень чувствительности зависит от таких внешних факторов, как интенсивность шума камеры и естественные колебания освещенности сцены.

2.2.2 Случай различных экспозиций

В этом случае для сравнения мы выполняем несколько дополнительных шагов. Сначала изображения I_1 и I_2 приводятся к одному масштабу яркости, для чего значения пикселей просто делятся на величину экспозиции. Получаем новые изображения R_1 и R_2, которые можно считать картами излучения сцены (radiance map):

$$R_1(x,y) = \frac{I_1(x,y)}{EV_1}, \quad R_2(x,y) = \frac{I_2(x,y)}{EV_2}.$$

Затем мы отсеиваем пиксели, недо- или переэкспонированные хотя бы на одном из изображений. Будем считать что, пиксель (x, y) изображения I правильно отэкспонирован, если

$$\varepsilon_1 < I(x,y) < 1 - \varepsilon_2.$$

Таким образом, пиксель (x, y) будет участвовать в сравнении, если

$$\varepsilon_1 < I_1(x,y) < 1 - \varepsilon_2, \quad \varepsilon_1 < I_2(x,y) < 1 - \varepsilon_2.$$

После этого сравниваются удовлетворяющие этим неравенствам значения $R_1(x, y)$ и $R_2(x, y)$. Однако не совсем корректно рассматривать их разность, т. к. эти величины могут быть совершенно разных порядков в зависимости от значений экспозиций EV_1 и EV_2. Поэтому мы вычисляем их отношение. Итак, будем считать, что изображение I_1 отличается от изображения I_2 в пикселе (x, y) если

$$\frac{1}{1+\varepsilon} < \frac{R_1(x,y)}{R_2(x,y)} < 1 + \varepsilon.$$

Наконец, как и ранее, будем считать, что на изображении I_2 зафиксированы изменения относительно изображения I_1, если отношение количества пикселей, в которых они отличаются, к общему количеству пикселей больше некоторой наперед заданной величины δ.

Значения величин ε и δ вполне могут отличаться от значений соответствующих величин в предыдущем пункте.

2.3 Синтез HDR-изображения и tone-mapping

Существует множество методов для синтеза HDR-изображения по набору снимков с различной экспозицией (см. [3] и [4]). Большая часть из них посвящена решению возникающих при этом проблем, неактуальных для нашей задачи. Поэтому мы использовали простейший метод, описанный, например, в [2, ch. 4].

Существует также множество различных алгоритмов tone-mapping'a (см. [5] и [6]). Мы использовали подходящий для нашей задачи с точки зрения соотношения скорости и качества работы алгоритм, описанный в [7].

3 ЭКСПЕРИМЕНТЫ

Предлагаемый метод был реализован и протестирован на реальной системе. В системе использовалась камера Prosilica GC1380C, установленная на микроскоп Leica MZ6. Метод бы реализован на языке C++ в среде Microsoft Visual Studio 2005 с использованием высокопроизводительной библиотеки для обработки изображений Intel IPP 6.0.

В процессе тестирования измерялось время работы конкретных участков программы. Измерения были усреднены на продолжительном (в несколько минут) отрезке времени. В течение этого времени программа совершала серию переходов между состояниями, для чего сцене периодически обеспечивались изменения. Например, за счет передвижения наблюдаемого объекта.

В таблице 1 приводятся показатели скорости работы метода. В первой колонке таблицы указана измеренная величина, во второй — состояние программы, для которого производилось измерение, в третьей — полученное значение. Для измерений был использован компьютер с процессором Intel Pentium 4 @ 2.8GHz, размер изображений составлял 1360 × 1024 пикселей (максимально возможный для данной модели), изображения цветные, глубина цвета 24 бита на пиксель. Для формирования HDR-изображения использовалось пять снимков с экспозициями в 8, 16, 32, 64 и 128 миллисекунд. Экспозиция снимков в режимах "motion" и "display" составляла 32 миллисекунды.

Величина	Состояние	Значение
Скорость обновления	motion	13.2 fps
Задержка на улучшение	synthesis	2.6 s
Время реакции на изменение сцены	synthesis	450 ms
	display	75 ms
Скорость обновления для неинтерактивной версии		0.38 fps

Таблица 1: Показатели скорости работы метода

В первой строке указана скорость обновления изображения на мониторе (в кадрах в секунду) для изменяющейся сцены, т. е. для состояния "motion". Во второй указано время (в секундах), требуемое для получения HDR-изображения и формирования TM-изображения (для состоянии "synthesis"). В третьей и четвертой строках приводится время (в миллисекундах), требуемое программе для обнаружения изменений в сцене, значения даны для состояний "synthesis" и "display" соответственно.

Наконец, в пятой строке приводится скорость обновления изображения на мониторе (в кадрах в секунду) для аналогичной неинтерактивной системы. Такая система может быть получена, если фиксировать программу в состоянии "synthesis" и не использовать детектор движений. В этом случае с максимально возможной скоростью будут строиться HDR-изображения и выводиться на монитор TM-изображения.

4 ЗАКЛЮЧЕНИЕ

В работе описан метод для улучшения изображения, полученного с оптического микроскопа, с помощью технологии HDRI в интерактивном режиме. Представлена идея организации интерактивности и подробно описан алгоритм ее осуществления. Метод реализован и протестирован на реальной системе, по результатам тестирования показана его высокая производительность и возможность успешного применения в рамках поставленной задачи.

В будущем планируется внедрение в алгоритм автоматического выбора значений $\{EV_n\}$, а также значений экспозиции, которая используется в состояниях "motion" и "display". Кроме того, возможно, будет добавлено еще одно состояние "tonemapping" между "synthesis" и "display", на котором будет вычисляться более аккуратное TM-изображение.

БЛАГОДАРНОСТИ

Работа выполнена в сотрудничестве и при финансовой поддержке OctoNus Software Ltd.

Список литературы

[1] Karol Myszkowski, Rafal Mantiuk, and Grzegorz Krawczyk, *High Dynamic Range Video*, Synthesis Digital Library of Engineering and Computer Science. Morgan & Claypool Publishers, San Rafael, USA, 2008.

[2] Erik Reinhard, Greg Ward, Sumanta Pattanaik, and Paul Debevec, *High Dynamic Range Imaging: Acquisition, Display and Image-Based Lighting*, Morgan Kaufmann Publishers, Dec. 2005.

[3] T. Mitsunaga and S. K. Nayar, "Radiometric self calibration," in *Proceedings of the IEEE Computer Science Conference on Computer Vision and Pattern Recognition (CVPR-99)*, Los Alamitos, June 23–25 1999, pp. 374–380, IEEE.

[4] Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski, "High dynamic range video," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 319–325, July 2003.

[5] Greg Ward, "A contrast-based scalefactor for luminance display," in *Graphics Gems IV*, Paul Heckbert, Ed., pp. 415–421. Academic Press, Boston, 1994.

[6] Sumanta N. Pattanaik, James A. Ferwerda, Mark D. Fairchild, and Donald P. Greenberg, "A multiscale model of adaptation and spatial vision for realistic image display," in *SIGGRAPH 98 Conference Proceedings*, Michael Cohen, Ed. ACM SIGGRAPH, July 1998, Annual Conference Series, pp. 287–298, Addison Wesley.

[7] Michael Ashikhmin, "A tone mapping algorithm for high contrast images," in *Rendering Techniques 2002 (Proceedings of the Thirteenth Eurographics Workshop on Rendering)*, June 2002.

Interactive enhancement of microscopy images using HDRI technology

ABSTRACT

In this paper we study a problem of interactive enhancement of microscopic images using the HDRI technology. Dynamic range of conventional digital cameras and monitors is much smaller than the human visual system can perceive. Hence an image on the screen appears much less contrast than an image that is seen directly through the eyepieces of the microscope. Contrast of an image can be enhanced by means of the HDRI technology. However to apply the HDRI multiple images of the scene taken with deferent exposures are required. For this reason HDRI technology cannot be directly applied in frames of an interactive imaging system. In this paper we present a framework that enhances microscopic images with an HDRI technology, and remains interactive. Proposed method permanently analyzes an observed scene in order to detect motion. If the scene remained static long enough our system enhances an image with HDRI, otherwise it interactively displays live image from the camera. Proposed framework has been implemented and tested on a real microscope. Performance measurements, presented in the paper demonstrate that our system can improve microscopic images with the HDRI and remaining interactive in the same time.

Keywords: *interactive image enhancement, motion detector, HDRI, tone-mapping, digital microscope*

ABOUT THE AUTHORS

Mikhail Matrosov is a student at Computational Mathematics and Cybernetics department of Moscow State University. His research interests include interactive image enhancement and adjacent fields. His contact e-mail is fsgs2k@gmail.com.

Victor Gaganov is a PhD student at Keldysh Institute of Applied Mathematics of Russian Academy of Science. His research interests include computer vision, 3D reconstruction and adjacent fields. His contact e-mail is vgaganov@graphics.cs.msu.ru.

Alexey Ignatenko is a researcher at Computational Mathematics and Cybernetics department of Moscow State University. His research interests include photorealistic 3D rendering, 3D modelling and reconstruction, image-based rendering and adjacent fields. His contact e-mail is ignatenko@graphics.cs.msu.ru.

Sergey Sivovolenko is a CEO of OctoNus Software Ltd. His contact e-mail is sivovolenko@octonus.com.

Автоматическое определение интенсивности деления клеток по снимкам с микроскопа

Д. А. Лаптев,
Факультет вычислительной математики и кибернетики МГУ
laptev.d.a@gmail.com

Аннотация

В статье рассматривается задача распознавания делящихся клеток на фотографии с микроскопа. Ее целью является определение числа клеток, которые находятся в стадии митоза (деления). Актуальность задачи вызвана необходимостью исследования благотворного влияния различных сред на интенсивность процесса деления. В процессе решения задачи были рассмотрены несколько распространенных методов, некоторые эвристики, были модифицированы алгоритмы свертки, бинаризации и анализа полученного бинарного изображения.

Ключевые слова: Свертка, бинаризация, учет информации о цвете, симметричность, выпуклая оболочка, разбиение связных областей.

1. ВВЕДЕНИЕ

В настоящее время задача выделения объектов на изображении является очень распространенной в самых разных областях. Проблема выделения клеток на изображении не является новой, существует большое число публикаций, посвященных выделению отдельных клеток, подсчету их числа, определению размеров, например, работа [1]. Все эти задачи успешно решаются с использованием методов свертки, бинаризации и геометрических преобразований.

Задача, которая рассматривается в данной статье, отличается от них тем, что необходимо учитывать и корректно распознавать очень маленькие объекты — почки. На ее примере рассмотрены несколько методов: пример работы с заданными формами объектов и учетом априорной информации (цвета). Одним из распространенных методов выделения объекта на фоне является алгоритм разреза графов [2], который учитывает априорную информацию. В данной статье рассмотрен иной подход, который учитывает цветовую информацию лишь косвенно.

Во второй и третьей частях статьи рассмотрены постановка задачи и ее особенности. Четвертая часть посвящена описанию алгоритма бинаризации изображения. В пятой части речь идет о выделении связных областей и о сглаживании координат. В шестой части рассмотрены два алгоритма определения клеток с почками, основанные на анализе формы, а также рассмотрен вопрос обработки нескольких „слипшихся“ клеток.

2. ПОСТАНОВКА ЗАДАЧИ

Задача распознавания клеток в состоянии митоза на изображении была поставлена для фотографий с микроскопа, сделанных в спектральном диапазоне. Основной целью является определение процента делящихся клеток (именно это число характеризует среду, благотворную для деления).

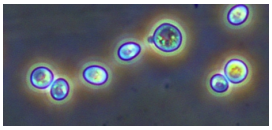


Рисунок 1: Исходное изображение

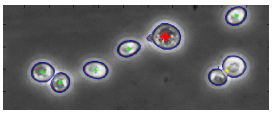


Рисунок 2: Распознанные области

На фотографиях клетки без почек в основном имеют форму, близкую к эллиптической и любой цвет от белого до темно-синего. Единственной цветовой характеристикой клетки является синий ореол вокруг нее, хотя наблюдаются частые исключения из этого правила. Клеткой в стадии митоза можно считать объект, похожий по форме на ассиметричную восьмерку, где одна часть (почка) меньше другой (рис. 2, клетка, помеченная красной звездочкой).

3. СПЕЦИФИКА ЗАДАЧИ

1. Размер данного изображения достаточно большой (2560 на 1920 пикселей), поэтому при выборе решения необходимо ориентироваться на быстродействие методов. При этом разрешение не может быть сильно уменьшено, так как уменьшение разрешения сделает делящиеся клетки с небольшими почками неотличимыми от обычных клеток.

2. Клетки на фотографии, в среднем, являются маленькими объектами, на некоторых изображениях с микроскопа их число достигает 200 и более. При этом почки являются еще более мелкими объектами, поэтому их распознавание усложнено слабой детализацией изображения, несмотря на высокое разрешение фотографий.

3. Цветовая гамма и освещение могут меняться в зависимости от среды, в которой были сделаны фотографии.

4. На изображениях число клеток может различаться в несколько раз, что не позволяет использовать априорные знания об их количестве.

4. СВЕРТКА И БИНАРИЗАЦИЯ ИЗОБРАЖЕНИЯ

Одним из наиболее удобных способов представления изображения является бинарное изображение, которое и было использовано в задаче (см. также [3]). Чтобы соста-

Рисунок 3

Рисунок 4

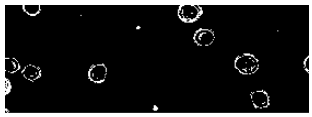


Рисунок 5

Рисунок 6

вить бинарное изображение, необходимо разделить пиксели изображения на два класса: объект и фон. Одним из распространенных методов такого разделения является алгоритм свертки [4]. Идея заключается в том, чтобы в каждом пикселе посчитать модуль разности интенсивности цвета в этой точке и соседних с ней, а затем, подставив эти значения в неубывающую функцию, отсечь вычисленные значения по некоторому порогу: те пиксели, для которых значения окажутся больше порога — пометить объектом, остальные — фоном. В нашей задаче была выбрана следующая формула:

$$B_{i,j} = \begin{cases} 0, & \text{если } \max\limits_{i_1,j_1 \in I} |A_{i,j} - A_{i_1,j_1}| < C_{edge}; \\ 1, & \text{иначе,} \end{cases}$$

A — исходное изображение, B — бинарное изображение, I — область связности, C_{edge} — пороговое значение.

Чтобы избежать рваных контуров было принято решение выбрать в качестве множества I 24-связную область (двойное кольцо вокруг пикселя). При выборе такого параметра выделенные границы могут получиться минимум три пикселя шириной и вероятность получить разрывный контур существенно снижается при адекватном выборе параметра C_{edge}.

Существуют методы автоматического определения порога для изображения в серых тонах на основе анализа гистограммы (см. работу [5]). К сожалению, в данной задаче этот метод применить нельзя: мы работаем не с изображением в явном виде, а с градиентом изображения, и если рассмотреть гистограмму значений градиента в точках по всему изображению, график получается монотонным и гладким. Таким образом, нужно найти другие методы определения порога.

Для начала, перейдем к порогу, равному отношению числа пикселей объекта к общему числу пикселей (обозначим это значение $C_{percent}$). Обозначим гистограмму значений градиента через $f(x)$, тогда получим следующее взаимно однозначное соотношение:

$$C_{edge} : \int_0^{C_{edge}} f(x)dx = C_{percent}.$$

Введенное таким образом, значение C_{edge} легко вычисляется и позволяет отойти от оперирования абстрактными значениями и перейти к универсальному представлению порога через процент пикселей объекта. Зная значение $C_{percent}$, можно получить бинарное изображение — назовем это функцией вычисления границ по порогу.

Для корректного определения порога была использована информация о цвете границ клеток (они в большинстве случаев синие). Можно построить бинарное изображение по этому признаку: синие пиксели — объект, остальные — фон. Назовем последовательность этих действий функцией определения границ по цвету. Помимо рваных границ, проблема здесь возникает в том, что фоновые пиксели, отстающие на некоторое расстояние от клетки, также часто удовлетворяют этому условию, хотя они не принадлежат клеткам и их не надо учитывать (см. рисунок 4). Эта проблема решается вычитанием фона изображения, при этом интенсивность фоновых пикселей становится близка к нулю, а границы клеток остаются практически неизменными. Однако при реализации алгоритма был выбран другой вариант, который выигрывает в быстродействии:

1. Выбрать значение $C_{percent}$ заведомо большим, чем процент пикселей-объектов на изображении.
2. Применить функцию определения границ по порогу (границы очень размазаны и много постороннего шума) — рисунок 3
3. Применить функцию определения границ по цвету (границы определяются немного рваными, пиксели фона частично относятся к объекту) — рисунок 4
4. Пересечь изображения, полученные двумя функциями, получив при этом первое приближение (фон почти отделяется, границы тонкие) — рисунок 5
5. Вычислить $C_{percent}$ как отношение числа пикселей объекта к общему числу пикселей
6. Снова запустить функцию определения границ по порогу — рисунок 6

Полученное таким образом бинарное изображение практически не имеет шума на пикселях фона, а также имеет достаточно толстые границы, которые в большинстве случаев являются замкнутыми.

Кроме того, в процессе предобработки применяется алгоритм борьбы с мелким шумом, медианная фильтрация [6], которая позволяет избавиться от отдельных небольших выбросов.

5. ВЫДЕЛЕНИЕ СВЯЗНЫХ ОБЛАСТЕЙ И СГЛАЖИВАНИЕ

После того, как изображение бинаризовано, необходимо перейти к какому-либо признаковому пространству. В на-

шем случае признаковое пространство выбрано как координаты точек, составляющих границу связной области. Для выделения связных областей можно использовать метод, описанный, например, в работе [7]. Центр масс для удобства лучше смещать в начало координат и после этого переводить все координаты в полярные.

В силу специфики задачи, рассматриваются только те области, которые имеют нулевой уровень вложенности (области, вложенные друг в друга рассматриваются как шум — чаще всего это внутренние структуры клетки).

На практике полученные координаты непригодны для распознавания: изображение является массивом точек, поэтому координаты получаются дискретными и график сильно изломан. Бороться с этим можно, например, применив сглаживание или фильтрацию [8] с переменной шириной окна: $WindowSize = \max\left(\frac{1}{4\pi}sl, \frac{l}{35}\right)$, где s — минимальное отношение радиуса почки к радиусу клетки, l — число пикселей, составляющих границу, длина массива x. Численные параметры подобраны исходя из результатов работы алгоритма.

6. АЛГОРИТМ ВЫДЕЛЕНИЯ КЛЕТОК И ПОЧЕК

Для определения на изображении клеток и почек необходимо в первую очередь проверить, является ли выделенный в связную область объект клеткой (возможно с почкой), шумом, или группой клеток, выделенных в одну связную область. Для этого проверяются следующие факты:

1. Лежит ли центр масс связной области за пределами выделенной области, или внутри. Если центр масс лежит за пределами выделенной связной области, связная область помечается шумом.
2. Превосходит ли размер связной области максимальный размер клетки. Если размер связной области больше размера максимальной клетки — это скорее всего несколько клеток, объединенных в одну связную область.
3. Минимальный размер клетки превосходит размер связной области. Если он окажется больше размера минимальной клетки — это шум.
4. Наконец для всех оставшихся областей, проверяется наличие у них элемента, напоминающего клетку и почку. Остановимся на этом подробнее ниже.

Для определения клетки и почки у нее, после проверки нескольких методов на эффективность было отобрано два метода, которые показали лучшие результаты.

Первый метод заключается в том, чтобы построить выпуклую оболочку (вычислительно эффективный алгоритм приведен в книге [9]), а далее анализировать отклонения от нее. Если граница связной области является приближенной к выпуклой оболочке, скорее всего это отдельная клетка и этот случай можно легко выявить. Параметром в данном случае является максимальное допустимое значение невязки. Далее, некоторая последовательность точек, в которых отклонение от выпуклой оболочки ненулевое (точки не входят в выпуклую оболочку), анализируется следующим образом: если кривизна на этом участке больше некоторой величины, зависящей от длины границы связной области, а длина промежутка больше радиуса минимальной клетки (и меньше радиуса большей клетки), то область помечается как клетка с почкой:

$$\left|\Sigma_{k=i+1}^{j}\left(\rho_k - \rho_i + \frac{k-i}{j-i}(\rho_i - \rho_j)\right)\right| > Cl,$$

$$j - i \geq R_{min}, j - i \leq R_{max},$$

$$i, j : i \in L, j \in L, \forall k(i < k < j) : k \notin L,$$

$$L \text{ — выпуклая оболочка}, l = \text{len}(\rho), C = \text{const}.$$

Тем не менее, метод не всегда подходит из-за того, что не всегда корректно определяются границы клетки, а также из-за дискретности координат границы.

Чтобы компенсировать некоторые недостатки первого метода, используется второй метод, идея которого очень проста, но действенна. Выделенные границы проверяются на симметричность: фактически, проверяется несколько точек границы (в реализации алгоритма 32), имеющих угловую координату ϕ в промежутке от 0 до π, для каждой из них находится точка с угловой координатой $2\pi - \phi$. Для них оценивается разница в координатах ρ, если она больше, чем размер минимальной клетки, умноженной на некоторую константу, и кроме того для двух соседних точек (уже с меньшим шагом, например, 64) разница также больше, область помечается как клетка с почкой:

$$\exists i = 0, ...31 : \rho_{\text{ind}(\frac{\pi * i}{32})} \geq cl,$$

$$\rho_{\text{ind}(\frac{\pi * i - 1}{64})} \geq cl,$$

$$\rho_{\text{ind}(\frac{\pi * i + 1}{64})} \geq cl,$$

$$l = \text{len}(\rho), c = \text{const},$$

$$\text{ind}(\phi) = \underset{k}{\text{argmin}}|\phi_k - \phi|.$$

Особенность метода в том, что нередко он определяет две спаренные клетки как одну, или как клетку с почкой, так как очертания спаренных клеток могут быть достаточно симметричны. Устранить этот недостаток достаточно просто: область не является клеткой с почкой, если $\frac{\max(\rho)}{\min(\rho)} > C = \text{const}$.

После всех проделанных этапов, нераспознанными остаются только большие связные области. Одним из вариантов их обработки является выделение их с меньшим порогом на этапе выделения границ. В этом случае алгоритм выделения границ оставит границы более тонкими и некоторые спаренные клетки, возможно, будут выделены алгоритмом в разные области. При этом нет необходимости пересчитывать все изображение, только его части, которые содержат нераспознанные области.

Другой способ состоит в последовательном приближении области двумя эллипсами, тремя, и так далее, пока их число не превысит максимального заданного (или зависящего от площади) значения, или пока они не приблизят область с достаточной точностью. Проводить эту процедуру можно, например, с помощью EM-алгоритма [10]. Можно свести задачу распознавания отдельных клеток на одной связной области к задаче разделения смеси двумерных нормальных распределений: сгенерировать точки, несколько удаленные от границы, выделенной алгоритмом, а затем по ним восстановить плотность, не учитывая при этом граничные точки.

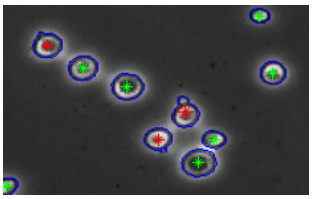


Рисунок 7

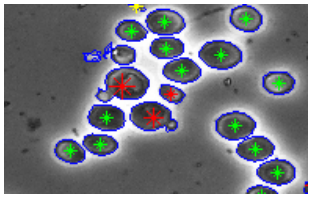


Рисунок 8

7. ВЫВОДЫ

В статье рассмотрена задача распознавания клеток и клеток в состоянии митоза на фотографии. Данная задача решена с помощью последовательного применения методов, каждый из которых был модифицирован с учетом ее специфики: алгоритм свертки, бинаризация с учетом цветовой компоненты, выделение связных областей, фильтрация и отсеивание шума. Для анализа наличия почек проведены проверки на симметричность и отклонения от выпуклой оболочки. Кроме того, рассмотрены пути обработки сгустков клеток, отнесенных в одну связную область. Итоговый алгоритм допускает на изображениях с общим числом клеток около ста примерно однудве ошибки первого и второго рода — точность, сравнимая с ручным подсчетом.

Возможные дальнейшие пути работы над этой задачей:

1. Учет априорной информаци о форме объектов для улучшения алгоритма выделения границ [11], а также учет информации о цвете, например, модификация алгоритмов разреза графов, на примере, изложенном в работе [12].
2. С другой стороны, расширение класса фотографий, с которыми может работать алгоритм (например, анализ фотографий, снятых в других цветовых режимах).
3. Реализация упомянутых выше способов обработки сгустков клеток из одной связной области.

8. БЛАГОДАРНОСТИ

Автор выражает благодарность И. В. Володяеву за предоставленные материалы, Д. П. Ветрову и Д. А. Кропотову за предоставленные идеи и поддержку в решении задачи. Работа выполнена при частичной поддержке гранта РФФИ 08-01-00405.

9. СПИСОК ЛИТЕРАТУРЫ

[1] Yingying Deng Qingmin Liao, "An accurate segmentation method for white blood cell images," Biomedical Imaging, pp. 245– 248, 2002.

[2] Vladimir Kolmogorov Yuri Boykov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," EMMCVPR, pp. 359–374, 2001.

[3] Pietik*ä*inen M. M*ä*enp*ää* T. Ojala, T., "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," IEEE Trans. Pattern Anal. Mach. Intell., 2002.

[4] D. Ziou and S. Tabbone, "Edge detection techniques an overview," International Journal of Pattern Recognition and Image Analysis, , no. 8(4), pp. 537–559, 1998.

[5] Tony Lindeberg, "Edge detection and ridge detection with automatic scale selection," Technical report ISRN KTH NA/P–96/06–SE, 1996.

[6] Ben Weiss, "Fast median and bilateral filtering," Technical report Shell & Slate Software Corp., 2006.

[7] Игнатенко А. В. Баяковский Ю. М., "Машинная графика. Курс лекций.," http://graphics.cs.msu.ru/courses/cg, 2008.

[8] James V. Herod Evans M. Harrell II, "Linear methods of applied mathematics," http://mathphysics.com/pde/, 1994.

[9] Р.Ривест К.Штайн Т. Кормен, Ч. Лейзерсон, "Алгоритмы. Построение и анализ.," 1999.

[10] Christopher M. Bishop, "Pattern recognition and machine learning," 2006.

[11] Florian Tischh*ä*user Daniel Cremers, "Diffusion snakes: Introducing statistical shape knowledge into the mumford-shah functional," International Journal of Computer Vision, , no. 50(3), pp. 295–313, 2002.

[12] Yuri Boykov Vladimir Kolmogorov, "From photohulls to photoflux optimization," ICCV, 2005.

ОБ АВТОРАХ

Лаптев Дмитрий — студент 3-го курса кафедры математических методов прогнозирования ВМК МГУ им. М.В.Ломоносова. E-mail: laptev.d.a@gmail.com

ABSTRACT

In the article the problem of finding dividing cells on the photos is observed. The point is to find out the number of cells in the state of mithosis. The reason is that it is necessary to explore what environment is better for the cell's rate of division. While solving this task few methods were observed, some heuristic algorithms were developed, convolution and binarisation algorithms were modified. To analyse the binary image different approaches were used, the combination of them made up the resulting algorithm.

A Novel Approach to Video Matting using Optical Flow

Mikhail Sindeyev[1], Vadim Konushin[1,2]
[1] Graphics and Media Lab, Moscow State Lomonosov University, Moscow, Russia
[2] Keldysh Institute of Applied Mathematics, Russian Academy of Sciences
E-mail: {msindeev, vadim}@graphics.cs.msu.ru

Abstract

The image matting problem refers to foreground object extraction from an image. Similarly, video matting problem refers to extraction of a foreground object from each frame of a video-sequence producing a moving foreground layer. Layer extraction process should deal with the transparency caused by camera point spread function (PSF) and motion blur, thus the natural way is to store extracted layer as a pair of images: color and opacity. The latter is referred to as an alpha-channel.

In this paper we propose an algorithm that takes alpha for the first frame (or an arbitrary key-frame) provided by the user (e.g. by using some image matting algorithm) and tracks its motion in time through the video sequence.

Unlike the obvious approach that models the whole scene motion from one frame to the next with an optical flow (e.g. to warp the rough input segmentation (trimap) or the alpha channel with this flow), we use it to model the foreground layer motion only. This prevents us from otherwise unavoidable artifacts on the boundaries of foreground objects, which are of the main interest in the matting problem.

Instead of matching pixel colors in the pair of frames, we measure how the optical-flow-warped alpha-channel fits the next frame (we use color matching as an additional regularization though). By minimizing this measure with respect to optical flow we fully preserve the structure of the foreground object and prevent temporal incoherence artifacts.

Keywords: *video matting, optical flow, digital compositing, foreground extraction.*

1. INTRODUCTION

In this paper we address the problem of extraction of a moving foreground object from a natural video sequence. This is referred to as a video matting problem. Similarly to the image matting problem, in each frame a source image C is assumed to be a composite of two image layers F and B (foreground and background) with opacity channel α. In each pixel their RGB values should satisfy *the compositing equation*:

$$C = \alpha F + (1-\alpha)B, \quad (1)$$

where C, F and B are 3D vectors of RGB values, $0 \le \alpha \le 1$. The problem is to reconstruct the α, F and sometimes B images from the source image C using some additional user input. An example of such input is trimap – a rough segmentation of the image into foreground ($\alpha=1$), background ($\alpha=0$) and unknown region (where α is to be reconstructed by the algorithm).

The goal is usually to put the foreground layer onto a new background or process foreground and background layers separately using some image filter.

According to [11] video matting has the following challenges in comparison to image matting:

Figure 1. *Harbor* video sequence from [3]. **Top**: initial user-specified mask for frame #0. **Middle**: frame #30. **Bottom**: generated mask for frame #30 by forward-tracking the mask from frame #0 using the proposed algorithm.

- Large data size (width × height × duration vs. width × height)
- Temporal coherence: video frames may look well when observed independently, but produce notable motion artifacts
- Fast motion vs. low temporal resolution: it may be difficult to automatically identify motion between frames at typical frame rates (15-30 FPS for most cameras)

and has the following advantages:

- Ability to create more complete model of background by analyzing its motion over a range of frames
- Possibility to detect foreground/background edge when part of the background is covered up or revealed as a result of the background or object motion.

2. PREVIOUS WORK

Despite of the recent success in the image matting field (Closed-Form Solution [7], RobustMatting [12]), video matting is poorly developed. Only simple approaches exist. We overview them below.

Obvious extension of image matting to video matting is to process all frames independently using some image matting algorithm. Additional input information used by the algorithm (such as trimap or user strokes) however can be interpolated in some way (either of itself or by using motion from the video sequence). An

example of such method is [3], where a trimap is interpolated using optical flow, after which video frames are matted with Bayesian Matting algorithm independently.

Disadvantages of the independent frame matting are:

- Temporal incoherence
- Needs much user work to provide enough input for each frame

Another approach is to treat video volume as a 3-D image. This is typically used with binary segmentation methods which perform boundary matting as a post-processing. 3-D volume is usually oversegmented first and at this step the spatial and temporal dimensions may be treated asymmetrically. An example of such method is [8]. Another method [2] is fully automatic but works only on simple videos with distinct separation of foreground and background layers by their motion.

Disadvantage of this method is a lack of consistent motion model: parts of moving objects in one frame do not necessarily match to the same parts in the next frame – they tend to disappear and re-appear rapidly in the result mask when the motion is fast. The definition of 'fast' here assumes that these parts of object do not overlap (or have little overlap) in two consecutive frames when projected onto one frame along the temporal axis.

3. PROPOSED ALGORITHM

3.1 Workflow overview

We start from a known alpha-channel for the first frame. It can be produced by a user using any existing image matting algorithm. Then we try and deform it with a smooth optical flow, until we get the best match for the next frame. We process consequent frames in the same way.

In the proposed scheme input data during processing the i-th frame consists of just alpha of the (i-1)-th frame and i-th frame image.

Thus we need a cost function that measures how well the warped alpha-channel fits the current image. In general case we use the Laplacian proposed in [7].

In case of a known background we use a simpler functional of squared difference between left-hand and right-hand parts of the compositing equation (1).

3.2 Energy function

When processing the current frame, we use the following energy:

$$E(V,\alpha) = E_d(V,\alpha) + \lambda E_s(V), \tag{2}$$

where V is the optical flow, α is the alpha-channel of the previous frame (as a column-vector) and λ is a smoothness parameter. α is fixed.

An image I can be warped by an optical flow what we denote by $V(I)$. Each pixel of the optical flow map consists of two components: $V=(V_x, V_y)$. Warping is performed in the following way:

$$V(I)\big|_{(x,y)} = I(x + V_x(x,y), y + V_y(x,y)), \tag{3}$$

where I is an arbitrary image to be warped. For fractional coordinate values we use a bilinear interpolation. Later on we'll refer to pixel index just as i, not as (x,y), thus treating images as vectors.

In our workflow we obtain only V by solving

$$V = \arg\min_V E(V,\alpha) \tag{4}$$

Then an alpha-channel for the current frame is constructed by warping α with the found optical flow V:

$$\alpha_{next} = V(\alpha). \tag{5}$$

The constructed alpha is used to process the next frame.

3.2.1 Laplacian-based data term

The data term uses the Laplacian from the Closed-form Matting algorithm [7]:

$$E_{d1}(V,\alpha) = V(\alpha)^T L V(\alpha). \tag{6}$$

Additionally, we use color information too, weighted by the alpha values:

$$E_{d2}(V,\alpha) = \sum_i V(\alpha)_i \left\| C_i - V(C_{prev})_i \right\|^2, \tag{7}$$

where the sum is taken over all pixels and subscript I refers to the i-th component of vector (that is, the i-th pixel of the image). C and C_{prev} are color images of the current and previous frame, respectively. This data term allows to consider motion in foreground region which also affects boundary via smoothness. Finally,

$$E_d(V,\alpha) = E_{d1}(V,\alpha) + \mu E_{d2}(V,\alpha). \tag{8}$$

3.2.2 Data term for known background

In case of a known background (which can be filmed separately if it's static, or reconstructed by planar tracking, e.g. with Mokey software [6]) we use a simpler functional, that however involves also the foreground color image F. F and α are being warped with the same optical flow, so the compositing equation (1) becomes:

$$C = V(\alpha)V(F) + (1 - V(\alpha))B, \tag{9}$$

where F and α are from the previous frame, while B and C are from the current frame. Data term for the known background case becomes:

$$E_d(V,\alpha) = \left\| C - V(\alpha)V(F) + (1 - V(\alpha))B \right\|^2. \tag{10}$$

After processing each frame, F (to be used in the next frame) is reconstructed from B, C and α using the equation (1), or better by using Bayesian Matting [4] (i.e. F is not being tracked from the very first frame, as is α).

3.2.3 Smoothness term

To regularize the optical flow we use a simple first-order smoothness term:

$$E_s(V) = \sum_i \sum_{j \in w(i)} \left\| V_i - V_j \right\|^2, \tag{11}$$

where $w(i)$ is a 3x3 pixel neighborhood of the i-th pixel.

3.3 Energy minimization

In our implementation we use QPBO method of discrete optimization [9] . We limit the search space at each pixel by a fixed-size window and use alpha-expansion algorithm iteratively to obtain a labeling for each pixel, where the label is optical flow vector $V=(V_x,V_y)$.

3.4 Hierarchical processing

The search-space of the optical flow estimation problem can be reduced by starting at lower resolution and then upscaling the intermediate results while using small search-window at each scale (e.g. 3x3 pixels).

3.5 Refinement

The result of tracking can be refined by applying Bayesian Matting with smoothness [10] to prevent the accumulation of tracking error and interpolation artifacts. Unknown region to be processed is constructed from all pixels where alpha is not equal 0 or 1. The mean value for the alpha at each pixel is taken from the generated

alpha map (4). The variance depends on the distance between foreground and background color distributions, thus regions with strict color separation are adjusted according to the image information, while in the undetermined regions alpha channel structure is preserved.

4. RESULTS AND COMPARISON

We have tested our algorithm on several videos. Due to the lack of test datasets for video matting, we used example videos from video matting/segmentation papers and created our own keyframe masks. To test known-background functional, we reconstructed background for some of the video sequences using homography tracking (i.e. approximating the background with a plane in 3D-space).

For comparison we used RobustMatting implemented by its authors. It has a video matting feature, which interpolates the trimap using optical flow or color difference (selectable by the user)

The testing showed that the resulting matte quality varies heavily depending on the 'complexity' of the video. The best results were achieved on simple videos from [3]. An example is shown in figure 1. When testing on the videos taken from different papers, our algorithm in most cases gave better result than the existing approaches. Examples are shown in figures 2 and 3.

Some additional videos from various sources demonstrated poor results for all algorithms including the proposed one. The difference on such 'hard' videos is that our algorithm fails to track the mask and it drifts and distorts inadequately, while other approaches produce random semi-transparent regions that quickly cover the entire image during the course of processing.

We also note that adding a known (or reconstructed) background improves the video matting result.

Processing time of our unoptimized C++ implementation of the algorithm is ~3 minutes per frame (640x480) on a 1.8GHz processor.

5. CONCLUSION AND DISCUSSION

Our approach works most well on video segments in which foreground objects do not drastically change their shape. Appearance of new parts of objects usually causes our method to fail.

Some of the examples of motion types for which our algorithm demonstrates noticeable advantages over the earlier approaches are:

- Hair blowing in the wind over complex background: overall hair structure is preserved while earlier approaches tend to catch some foreign background elements that show through the gaps in the hair
- Out-of-image-plane rotation of people arms, heads and bodies: high-speed motion of texture usually occurs near the boundaries of rotating objects while the boundary contour itself doesn't change much and is modeled well enough with smooth optical flow in the alpha-channel

Typical conditions of applicability of a matting algorithm usually include some (local) color separation between foreground and background across the object boundary. In our global minimization approach we weaken this condition: we require either color separation or motion smoothness in different parts of foreground object.

Assume that the left part of some solid object has a distinct edge with good color separation while the right part does not. In our approach the right side contour will most likely be dragged by the left one due to smoothness of the optical flow. Graph cut and trimap interpolation based methods will in contrast produce noisy and chaotic matte on the right side of the object (because they match pixels/segments on both sides of the object independently).

Additionally, our algorithm allows tracking of non-opaque objects to some extent, if their color-blending model satisfies the compositing equation (1).

We cannot however account for topology changes of the object silhouette. Indeed such cases need some information from the user, though simpler cases could be handled semi-automatically by adding some heuristics, which we do not address in our work.

5.1 Future work

Being rather immature, our algorithm has numerous possibilities for improvement, namely:

- Better regularization of optical flow smoothness, e.g. accounting for affine warping
- Using a pair of key-frames (at the both ends of the video segment being processed) and replace the extrapolation problem with the interpolation one to prevent the error accumulation
- Using different approaches of measuring how well the warped alpha-channel fits the image
- Model both background and foreground layers with independent optical flows (the background flow however should somehow be global rather than interframe, because the background layer suffers from occlusions, which may be pretty hard to achieve)
- It may be possible to switch to continuous optimization, similarly to Horn-Schunck method [5], by taking the linear term of the Taylor series for the α (and also for αF for the known-background case)

6. REFERENCES

[1] Agarwala, A., Hertzmann, A., Salesin, D., Seitz, S. *Keyframe-Based Tracking for Rotoscoping and Animation*, Proc. of SIGGRAPH, Vol. 32, No. 3, pp. 584–591, 2004

[2] Apostoloff, N., Fitzgibbon, A., *Automatic video segmentation using spatiotemporal T-junctions*, Proc. of the British Machine Vision Conference, 2006.

[3] Chuang, Y.-Y., Agarwala, A., Curless, B., Salesin, D., Szeliski, R. *Video Matting of Complex Scenes*, ACM Transactions on Graphics, Vol. 21, No. 3, pp. 243–248, 2002

[4] Chuang, Y., Curless, B., Salesin, D. and Szeliski, R.. *A Bayesian Approach to Digital Matting*, Proc. of IEEE CVPR, pp. 264–271, 2001

[5] Horn, B., Schunck, B., *Determining optical flow*, Artificial Intelligence, Vol 17, pp. 185–203, 1981

[6] Imagineer Systems Mokey http://www.mokey.com/products/mokey/

[7] Levin, A., Lischinski, D., Weiss, Y. *A Closed Form Solution to Natural Image Matting*, Proc. of IEEE CVPR, pp. 61–68, 2006

[8] Li, Y., Sun, J., Shum, H.-Y. *Video Object Cut and Paste*. Proc. of SIGGRAPH, Vol. 24, No. 3, pp. 595–600, 2005

[9] Rother C., Kolmogorov V., Lempitsky V., Szummer M. *Optimizing Binary MRFs via Extended Roof Duality*, Proc. of IEEE CVPR, pp. 1–8, 2007

[10] Sindeyev M., Konushin, V. and Vezhnevets, V. *Improvements of Bayesian Matting*, Proc. of Graphicon, pp. 88–95, 2007

[11] Wang, J. and Cohen, M. *Image and Video Matting: A Survey*. Foundations and Trends in Computer Graphics and Vision, Vol. 3, No. 2, pp. 97–175, 2007

[12] Wang, J. and Cohen, M. *Optimized Color Sampling for Robust Matting*, Proc. of IEEE CVPR, pp. 1–8, 2007

About the authors

Mikhail Sindeyev is a 5th year student at Graphics and Media Laboratory of Moscow State Lomonosov University, Department of Computational Mathematics and Cybernetics. His research interests include image and video processing, 3D reconstruction, computer vision and adjacent fields. His email address is msindeev@graphics.cs.msu.ru.

Vadim Konushin is a Ph.D. student at Keldysh Institute of Applied Mathematics, Russian Academy of Sciences. His research interests include image and video processing, pattern recognition, computer vision and adjacent fields. His email address is vadim@graphics.cs.msu.ru.

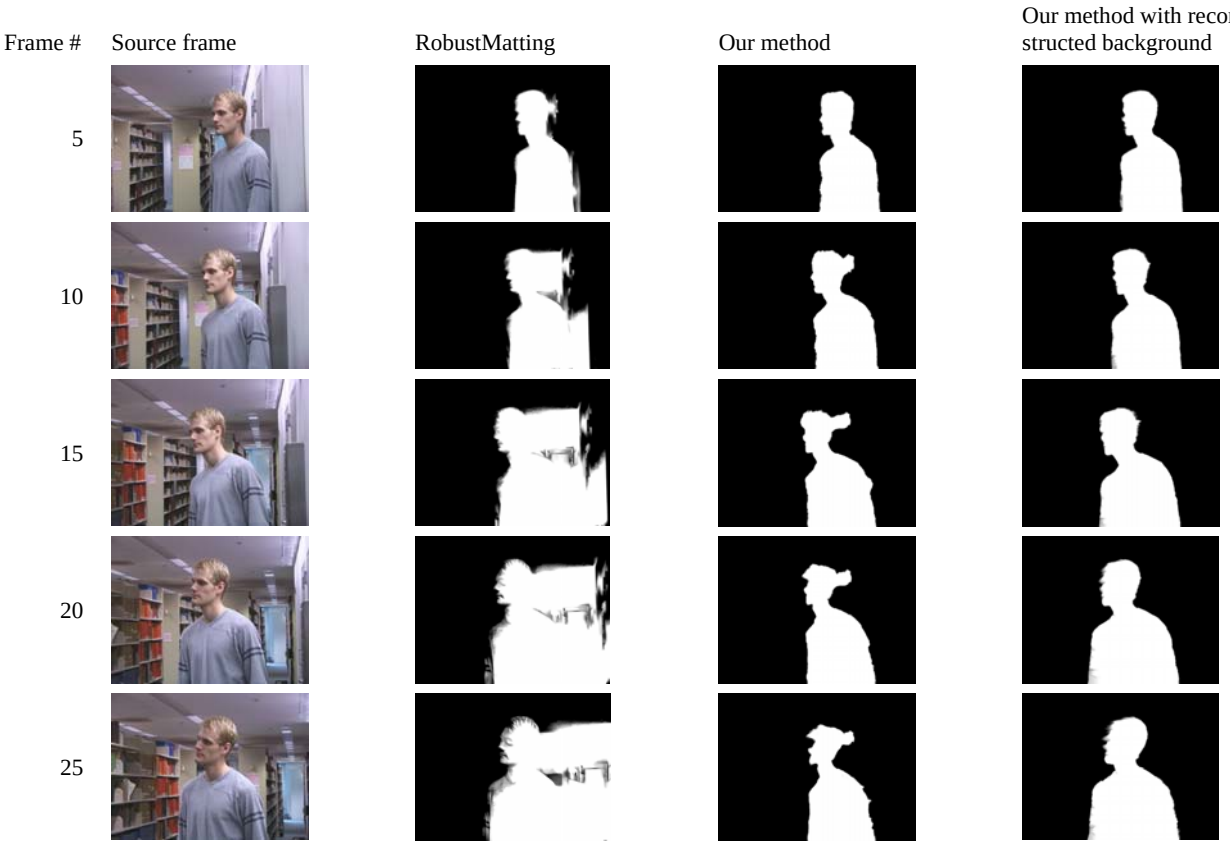


Figure 2. *Adam-lib-walk* videosequence from [1]. Keyframe was created for only the first frame.

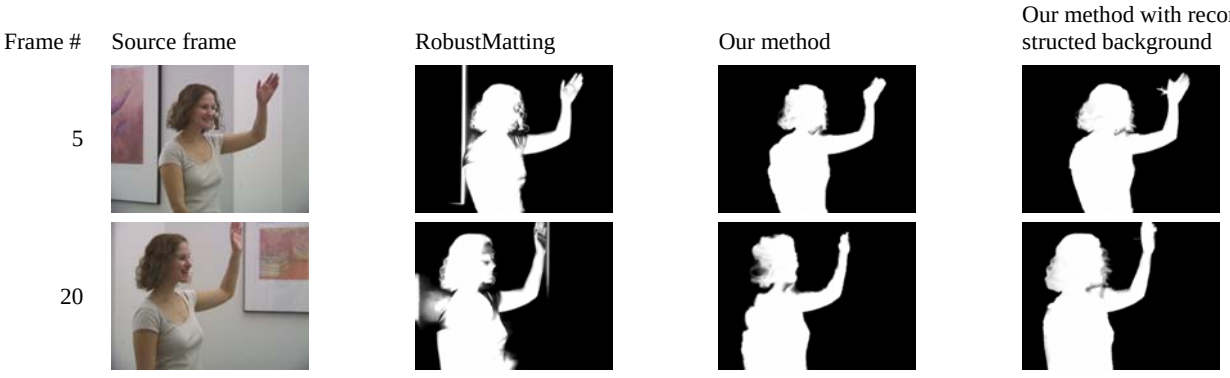


Figure 3. *Amira-queen* videosequence from [1]. Keyframe was created for only the first frame.

Построение 3D модели кровеносных сосудов по серии КТ изображений печени

Артем М. Ятченко*, Андрей С. Крылов*, Андрей В. Гаврилов**, Иван В. Архипов***
Факультет вычислительной математики и кибернетики*, НИИЯФ**
МГУ им. М.В.Ломоносова, РНЦХ РАМН***, Москва, Россия

АННОТАЦИЯ

Для планирования хирургической операции по пересадке фрагментов печени очень важно предоставить хирургу визуальную информацию о структуре кровеносных сосудов печени, их точном расположении внутри органа и о том, какие области печени омывает тот или иной сосуд. В данной статье для этого предлагается ряд методов построения и анализа объемных структур печени по серии рентгеновских компьютерных томографических снимков (КТ). В полуавтоматическом режиме сегментируются сосуды печени, строится скелет модели сосудов, по которому строится ориентированный граф, каждому ребру которого соответствует сосуд, а ориентации ребра – направление течения крови по нему. Интерактивно выделяются интересующие сосуды и сегментируется печень на области “омывания” этими сосудами.

1. ВВЕДЕНИЕ

При пересадке фрагментов печени пациенту пересаживают часть здоровой печени донора. Регенеративные свойства печени делают подобную операцию реальной. Однако очень важно заранее детально спланировать операцию – проверить совместимость донора с пациентом, определить области печени, подлежащие пересадке. 3D-представление кровеносных сосудов печени, а также визуальное отображение “омывания” тем или иным сосудом областей печени (рис. 1) повышает эффективность планирования [1].

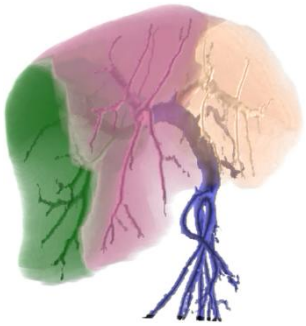


Рисунок 1: 3D-визуализация результата сегментации печени.

2. БИНАРИЗАЦИЯ

Первым шагом для построения 3D модели системы кровеносных сосудов является бинаризация изображения. Результатом бинаризации является 3D модель кровеносных сосудов.

Перед КТ сканированием, для того, чтобы лучше выделить сосуды, в кровь пациента вводится специальное вещество, называемое контрастом. В результате данного действия сосуды на изображениях становятся более яркими (рис. 2).

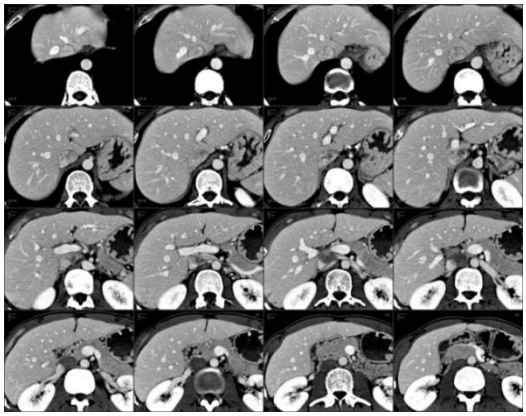


Рисунок 2: Серия КТ печени с введённым контрастом.

Для выделения кровеносных сосудов применяется итерационный алгоритм сегментации с переменным порогом яркости [2]:

1. Доктором вручную выбирается стартовый пиксель p_0 основания портальной вены (сосуда, по которому кровь притекает в печень). Его яркость принимается за начальный порог $\theta_0 = I(p_0)$.
2. Выбирается примыкающая к p_0 односвязная область пикселей, яркость которых не ниже текущего порога (θ_0), и эти пиксели заносятся в список $L_0 = L(\theta_0, p_0)$. Текущий порог уменьшается на единицу: $\theta_1 = \theta_0 - 1$.
3. Строится примыкающая к L_0 область с порогом θ_1 и пиксели этой области заносятся в список $L_1 = L(\theta_1, L_0)$.
4. Шаги 3 и 4 повторяются до тех пор, пока не будет достигнут порог θ_{end}. В качестве порога θ_{end} выбирается средняя яркость печени.

На данном этапе генерируется множество списков $L_0, L_1 \dots L_{end}$, соответствующих порогам $\theta_0, \theta_1 \dots \theta_{end}$. Пример визуализации этих списков приведён на рисунке 3. Доктору остаётся выбрать наиболее подходящий порог θ_{opt}.

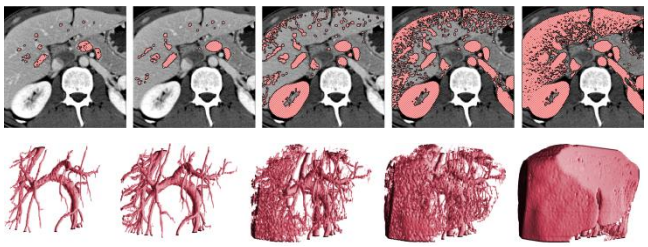


Рисунок 3: 2D и 3D визуализация модели кровеносных сосудов при различных выбранных порогах.

3. ПОДГОТОВКА ДЛЯ СКЕЛЕТИЗАЦИИ

Результаты, полученные на современном оборудовании, таком как КТ и МРТ, характеризуются очень большим размером 3D-данных высокого разрешения. Для обработки таких массивов данных должны применяться методы уменьшения размеров массивов, при этом сохраняя максимально возможное количество информации. Один из таких методов – скелетизация объектов.

Прежде чем применять скелетизацию, объект нужно подготовить к этой операции. Поскольку скелетизация сохраняет структуру объекта, она сохранит также и полости внутри объекта, оставив на их месте "пузыри" (рис. 4 (a)).

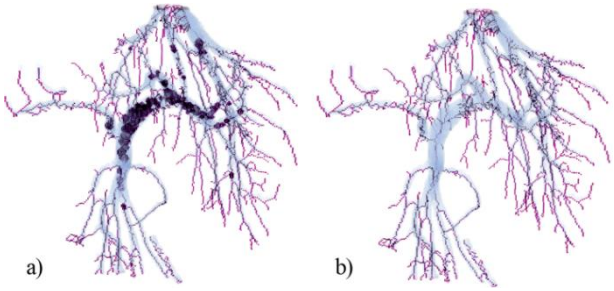


Рисунок 4: Пример скелетизации системы кровеносных сосудов без предварительного заполнения дыр (a) и с предварительным заполнением дыр (b).

Рассматриваемый объект – сосудистое дерево и полостей внутри объекта быть не должно. Чтобы предотвратить появление "пузырей" в скелете, перед скелетизацией нужно избавиться от полостей, возникающих из-за шумов на снимках.

Для заполнения полостей используется алгоритм последовательного сканирования: пиксели фона разбиваются на группы связности и оставляются лишь группы, прилегающие к границам куба данных. Остальные группы маркируются как пиксели сосудов. Пример заполнения дыр при формировании скелета приведен на рис. 4.

4. СКЕЛЕТИЗАЦИЯ

Скелетизация в трехмерном пространстве довольно сложная задача. Существует множество алгоритмов скелетизации в двухмерном пространстве, однако для 3D их не так много (см. [3], [4], [5]).

Используем понятия *6-соседства*, *18-соседства* и *26-соседства* (соседство по грани, по грани или ребру или по грани, ребру или вершине соответственно) [3].

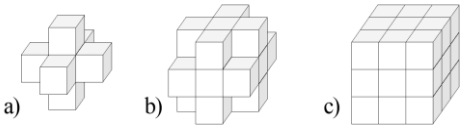


Рисунок 5: a – множество $N_6(p)$, b – множество $N_{18}(p)$, c – множество $N_{26}(p)$.

Будем обозначать множество всех *6*-соседей (*18*-соседей, *26*-соседей) пикселя p как $N_6(p)$ ($N_{18}(p)$, $N_{26}(p)$) (рис. 5).

Пиксели, принадлежащие объекту, будем называют *чёрными*, а пиксели, принадлежащие фону – *белыми*.

Для пикселя p множество всех соседним с ним чёрных пикселей обозначим как $S(p)$ (т.е. это множество всех чёрных пикселей из $N_{26}(p)\backslash p$). Множество всех соседних с p белых пикселей обозначим как $\overline{S}(p)$ ($\overline{S}(p) = N_{26}(p)\backslash(S(p) \cup p)$).

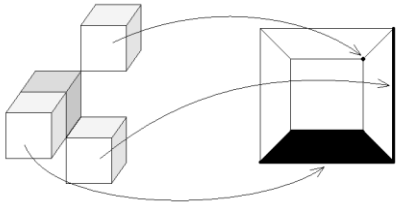


Рисунок 6: Пример соседних пикселей и соответствующая им диаграмма Шлегеля (соответствует виду сверху на p).

Диаграмма Шлегеля – схематическое планарное отображение соседних по *26*-соседству с p пикселей (рис. 6).

Пиксель называется *значимым*, если при его удалении нарушается структура объекта, т.е. выполняется хотя бы одно из следующих условий:

1. В $N_{26}(p)$ имеется не более одного черного пикселя.
2. Множество $S(p)$ и $\overline{S}(p)$ не связны.

Процесс скелетизации состоит в том, что удаляются все граничные не значимые пиксели до тех пор, пока такие пиксели существуют.

Для проверки пикселя на значимость, достаточно проверить количество его чёрных соседей и связность множеств $S(p)$ и $\overline{S}(p)$, однако такая проверка занимает много времени. Ниже приведен более быстрый алгоритм проверки значимости пикселей.

Для выбранного пикселя p обозначим за n_1 – количество вершин p, граничащих хотя бы с одним чёрным пикселем (кроме пикселя p), n_2 – количество ребер p, граничащих хотя бы с одним чёрным пикселем (кроме p), а n_3– количество граничных граней.

Числом Эйлера для окрестности S называется $\varepsilon(S(p)) = n_1 - n_2 + n_3$. При этом, если пиксель p не является значимым, то число Эйлера для него равно 1. Обратное, в общем случае, неверно.

Если у пикселя есть только один сосед, то такой пиксель будем называть *конечным* и будем считать его значимым.

Чтобы проверить, является ли пиксель значимым или нет в том случае, когда число Эйлера $\varepsilon(S(p)) = 1$, достаточно рассмотреть 4 случая: $n_3 = 0$, $n_3 = 1$, $n_3 = 2$ и $n_3 > 2$:

1. При $n_3 > 2$ пиксель не будет значимым.
2. При $n_3 = 2$ пиксель будет значимым, если либо существует изолированная вершина на диаграмме Шлегеля, либо $n_2 = 9$.
3. При $n_3 = 1$ пиксель будет значимым, если существует изолированная вершина на диаграмме Шлегеля, либо $n_2 = 8$.
4. При $n_3 = 0$ пиксель будет значимым, если существует изолированная вершина на диаграмме Шлегеля. Если изолированных точек нет (а также нет граней, т.к. $n_3 = 0$), то на диаграмме присутствуют только ребра. Вариантов, когда пиксель значимый, в этом случае довольно много, но все эти случаи можно просчитать заранее и занести в таблицу размером 512 байт (всего

ребер – 12, следовательно различных вариантов присутствия этих ребер может быть $2^{12} = 4096$ бит).

Справедливость случаев 1, 2 и 3 проверяется перебором всех вариантов соседей пикселя.

5. ПОСТРОЕНИЕ ГРАФА

После того как был получен скелет, количество рассматриваемых пикселей значительно уменьшилось при том, что сохранилась структура кровеносной системы.

Чтобы получить полную информацию о структуре модели, нужно построить ориентированный граф, каждому ребру которого соответствует сосуд, ориентации ребра – направление движения крови по сосуду, а вершинам – соединения сосудов или их окончания.

5. 1. Построение графа

Все пиксели скелета разбиваются на пиксели, принадлежащие ребрам графа, и пиксели, принадлежащие вершинам графа. Все пиксели ребер объединяются в группы связности и каждая такая группа считается ребром.

Таким образом получается граф $G = (V, E)$ (см. пример на рис. 7). Граф задается списком вершин V и списком ребер E.

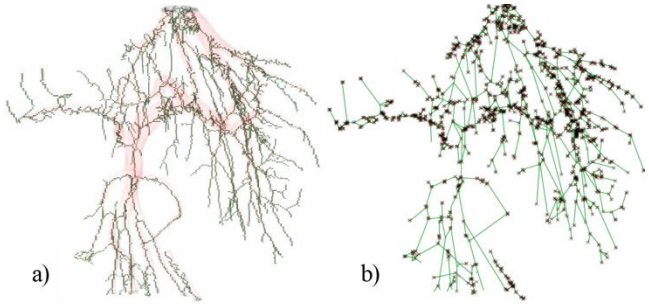


Рисунок 7: a – скелет системы кровеносных сосудов, b – граф кровеносной системы.

5. 2. Стрижка графа

Из-за шумов полученная модель исследуемой области может быть зашумлённой. Частично дефекты убираются с помощью заполнения полостей, однако помимо них есть ещё неровности стенок и сквозные отверстия в сосудах. Все эти шумы могут повлиять на структуру скелета. Такие ребра могут образовывать висячие ребра и циклы. Структуры, образованные шумами имеют размеры порядка толщины сосудов, что для кубов размером 512x512x512 пикселей обычно не превосходит 10 пикселей. Длины самих сосудов обычно составляют не менее нескольких десятков пикселей, так что избавиться от шумов можно разомкнув все маленькие циклы и убрав все короткие висячие ребра (стрижка).

На этапе стрижки размыкаются все циклы, длиной меньше порога θ_1 и удаляются все висячие ребра, длиной меньше θ_2.

На последнем этапе стрижки из графа выбрасываются фиктивные вершины: если в одной вершине сходятся ровно 2 ребра, то вершину можно удалить, а ребра объединить в одно.

Результатом является граф, ребра которого соответствуют сосудам, а вершины – соединениям сосудов или концам сосудов (см. пример на рис. 8).

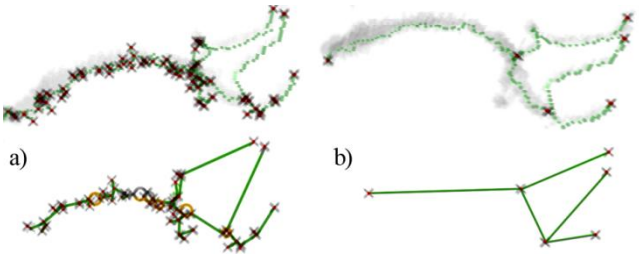


Рисунок 8: Скелет и соответствующий ему граф до стрижки (a) и после стрижки (b).

6. ОРИЕНТИРОВАНИЕ ГРАФА

Каждому ребру полученного графа соответствует отдельный сосуд, а каждой вершине – ветвление сосудов либо окончание сосуда. Следующий этап – определить, в каком направлении течёт кровь по сосудам.

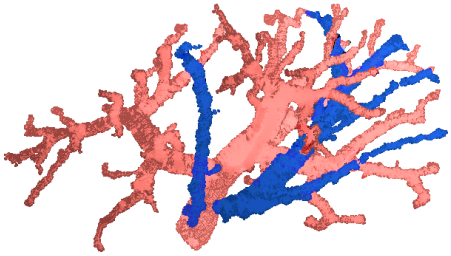


Рисунок 9: Портальная (светлая) и печеночная вена (темная).

Сосудистая система представляет собой набор отдельных сосудистых деревьев (рис. 9). Граф $G = (V, E)$ разбивается на несколько деревьев G_i, каждое из которых соответствует отдельному сосудистому дереву.

Основным критерием для ориентирования является то, что кровь течёт от более широких сосудов к более узким.

6. 1. Определение толщины сосудов

Для определения средней толщины сосудов 3D модель кровеносных сосудов разбивается на отдельные сосуды, и вычисляется среднее расстояние от границы каждого сосуда до его скелета.

Чтобы разбить всю модель на отдельные области, отнесём каждый пиксель модели к сосуду, соответствующему ближайшему участку скелета к этому пикселю (рис. 10).

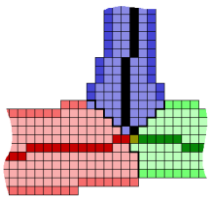


Рисунок 10: Пример выделения сосудов.

За среднюю толщину сосуда будем брать усредненное расстояние от всех пикселей границы сосуда до его скелета.

6. 2. Разбиение графа на отдельные деревья

Итак, дан связный граф G, каждому ребру которого дано в соответствие положительное число – толщина соответствующего этому ребру сосуда. Требуется разбить граф G на несколько деревьев G_i со следующими условиями: корень каждого дерева должен совпадать с корнем

сосудистого дерева, все пути внутри одного дерева от корня до висячих вершин соответствуют направлению течения крови по этим сосудам.

Сначала выбирается корень самого первого дерева G_0. В качестве корня берётся самый толстый из имеющихся сосудов e_0 и заносится в дерево G_0. Затем рассматриваются все соседние с e_0 сосуды и самый широкий из них e_1 заносится в G_0. Затем рассматриваются все соседние с e_0 и e_1 сосуды и выбирается самый широкий сосуд и так далее.

Если на очередном шаге из множества всех до сих пор неотмеченных сосудов самый широкий сосуд более чем в α раз ($\alpha > 1$ – параметр метода) шире самого широкого из соседей дерева G_0, то этот сосуд выбирается как корень нового дерева G_1. В реализованном алгоритме $\alpha = 2$.

Когда деревьев несколько – рассматриваются все примыкающие хотя бы к одному из деревьев ребра, и среди них выбирается самое широкое. Это ребро добавляется к тому дереву, к которому оно примыкает. Аналогично, если на каком-то шаге существует ребро, ширина которого больше чем в α раз превосходит самое широкое из граничащих ребер, то оно выносится как корень нового дерева.

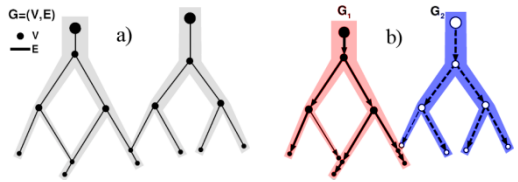


Рисунок 11: a – граф двух пересекающихся сосудов, b – ориентированные деревья.

Ориентация ребер проходит от корня к висячим вершинам (рис. 11).

7. ИНТЕРАКТИВНАЯ СЕГМЕНТАЦИЯ СОСУДОВ

Врачу на экране монитора компьютера предоставляется визуальное представление 3D-модели сосудов и скелета, на котором он может указать основания интересующих его ветвей сосудов, присвоить им названия и раскрасить их в произвольные цвета (рис. 12).

8. ЗАКЛЮЧЕНИЕ

Разработанные методы встроены в станцию обработки и анализа медицинских данных АРИС MultiVox 5.5. Испытания работы алгоритмов в Российском Научном Центре Хирургии РАМН (лаборатории Нагрузочных тестов и Абдоминальных методов исследования) показали их пригодность для использования в качестве диагностических инструментов.

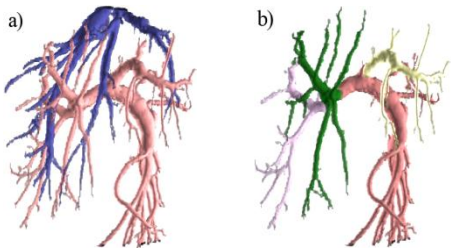


Рисунок 12: Выделенные сосудистые деревья (a) и вручную отсегментированные ветви (b).

Результаты использования разработанного алгоритма для планирования хирургических операций по пересадке фрагментов печени были представлены совместно с сотрудниками ГУ Российский научный центр хирургии им.Б.В. Петровского РАМН, в докладе на XX Всероссийском радиологическом съезде [6]. Работа выполнена при частичной поддержке ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009 – 2013 годы.

ЛИТЕРАТУРА

[1] Reitinger B., Bornik A. et al., *Liver Surgery Planning Using Virtual Reality// IEEE Comp. Graph. & Ap., v. 5, 2006, p.36-47.*

[2] Selle D., Preim B. et al., *Analysis of vasculature for liver surgical planning// IEEE Med. Imag., v. 21, 2002, p. 1344 – 1357.*

[3] Mian P., Gisela K., *A revision of a 3D Skeletonization algorithm// CITR, Univ. Auckland, NZ, 2004 Res. Tech. Rep. 143.*

[4] Antoine Manzanera, Thierry M. Bernard, *n-dimensional skeletonization: a unified mathematical framework// J. of Electronic Imaging, v. 11, 2002, p. 25 – 37.*

[5] Jonker P., *Morphological Operations on 3D and 4D Images: From Shape Primitive Detection to Skeletonization// DGCI, 2000, p. 371-391.*

[6] Ховрин В., Камалов Ю.Р., Ким Е.Ф., Филин А.В., Гаврилов А.В., Архипов И.В., Ятченко А.М., *Первый опыт применения отечественной рабочей станции MultiVox 2D/3D для оценки ангиоархитектоники печени у потенциальных родственных доноров фрагментов печени// Тезисы докладов XX Всероссийского радиологического съезда, 2009, с. 53-56.*

ABSTRACT

3D Liver Vessels Model Design Using CT Data

Methods for liver volume structures construction and analysis using roentgen computer tomography (CT) 2D data have been developed. The liver vessels are segmented semi-automatically, the vessel system skeleton is constructed and transformed into oriented graph. The vessels of interest are interactively selected to segment the liver into corresponding volume segments.

ОБ АВТОРАХ

Артем М. Ятченко – студент ф-та ВМК МГУ,
email: artyom@yatchenko.com.ua
Artem M. Yatchenko is a student of CMC MSU

Андрей С. Крылов – доцент, зав. лаб. математических методов обработки изображений ф-та ВМК МГУ,
email: kryl@cs.msu.ru
Andrey S. Krylov is associated professor, Head of the laboratory of mathematical methods of image processing, CMC MSU

Андрей В. Гаврилов – с.н.с. отдел микроэлектроники НИИЯФ МГУ
email: gavrilov@multivox.ru
Andrey V. Gavrilov is senior staff scientist SINP MSU

Иван В. Архипов – ведущий программист РНЦХ РАМН
email: arkhivania@gmail.com
Ivan V. Arkhipov is a software engineer of the NRCS RAMS

Построение модели динамики движения челюсти человека в процессе жевания по серии трехмерных изображений*

Дмитрий Гордеев, Наталья Дышкант
Факультет вычислительной математики и кибернетики
МГУ имени М.В. Ломоносова, Москва, Россия
dott1718@gmail.com, Natalia.Dyshkant@gmail.com

Аннотация

В работе рассматривается задача анализа изменений модели лица человека по серии последовательных трехмерных изображений — трехмерному видеоряду. Рассматривается видеоряд, полученный при съемке жующего человека. Предлагается подход к построению модели механического движения нижней челюсти в процессе жевания, основывающийся на выделении из модели динамической и статической частей. Такая модель может быть использована при биометрической идентификации личности, медицинской диагностике и оценке результатов операций.

Ключевые слова: 3D модель лица, биометрия, движение челюсти, трехмерное моделирование, подгонка поверхностей, захват движения.

1. ВВЕДЕНИЕ

В связи с активным развитием технологий трехмерного сканирования, число исследований, посвященных задачам анализа и классификации получаемых изображений, резко возросло. Решение таких задач актуально для многих приложений, связанных с биометрической идентификацией, распознаванием эмоций по лицу, медицинской диагностикой, планированием и оценкой результатов операций в косметологии и ортодонтии (см. [1]).

Современные трехмерные сканеры позволяют не только получить модель лица необходимой точности, но и произвести в режиме реального времени съемку изменений лица, захватывая любые мимические движения, в том числе движения нижней челюсти, например, во время жевания или разговора.

В последние годы технологии трехмерного моделирования получили широкое распространение в стоматологии и косметологии: работа с 3D моделями позволяет производить предклинические исследования, планировать операции и оценивать возможные результаты лечения. В [2] предлагается метод лазерного сканирования и система визуализации для виртуального планирования операций. В [3] предлагается использование фотограмметрической системы для получения трехмерных моделей челюстей и лица и дальнейшего определения взаимного расположения нижней и верхней челюстей. В [4] предлагается метод моделирования челюсти пациента по снимкам, полученным компьютерным томографичским сканированием.

Задачи анализа динамики движения челюсти и построение модели такого движения актуальны и востребованы для исследований в областях хирургической стоматологии и челюстно-лицевой хирургии [5].

В статье рассматривается задача регистрации изменений модели лица и нижней челюсти по трехмерному видеоряду, полученному при съемке жующего человека. Для этого в модели выделяются статические и динамические для исследуемого видеоряда области.

Исходными данными в решаемой задаче являются модели, заданные в виде облаков точек: 1) статичная трехмерная модель лица S; 2) видеопоследовательность из трехмерных изображений $D_1, D_2, \ldots, D_n$.

Анализируя данные, полученные в результате обработки последовательности кадров, необходимо произвести сегментацию модели и описать движение динамической области относительно статической в процессе съемки.

Таким образом, поставленная задача разбивается на следующий ряд подзадач: 1) «подгонка» моделей — введение меры различия двух моделей и ее минимизация в зависимости от их взаимного расположения; 2) выделение в модели статической и динамической областей; 3) «подгонка» статической и динамической частей и определение их взаимного расположения.

2. МЕТОД СЕГМЕНТАЦИИ МОДЕЛЕЙ

Каждая исходная модель задана в трехмерном пространстве набором координат точек: $\{x_i, y_i, F(x_i, y_i)\}_{i=1}^N$. Обозначим набор точек плоскости, на котором задается поверхность модели за G: $G = \{x_i, y_i\}_{i=1}^N$.

2.1 Стандартная система координат

На первом шаге работы происходит нормализация моделей: с каждой из моделей $S, D_1, D_2, \ldots, D_n$ связывается своя система координат.

Съемка лиц для рассматриваемой базы производилась в фас, и сканер выдавал полученное облако точек в системе координат, в которой ось Oz располагалась вдоль оси визирования и была направлена в камеру, ось Oy проходила вдоль лица по направлению от подбородка ко лбу, ось Ox — поперек лица, от правой щеки к левой. В дальнейшем подразумевается, что во всех обрабатываемых моделях лиц оси координат направленны таким же образом, и под «верхней» и «нижней» частями модели понимаются части с большими и, соответственно, меньшими значениями координаты Oy. В процессе нормализации первоначальная система координат преобразовывается с помощью параллельного переноса в новую таким

*Работа выполнена при финансовой поддержке РФФИ: проекты 08-07-00305, 09-07-92652.

образом, чтобы для модели были выполнены следующие условия:

$$\begin{cases} \sum\limits_{(x,y)\in G} x = \sum\limits_{(x,y)\in G} y = 0; \\ \max\limits_{(x,y)\in G} F(x,y) = 0. \end{cases}$$

Здесь суммирование производится по всем точкам (x,y) из G. Описанную систему координат будем называть *стандартной* для конкретной модели.

2.2 Сегментация

При движении нижней челюсти на снимках видеоряда наибольшей статической частью является верхняя часть лица, к которой относятся лоб и нос; нижняя часть лица (щеки, губы и подбородок) относятся к динамической части.

Для сравнения двух моделей лица со снимков предлагается подход, заключающийся в определении статической части для обоих снимков методом подгонки и описании движения движущейся (динамической) части относительно статической.

Пусть S_B, D_B — статические (верхние) части модели S и модели D из видеоряда, а S_H, D_H — динамические части этих моделей (см. Рис. 1); O_1, O_2 — системы координат, связанные с моделями S и D соответственно.

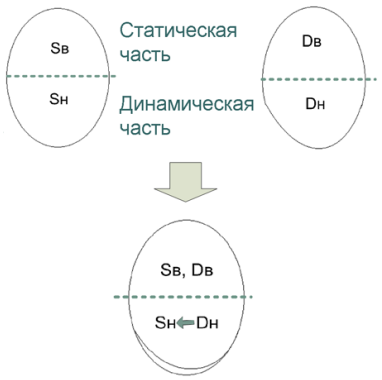


Figure 1: Подгонка статических частей двух моделей. S_B, S_H — верхняя и нижняя части статичной модели S; D_B, D_H — верхняя и нижняя части модели D видеоряда.

Модель D приводится к стандартной системе координат модели S и статические части S_B, D_B сопоставляются друг с другом методом подгонки так, чтобы мера различия между ними была наименьшей. При этом динамические части моделей S_H и D_H будут отличаться. Задача описания динамики движения подвижной части модели относительно статической состоит в регистрации таких различий для каждого снимка D из видеоряда.

Для **выделения статических частей** модели разделяются плоскостью P, первоначально совпадающей с плоскостью Oxz, верхние части моделей сравниваются, далее плоскость P смещается вдоль оси Oy с некоторым шагом и отделяемые ей верхние части моделей снова сравниваются. При этом используется мера различия, учитывающая площадь, на которой задана верхняя часть модели, отсекаемая плоскостью P. Смещение плоскости P происходит до тех пор, пока мера различия поверхностей уменьшается.

Для **выделения динамических частей** используется похожий метод. Первоначальное положение плоскости P устанавливается на некотором уровне (отсекающем заранее определенную "нижнюю"часть модели — с наименьшими y-координатами). Затем плоскость P смещается с некоторым шагом до тех пор, пока мера различия уменьшается. Эти смещения производятся в обоих направлениях, а в качестве динамической части определяется часть с минимальным значением меры различия из полученных двух.

Из-за неустойчивости значений функционала при смещении плоскости P в качестве сглаживающего значения предлагается использовать минимальное значение нескольких последних итераций. В качестве ответа предлагается использовать ту часть, на которой было зафиксировано минимальное значение функционала среди всех вычисленных.

В качестве метода сравнения поверхностей используется метод, предложенный в [6] и более подробно описанный в [7], позволяющий вычислять меру различия между поверхностями, заданными функциями на двух разных нерегулярных сетках. Данный метод основан на аппроксимации поверхностей кусочно-линейными функциями по триангуляциям Делоне. Представленный алгоритм является обобщением изложенного в [6], [7] с применением различных полуметрик и методов оптимизации.

Используемые меры различия между поверхностями двух моделей и методы оптимизации функционала различия между ними приведены в следующем разделе. Сопоставление (подгонка) поверхностей состоит в нахождении такого движения, при котором мера различия между поверхностями, описывающими лица, минимальна (см. Рис. 2).

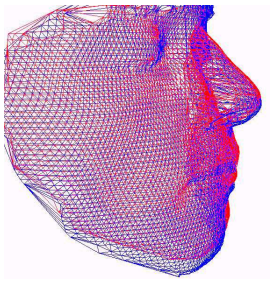


Figure 2: Пример подгонки двух моделей. Слабое отклонение в статических частях - нос и лоб, сильное отклонение в динамической части - челюсти

3. ФУНКЦИОНАЛЫ РАЗЛИЧИЯ ПОВЕРХНОСТЕЙ

Пусть M — движение в пространстве $\mathbb{R}^3$. Далее будем рассматривать его как композицию последовательных поворотов на углы $\alpha_M, \beta_M, \gamma_M$ вокруг осей Ox, Oy, Oz соответственно и параллельного переноса на вектор $(\Delta x_M, \Delta y_M, \Delta z_M)$.

Пусть модели D_1 и D_2 заданы как однолистные функции на двумерных сетках G_1 и G_2.

Определение 1. Пусть на множестве однолистных

поверхностей задана полуметрика ρ. *Мерой различия двух однолистных поверхностей* D_1 и D_2 назовем величину $\inf\limits_{M\in\mathscr{M}} \rho(D_1, M(D_2))$, где $\mathscr{M}$ — множество движений в пространстве, сохраняющих свойство однолистности поверхности G_2.

Определение 2. *Внутренним подмножеством* двух двумерных сеток G_1 и G_2 назовем множество $Int_{G_1,G_2} = (G_1 \cup G_2) \cap Conv(G_1) \cap Conv(G_2)$, где $Conv(G)$ — выпуклая оболочка множества G.

Определение 3. *Совместная триангуляция Делоне* T двух сеток G_1 и G_2 — триангуляция Делоне, построенная на множестве узлов Int_{G_1,G_2}. Обозначим $N_T = |Int_{G_1,G_2}|$

3.1 Меры различия

Рассмотрим примеры мер различия двух однолистных поверхнстей D_1 и D_2, заданных в виде функций F_1 и F_2 на множестве Int_{G_1,G_2}.

Объем симметрической разности. Рассмотрим непрерывные функции $\hat{F_1}$ и $\hat{F_2}$, заданные на $\mathbb{R}^2$ и полученные линейной интерполяцией функций F_1 и F_2 по точкам сеток G_1 и G_2 соответственно. Таким образом, $\hat{F_1}$ и $\hat{F_2}$ задают триангулированные поверхности. Введем обозначение

$$V(A, B, C, F_1, F_2) = \iint\limits_{\triangle ABC} |\hat{F_1}(x, y) - \hat{F_2}(x, y)| dxdy.$$

Эта мера различия вычисляется по формуле

$$\rho_V(D_1, D_2) = \sum_{\triangle ABC \in T} V(A, B, C, F_1, F_2),$$

где за T обозначена совместная триангуляция Делоне сеток G_1 и G_2, а суммирование происходит по всем треугольникам из T.

Среднее осевое расстояние.

$$\rho_M(D_1, D_2) = \sum_{(x,y)\in Int_{G_1,G_2}} \frac{|F_1(x,y) - F_2(x,y)|}{N_T}.$$

Отсеченное осевое расстояние. Пусть задано число $0 \le \alpha \le 1$. Тогда величина $\rho_M^\alpha(D_1, D_2)$ такова, что

$$\begin{cases} \left|\left\{(x,y) : |F_1(x,y) - F_2(x,y)| \le \rho_M^\alpha\right\}\right| \ge \alpha * N_T; \\ \left|\left\{(x,y) : |F_1(x,y) - F_2(x,y)| \ge \rho_M^\alpha\right\}\right| \le (1-\alpha) * N_T. \end{cases}$$

Здесь суммирование происходит по всем точкам $(x, y) \in Int_{G_1,G_2}$. Мера различия ρ_M^α не требует построения объединенной триангуляции Делоне для двух сеток, на которых заданы модели.

3.2 Оптимизация функционала различия

Подгонка двух поверхностей заключается в минимизации меры различия между ними. Рассмотрим следующую оптимизационную задачу в пространстве $\mathbb{R}^6$: $\rho(G_1, M(G_2)) \to \inf$, где инфинум берчтся по всем шести параметрам движения M сетки G_2.

Так как ρ является полуметрикой, то минимум функционала равен 0 и достигается при таком сдвиге модели, когда $Conv(G_1) \cap Conv(G_2) = \emptyset$. Таким образом, при оптимизации необходимо использовать метод поиска локального экстремума, а не глобального. Кроме того, выбор начального приближения метода является критичным. Описанная выше процедура приведения моделей к стандартной системе координат является эффективным решением последней проблемы.

Предлагаются следующие методы поиска локального экстремума меры различия.

Метод покоординатного спуска. Как известно, суть метода заключается в фиксировании всех переменных за исключением одной и решении задачи минимизации для получившейся функции одной переменной. Данный метод весьма эффективен при решении поставленной задачи, однако его главным недостатком является медленная скорость работы из-за большого количества вычислений функционала ρ.

Метод Нелдера-Мида (метод деформируемого многогранника, симлекс-метод). Идея метода состоит в сравнении значений функции в вершинах начального симплекса и перемещении симплекса по направлению оптимальной точки с помощью итерационной процедуры. Как показали проведенные эксперименты, в рассматриваемой задаче этот метод проигрывает в качестве оптимизации методу покоординатного спуска, однако скорость его работы значительно выше.

4. ЭКСПЕРИМЕНТЫ

Съемка для настоящего исследования производилась трехмерным сканером Broadway^{TM} компании «Artec Group» (`http://artec-group.com`), позволяющим делать до 15 снимков в секунду; полученные трехмерные модели лица задаются облаками точек.

В рамках исследования были проведены вычислительные эксперименты на видеоряде из 83 моделей, полученном при съемке жующего человека. Количество точек в моделях варьируется от 4 000 до 5 000.

Проведенные эксперименты показали, что самая быстрая оптимизация функционала различия достигается в случае использования среднего осевого расстояния в качестве меры различия и метода Нелдера-Мида.

4.1 Выделение статической части

В процессе жевания большая часть верхней части человеческого лица остается неподвижной, и алгоритм подгонки дает хорошие результаты.

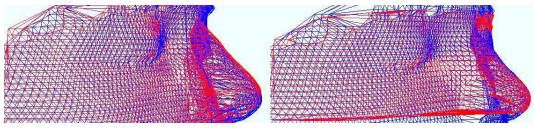


Figure 3: Сравнение качества наложения верхней части лица по всей модели (слева) и по сегментированной статической части (справа)

Несмотря на неплохое наложение верхней части лица при подгонке по всей модели, на Рис. 3 видно заметное улучшение при подгонке отдельно по сегментированной статической области.

4.2 Выделение динамической части

Для определения начального положения отсекающей плоскости был проведен вычислительный эксперимент на моделях из видеоряда $D_1, D_2, \ldots, D_n$. Динамическую область задается точками с координатой y меньше определенного значения. На Рис. 4 представлен график меры различия между динамическими областями в

зависимости от доли отсекаемой части лица (от 1% до 100% с шагом в 1%).

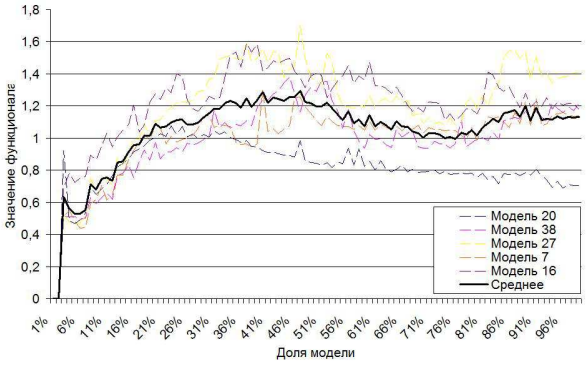


Figure 4: Зависимость меры различия между динамическими областями от доли динамической области в модели лица

Как показали эксперименты, в промежутке от 2% до 8% локальные минимумы функционала изменяются мало, что позволяет сделать предположение о том, что в качестве приближения динамической части можно использовать нижние 8% точек исходной модели (см. Рис. 5).

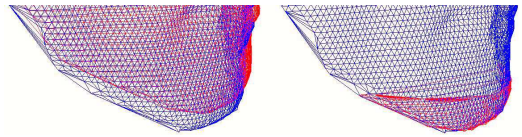


Figure 5: Сравнение качества наложения нижней челюсти по всей модели (слева) и по сегментированной динамической части (справа)

Преобразование системы координат, связанной с динамической областью, в систему координат, стандартную для исходной статичной модели, описывает динамику движения нижней челюсти. Формальное описание движения было получено в виде серии матриц преобразования и может быть представлено как анимация движения базисных векторов.

5. ЗАКЛЮЧЕНИЕ

В данной работе предложен подход к построению модели движения нижней челюсти человека в процессе жевания по последовательности трехмерных изображений. В качестве ключевых подзадач были рассмотрены задачи определения меры различия двух однолистных поверхностей, был предложен алгоритм выделения из трехмерного изображения динамичской и статической частей.

Изложенные в данной статье принципы легко применимы к другим задачам сегментации и определения динамики трехмерных изображений при условии наличия каких-либо эвристических подходов определения динамической части модели.

6. REFERENCES

[1] Hajeer MY, Millett DT, Ayoub AF, Siebert JP. Applications of 3D imaging in orthodontics // Journal of Ortodontics, 31(1), P.62–70, 2004.

[2] Koidis P., Patias P., Tsioukas V. 3D Visualization of Dental Data for Virtual Treatment Planning // ISPRS Congress Istanbul 2004, Proceedings of Commission V, 2004. — P. 996–1001.

[3] Knyaz V. A., Zheltov S.Yu. Photogrammetric Techniques for Dentistry Analysis, Planning and Visualisation // ISPRS Congress Beijing 2008, Proceedings of Commission V, 2008. — P. 783–788.

[4] Enciso, R., Memon, A., Fidaleo, D. A., Neumann, U., Mah J. The Virtual Craniofacial Patient: 3D Jaw Modeling and Animation. // The 11th Annual Medicine Meets Virtual Reality Conference, P. 65–71, 2003.

[5] Michihiko Koseki, Akira Niitsuma, Norio Inou and Koutarou Maki Three-dimensional Display System of Individual Mandibular Movement // Complex Medical Engineering, (Springer), P. 117–127, 2007.

[6] Дышкант Н.Ф., Местецкий Л.М. Сравнение однолистных поверхностей полученных при 3D сканировании // Proceedings of «GraphiCon'2008», P. 270–277, 2008.

[7] Natalya Dyshkant. Disparity measure construction for comparison of 3D objects' surfaces // Proceedings of the Workshop IMTA, Lisbon, Portugal, P. 43–52, 2009.

ОБ АВТОРАХ

Дмитрий Гордеев — студент факультета ВМиК МГУ имени М.В. Ломоносова. E-mail: dott1718@gmail.com.

Наталья Дышкант — аспирантка факультета ВМиК МГУ имени М.В. Ломоносова. E-mail: Natalia.Dyshkant@gmail.com.

CONSTRUCTION OF JAW MOVEMENT MODEL DURING CHEWING PROCESS BY 3D IMAGE SEQUENCE

Abstract

We consider a problem of 3D face model's changes registration by consistent series of 3D images — 3D video sequence. We introduce an approach for construction of jaw movement model during chewing process. The approach is based on segmentation of 3D face to static and dynamic parts. Such model can be used in biometric identification, medical diagnostics and estimates of surgery results.

Keywords: *3D face, biometrics, jaw movement, 3D modeling, surface matching, motion capture.*

ABOUT THE AUTHORS

Dmitry Gordeev — student of Computational Mathematics and Cybernetics department of Moscow State University. E-mail: dott1718@gmail.com.

Natalia Dyshkant — PhD student of Computational Mathematics and Cybernetics department of Moscow State University. E-mail: Natalia.Dyshkant@gmail.com.

Fast Weak Learner Based on Genetic Algorithm*

Boris Yangel
Department of Computational Mathematics and Cybernetics
Lomonosov Moscow State University, Moscow, Russia
hr0nix@acm.org

Abstract

An approach to the acceleration of parametric weak classifier boosting is proposed. Weak classifier is called parametric if it has fixed number of parameters and, therefore, can be represented as a point in multidimensional space. Genetic algorithm is used to learn parameters of such classifier. Proposed approach also takes cases when effective algorithm for learning some of the classifier parameters exists into account. Experiments confirm that such an approach can dramatically decrease classifier training time while keeping both training and test errors small, at least for some widely used pattern recognition algorithms.

Keywords: *boosting, genetic algorithm, classification, haar feature.*

1. INTRODUCTION

Boosting [1] is one of the commonly used classifier learning approaches. It is machine learning meta-algorithm that iteratively learns additive model consisting of weighed *weak* classifiers that belong to some classifier family W. In case of two-class classification problem (which we will consider in this paper) boosted classifier usually has form

$$s(x) = \operatorname{sgn}\left(\sum_{i=1}^{N} \alpha_i w_i(x)\right). \quad (1)$$

There $x \in X$ is a sample to classify, $w_i \in W$ are weak classifiers learned during boosting procedure, α_i are weak classifier weights, $w_i(x) \in \{-1, 1\}$, $s(x) \in \{-1, 1\}$. Set W is referred to as *weak* classifier family. That is because it elements should have error rate only slightly better than random guessing. It expresses the key idea of boosting: strong classifier can be built on top of many weak.

There are many boosting procedures that differ in the type of loss being optimized for the final classifier. But no matter what kind of boosting procedure is used, on each iteration it should select (learn) a weak classifier with minimal weighed loss from W family using special algorithm called *weak learner*. For some widely used families of weak classifiers that process can take a lot of time. For example, simple classifiers called *stumps* are often used in boosting. Those classifiers usually have form

$$w_i(x) = \operatorname{sgn}[g_i(\phi_i(x) - t_i)], \quad (2)$$

where $\phi_i \in \Phi$ is some object feature, t_i is a threshold and $g_i \in \{-1, 1\}$ controls the sign of the weak classifier output. When given some feature ϕ_i, optimal values for classifier parameters t_i and g_i can be calculated rather fast for most of the widely used loss functions. Problem of finding good feature ϕ_i is much harder. Exhaustive search over feature space Φ is often used. It's not a problem when object has few features, but sometimes feature space can be really huge.

*This work was supported by the Russian Fund for Fundamental Research through grant Nos. 08-07-445-a and 08-07-12081.

In pattern recognition, values of some function over various subregions of an image are often considered as the features of that image. Even small image has a lot of possible subregions, so exhaustive search over feature space becomes very expensive. For example, learning cascade of boosted stump classifiers based on haar features with *AdaBoost* and exhaustive search over feature space took several weeks in the famous work [2]. That's why it is often very important to decrease weak classifier learning time using some appropriate optimization technique.

One of the widely used approaches to the numerical optimization is genetic algorithm [3]. It is based on the biological evolution ideas. Optimization problem solution is coded as *chromosome* vector. *Initial population* of solutions is created using random number generator. *Fitness function* is then used to assign fitness value to every population member. Solutions with the largest fitness values are selected for the next step. In the next step, *genetic operators* (crossover and mutation usually) are applied to selected chromosomes to produce new solutions and to modify existing ones slightly. Those modified solutions form up a new generation. Then described process repeats. That's how evolution is modeled. It continues until global or suboptimal solution is found or time allowed for evolution is over. Genetic algorithms are often used for global extremum search in big and complicated search spaces.

2. RELATED WORK

Usage of genetic algorithm for weak learner acceleration was already proposed in several works. For example, in [4] genetic weak learner with special crossover and mutation operators was used to learn classifier based on extended haar feature set. In [5] genetic algorithm was used to select a few thousand weak classifiers with smallest error on unweighted training set before boosting process starts. Then exhaustive search over selected classifiers was performed on each boosting iteration to select the one with minimal weighed loss. In [6] boosting procedure was completely integrated with genetic algorithm. Few classifiers were selected on each boosting iteration from solution population and added to the strong classifier. Those selected classifiers were then used to produce new population members by applying genetic operators. Then, in [7] authors used special evolutionary algorithm they've called *Evolutionary Hill-Climbing* as a weak learner. Crossover operator was not used in it. Instead, 5 different mutations were applied to every population member on each algorithm iteration. Result of each mutation was rejected when it did not improve fitness function value.

There were two main reasons for using genetic search instead of any other approaches in these works. Most of the classifiers used in mentioned works were some extensions of the haar classifier family originally proposed in [2]. So, huge size of the feature space and, therefore, huge size of the weak classifier family did not allow to apply exhaustive search based optimization. And complicated discrete structure of a weak classifier blocked all other optimization options.

Another important observation is the fact that all the authors of the mentioned papers were forced to implement some specialized solu-

tion for genetic weak learner. So, ability to generalize evolutionary approach to learning weak classifier is investigated in this work.

3. PROPOSED METHOD

We are interested in developing some general approach to weak classifier learning. And we are especially interested in approach that will handle classifiers of form (2) because those classifiers are widely used in pattern recognition area. This approach should work much faster than exhaustive search over classifier parameter space. In the following document sections one such approach is presented. It is based on the fact that when number of classifier parameters to optimize is fixed, weighed loss optimization problem simply turns out into multivariate function minimization problem which is well-developed area of genetic algorithm application.

3.1 Population member

Let W be some parametric family of weak classifiers. It means that every weak $w \in W$ can be described by a set of its n real-valued parameters $x_1, \ldots, x_n$. Let's also assume that for the last l parameters (l can be equal to zero) there exists some effective learning algorithm $L_E : \mathbb{R}^{n-l} \to \mathbb{R}^l$. We will refer to such parameters as to *linked*. For given values of parameters $x_1, \ldots, x_{n-l}$, called *free*, L_E finds optimal values for linked parameters that minimize loss function $E : \mathbb{R}^n \to \mathbb{R}^+$. It means that our task is to find values of free parameters that deliver the minimum to the loss function $E[x_1, \ldots, x_{n-l}, L_E(x_1, \ldots, x_{n-l})]$. So, set of parameters $x_1, \ldots, x_{n-l}$ represents solution to our optimization problem and form up a member of genetic algorithm population.

For example, for the weak classifier of form (2) parameters t_i and g_i will be linked, while parameters describing feature ϕ_i will be made free. It should be mentioned that if small changes of feature representation can have unpredictable impact on the value of the loss function, genetic optimization turns into random search. It happens, for example, when feature can be simply described by its index in the feature pool and features with close indices are not correlated. On the other hand, small changes in the image region size or position usually lead to small changes in the region characteristic, at least for some "good" stable characteristic functions like pixel intensity sum in region. For such feature sets genetic optimization is possible.

3.2 Fitness function

It is natural to assume that classifier with small error on training set should have greater probability to get to the next generation of the genetic algorithm. That allows us to introduce fitness function $F : \mathbb{R}^{n-l} \to \mathbb{R}^+$ as follows:

$$F(x_1, \ldots, x_{n-l}) = \\ = 1/E[x_1, \ldots, x_{n-l}, L_E(x_1, \ldots, x_{n-l})]. \quad (3)$$

We do not consider $E = 0$ case. Classifier can not be called weak if it has zero error value on training set. If such a classifier is presented in a weak classifier family, we can select only that classifier as a whole boosting procedure result.

3.3 Genetic representation

Every approach that allows us to encode a set of free parameters is appropriate for population member representation. In this work we have selected binary string representation which was confirmed to be effective in function optimization problems. Some alternative representations can be found, for example, in [3].

To form the binary string classifier representation, each classifier parameter should be first represented as a binary string of fixed length, using fixed-precision encoding. Then all the parameters can be simply concatenated to form the final binary string of fixed length.

Sometimes point $p \in \mathbb{R}^n$ can have no corresponding classifier. For the different families of image region classifiers it is possible, for example, when one of the free parameters representing top-left corner of a classifier window is below zero. In this case fitness function value of the population member representing that point can be forced to be zero. That is how such situations were dealt with in experiments described in section 4.. Another possible approach is to select representation and genetic operators in a way that simply does not allow such points to appear. But that approach is less general.

3.4 Genetic operators

In this work we've used the two most common genetic operators: mutation and crossover. When binary string representation is chosen, mutation and crossover are usually defined as follows:

- Crossover operator selects random position in the binary string. Then it swaps all the bits to the right of the selected position between two chromosomes. Such crossover implementation is called 1-point crossover.
- Mutation operator changes value of the random chromosome bit to the opposite.

In our case, crossover operator produces two new solutions from the two given chromosomes as following: some of the parameters (placed to the left of the selected position) are taken from the first classifier, some of the parameters (placed to the right) — from the second. And one parameter, probably, can be made from both the the first and the second classifier. Mutation operator simply produces new solution by changing value of the random classifier parameter.

3.5 Algorithm summary

Algorithm 1 Genetic weak learner

1: Generate initial population of N random binary strings;
2: **for** $i = 1, \ldots, K_{max}$ **do**
3: Add $\lceil NR_c \rceil$ members to the population by applying crossover operator to the pairs of the random population members;
4: Add $\lceil NR_m \rceil$ members to the population by applying mutation operator to the random population members;
5: Calculate value of (3) for each population member;
6: Remove all the population members except of the N best (those with largest value of (3));
7: **end for**
8: **return** weak classifier associated with point represented by best population member as a result;

Algorithm 1 uses elitism as a population member selection approach. It has 4 parameters:

- $N > 0$ — population size.
- $K_{max} > 0$ — number of generations.
- $R_c \in (0, 1]$ — crossover rate.
- $R_m \in (0, 1]$ — mutation rate.

3.6 Discussion

Advantage of the proposed method lies in the fact that computational complexity of the weak learner does not depend on the size of the weak classifier family. One can achieve balance between training time and classifier performance only by changing values of N, K_{max} and S (discussed later). Similar effect can be achieved by shrinking weak classifier family itself. But in most cases prior knowledge about weak classifier performance in boosting is simply not available.

One of the main disadvantages of the proposed weak learner is the fact that many potentially interesting weak classifiers can not be represented as a parameter vector of constant length. For example, decision trees, widely used in boosting, can have variable number of nodes. Also, as it was already mentioned, misclassification loss we are trying to optimize should be more or less stable as a function of classifier free parameters. If small perturbations of the free parameter vector lead to the unpredictable changes in the loss function value, genetic optimization does not make much sense, becoming just a random search.

4. EXPERIMENTS

4.1 Algorithms for experiments

Two boosting-based algorithms were implemented to compare proposed genetic weak learner with original learners proposed by algorithm authors. *Viola-Jones* [2] and *Face alignment via boosted ranking model* [8] were selected for that purpose because both algorithms use parametric weak classifiers applied to image regions. These algorithms are based on distinct boosting procedures (*AdaBoost* and *GentleBoost*), so loss, sample weight and classifier weight functions used in them differ a lot. Another difference between selected algorithms is a problem they solve: two-class classification in [2] and ranking in [8]. Naïve weak learner implementation is quite slow in both algorithms, so acceleration of boosting process is necessary.

Weak classifiers used in both algorithms are based on haar features and have common set of adjustable parameters. So, weak classifier in both problems can be represented as $w_i = (x_i, y_i, width_i, height_i, type_i, g_i, t_i)$. There x_i, y_i, $width_i$ and $height_i$ describe image region, $type_i$ encodes haar feature type, g_i is a haar feature sign and t_i represents weak classifier threshold. Parameters g_i and t_i are linked because both algorithms have an effective algorithm for learning them. Parameter $type_i$ was also made linked: changing feature type during genetic optimization does not make much sense because it can change fitness function value significantly after just one mutation or crossover. Separate algorithm run was performed instead for each feature type. Best result from all the runs was then selected. We've used the same 5 haar feature types as in [8] for training both classifiers.

4.2 Run patterns

Comparison of two different genetic algorithm run patterns was also performed in this work. One pattern considered was running genetic optimization once with big population size. Another pattern used was running optimization algorithm multiple times (denoted as S) with small population size and then selecting best found classifier. When population size is small, final solution depends on initial population a lot. So, considerably different results can be obtained for different algorithm runs. While this run pattern produces worse classifiers, it can be implemented on multiprocessor and multicore architectures very efficiently: each processing unit can run its own genetic simulation. That makes perfect parallel algorithm acceleration possible.

Table 1: Viola-Jones, acceleration

Run pattern			Time (sec)	Acceleration
S	N	K_{max}		
1	50	10	**2.82**	**329.38**
1	100	20	9.40	98.77
1	400	40	100.29	9.26
10	10	20	4.00	231.94
20	20	40	28.74	32.31
Brute force			928.52	1.00

Table 2: Viola-Jones, error

Run pattern			Error	
S	N	K_{max}	Learning	Test
1	50	10	0.0005	0.0356
1	100	20	0.0002	0.0380
1	400	40	0.0000	**0.0328**
10	10	20	0.0003	0.0378
20	20	40	0.0000	0.0391
Brute force			0.0000	0.0349

4.3 Training and test sets

As in work [4], [9] human faces database was used to train and test classifier for Viola-Jones algorithm. Database was divided in half to form the training and test sets. Each sample has size of 24×24 pixels.

Face images with landmarks from FG-NET aging database were used to form the database for learning face alignment ranker proposed in [8]. 600 face images were selected from database and then resized to the size of 40×40 pixels. 400 images were used to produce training set and other 200 — for testing. 10 sequential 6-step random landmark position perturbations were then applied to selected face images to produce images of misaligned faces, as described in the original paper. Training and test set samples were then made of pairs of images with increasing alignment quality.

4.4 Hardware

All the experiments were performed on PC equipped with 2.33 GHz Intel Core 2 Quad processor and 2 GB of DDR2 RAM.

4.5 Results

Tables 1 and 3 show average duration of 1 boosting iteration together with comparison to exhaustive search. Tables 2 and 4 show error rate of the final classifiers on the training and test sets. We have not trained any classifier using exhaustive search for boosted ranking model because it would take about a year to finish the process on our training set.

Experiments with Viola-Jones object detector have shown that classifier trained using genetic weak learner performs only slightly worse than classifier trained using exhaustive search over feature space. For $N = 400$ final classifier even shows better performance. Classifier trained with $S = 1$, $N = 50$ and $K_{max} = 10$ accelerates boosting nearly 300 times compared to exhaustive search while still performing good on the test set. Classifiers trained with small values of N and big values of S (using second run pattern) perform worse than any other. But, as it was mentioned before, such classifiers can be trained on multiprocessor or multicore systems very efficiently.

Experiments with face alignment via boosted ranking model have shown how exactly classifier performance depends on values of S,

Table 3: Face alignment via BRM, acceleration

Run pattern			Time (sec)	Acceleration
S	N	K_{max}		
1	25	10	**68.15**	**5195.88**
1	50	10	173.33	2043.09
2	75	15	909.55	389.34
4	100	20	3582.37	98.85

Table 4: Face alignment via BRM, error

Run mode			Error	
S	N	K_{max}	Learning	Test
1	25	10	0.0278	0.0317
1	50	10	0.0246	0.0297
2	75	15	0.0199	0.0268
4	100	20	**0.0173**	**0.0259**

N and K_{max}. Increasing value of the each parameter results in increased training time, but also in increased classifier performance. Nevertheless, difference in training time is more significant compared to the difference in prediction error. Classifier with $S = 1$, $N = 25$ and $K_{max} = 10$ was trained 50 times faster than the best obtained classifier for BRM, but its error on the test set is only 1.2 times worse. It makes such a classifier a perfect candidate for preliminary experiments that usually take place before training of the final classifier starts.

5. CONCLUSION

An approach to boosting procedure acceleration was proposed in this work. Approach is based on usage of genetic weak learner for learning weak classifier of special parametric form on each boosting iteration. Genetic weak learner uses genetic algorithm with binary chromosomes. That genetic algorithm is designed to solve an optimization problem of selecting weak classifier with the smallest weighed loss from some parametric classifier family. Proposed method was generalized for the case when there exists an effective algorithm for learning some of the parameters of a weak classifier. Experiments have shown that such approach allows us to accelerate training process dramatically for practical tasks while keeping good generalization properties.

Genetic weak learner proposed in this work can't be used to boost any tree-based classifiers. That fact limits its usage in many scenarios because stump weak classifiers can not represent any deep interactions between different object features. So, in the future work we plan to generalize our approach for accelerating tree-based boosting.

Another option for future research is performing additional experiments with classifiers not related to haar features in any way. That will confirm the profit of the proposed algorithm in computer vision problems that are not biased towards haar feature usage. In fact, it would be nice to determine different parametric classifier families that can be efficiently boosted using proposed weak learner.

6. REFERENCES

[1] Robert E. Schapire, "The boosting approach to machine learning an overview," in *MSRI Workshop on Nonlinear Estimation and Classification*, 2002.

[2] Paul Viola and Michael Jones, "Robust real-time object detection," in *International Journal of Computer Vision*, 2001.

[3] David E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Professional, January 1989.

[4] Andre Treptow and Andreas Zell, "Combining adaboost learning and evolutionary search to select features for real-time object detection," 2004.

[5] Geovany A. Ramirez, "Face and street detection with asymmetric haar features," 2007.

[6] K. Masada, Qian Chen, Haiyuan Wu, and T. Wada, "Ga based feature generation for training cascade object detector," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 2008, pp. 1–4.

[7] Y. Abramson, F. Moutarde, B. Stanciulescu, and B. Steux, "Combining adaboost with a hill-climbing evolutionary feature search for efficient training of performant visual object detectors," in *FLINS06*, March 2006.

[8] Hao Wu, Xiaoming Liu, and Gianfranco Doretto, "Face alignment via boosted ranking model," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008, pp. 1–8.

[9] P. Carbonetto, "Viola-jones training data," `http://www.cs.ubc.ca/~pcarbo/viola-traindata.tar.gz`, 2002.

Модель улично-дорожной сети на основе скелета

Мехедов И. С.,* Козлов А. В.†

Аннотация

В работе рассматривается подход к построению модели улично-дорожной сети векторной карты на основе скелета многоугольной многосвязной фигуры. Модель представляет собой связный граф, топологически соответствующий конфигурации улично-дорожной сети. Приводится классификация перекрестков на основе топологии графа. Для одноуровневой модели (проезжие части всех дорог пересекаются на одном уровне) дается алгоритм ее построения.

Ключевые слова: *модель улично-дорожной сети, скелет, классификация перекрестков, осевая линия улицы.*

1. ВВЕДЕНИЕ

В настоящее время в некоторых геоинформационных системах и многих интерактивных картах используется модель улично-дорожной сети (УДС). Она предназначена для описания в виде графа схемы автомобильных дорог с целью последующего сетевого анализа (поиск кратчайшего по времени или расстоянию маршрута, поиск кратчайшего обхода заданного набора пунктов, поиск ближайших пунктов обслуживания). Также модель используется при анализе транспортной ситуации: ребрам графа могут быть приписаны различные дорожные характеристики, например, интенсивность движения между перекрестками, количество полос проезжей части и некоторые другие).

В статье рассматривается УДС г. Москвы. В качестве картографических данных УДС используется информация ресурса «Единая государственная картографическая основа г. Москвы» [9] (рис.1), в котором УДС представляет собой множество многоугольных фигур, географические координаты вершин которых с определенной точностью описывают реальные границы проезжей части.

Такой вид представления пространственных данных позволяет вычислить многие характеристики УДС, например: ширину проезжей части в любой точке, протяженность перегонов между перекрестками, – а также определить область и конфигурацию практически любого перекрестка на карте. Инструментом проведения такого анализа свойств УДС является модель УДС, описываемая в статье.

Под моделью городской УДС будем понимать связный граф, каждой вершине которого сопоставлена пара географических координат, а также некоторым вершинам и ребрам поставлены в соответствие числовые или качественные характеристики. Такой граф при этом должен топологически корректно соответствовать конфигурации УДС.

В том случае, когда все проезжие части улиц пересекаются на одном уровне, говорят об одноуровневой УДС (рис.1а). Многоугольные фигуры, составляющие УДС, могут быть объединены в одну многоугольную фигуру, граница которой определяет границу всей УДС города. Если же проезжие части могут находится друг под другом, то речь идет о многоуровневой УДС (рис.1б). Последняя практически всегда является составляющей крупного мегаполиса за счет наличия транспортных развязок. В этом случае объединение многоугольных фигур УДС образует некоторую поверхность с границей, задающей границу всей УДС. Заметим, что любая многоуровневая УДС может быть разбита на одноуровневые фрагменты.

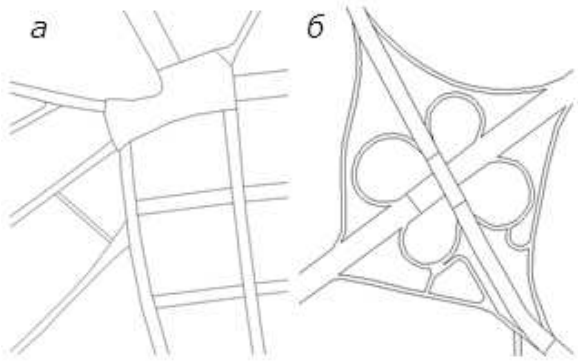


Рисунок 1: Фрагменты улично-дорожной сети Единой Государственной Картографической Основы г. Москвы. Одноуровневая (а) и многоуровневая УДС (б).

Модель многоуровневой УДС является предметом будущих исследований авторов. В настоящей статье рассмотрим построение модели одноуровневой УДС (или одноуровневых фрагментов многоуровневой УДС) – одноуровневой модели – на основе множества многоугольных фигур.

2. ПОДХОД К ПОСТРОЕНИЮ МОДЕЛИ

Исходными данными является множество многоугольных фигур (*элементов УДС*), любые две из которых не имеют общих точек, за исключением, быть может, общего участка границы. Требуется построить модель УДС в виде связного плоского графа, обладающего следующими свойствами:

1. граф должен топологически корректно соответствовать конфигурации УДС;
2. должно существовать *адекватное* сюръективное отображение множества вершин графа на множество перекрестков УДС;
3. должно существовать инъективное отображение множества *простых цепочек* графа (т. е. цепей графа, все внутренние вершины которых имеют степень 2, а концевые вершины – степень, отличную от 2) на множество участков УДС между перекрестками;
4. граф должен служить инструментом анализа некоторых характеристик УДС (о них уже было упомянуто выше).

Элементы одноуровневой УДС могут быть объединены в сложную многосвязную многоугольную фигуру. Математической моделью, наиболее точно описывающей топологию такой фигуры, является *скелет*, т.е. множество всех центров максимальных пустых кругов фигуры [2], поэтому в статье рассматривается метод построения модели УДС на основе скелета.

*Вычислительный центр РАН, e-mail: mehedov@mail.ru

†ООО «Лаборатория транспортных систем», e-mail: kav57@inbox.ru

Переход к линейной модели от совокупности площадных объектов является известной задачей геоинформатики. В известных работах по этой тематике также используется скелет многоугольной фигуры. Так, в [6, 7] линейная модель УДС строится на основе *спрямленного скелета (straight skeleton)*, подробно описанного в [5]. Недостатками такого скелета является неопределенность *радиальной функции* точки скелета, т. е. величину радиуса пустого круга с центром в этой точке [2]. В [8] используется скелет на основе *диаграммы Вороного* [2], но он плохо применим к сложным многоугольным фигурам с большим количеством «дыр». Вообще, большинство известных алгоритмов скелетизации хорошо работают с простыми многоугольниками, но либо «боятся» многосвязных многоугольных фигур, либо имеют большую вычислительную сложность.

Для построения модели будем использовать алгоритм скелетизации, предложенный в [3], применимый к произвольным многоугольным фигурам и имеющий сложность $N \log N$ (N – число вершин фигуры).

Основой модели служит не весь скелет, а его подграф, устойчивый к граничным шумам (незначительным «неровностям» границы)— *базовый скелет* [4]. Базовый скелет может быть получен путем итерационной стрижки тупиковых ветвей скелета [2] (рис.2а). Он обладает свойствами (1), (3) и (4) модели.

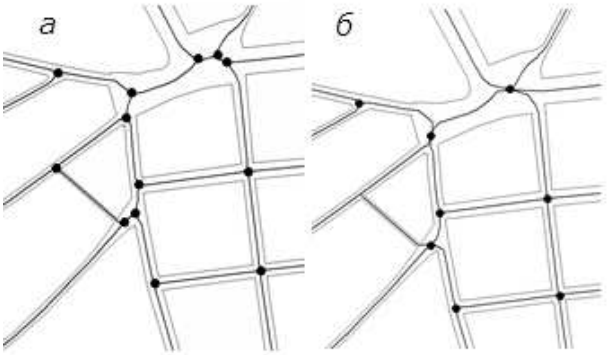


Рисунок 2: Базовый скелет фрагмента УДС (а) и модель фрагмента УДС (б). Точками отмечены вершины графа степени более двух.

Свойство (2) связано с нестрого определенным понятием «адекватное сюръективное отображение». Это означает, что модель должна быть таковой, чтобы разветвления скелета соответствовали разветвлениям дорог, т.е. перекресткам соответствовали вершины скелета степени более двух. Для этого необходимо произвести некоторые операции над базовым скелетом, приводящие его к соответствующему виду. Эти операции тесно связаны с классификацией перекрестков на основе скелетной модели УДС, о чем будет подробно рассказано в следующем разделе.

3. КЛАССИФИКАЦИЯ ПЕРЕКРЕСТКОВ НА ОСНОВЕ СКЕЛЕТА

Разделим все перекрестки на две большие группы — одноуровневые и многоуровневые (развязки). В группе одноуровневых перекрестков (именно они рассматриваются в одноуровневой модели) выделим три подгруппы: простые, сложные и комбинированные. К простым перекресткам относятся: Т-образный перекресток (рис.3а вверху), крестообразный перекресток (рис.3а внизу), площадь (рис.3б). К сложным перекресткам относятся: сложный Т-образный (рис.3в), сложный крестообразный (рис.3г) и сложная площадь (рис.3д). Среди комбинированных перекрестков будем по аналогии различать комбинированные Т-образный (рис.3е), комбинированный крестообразный (рис.3ж), комбинированную площадь (рис.3з).

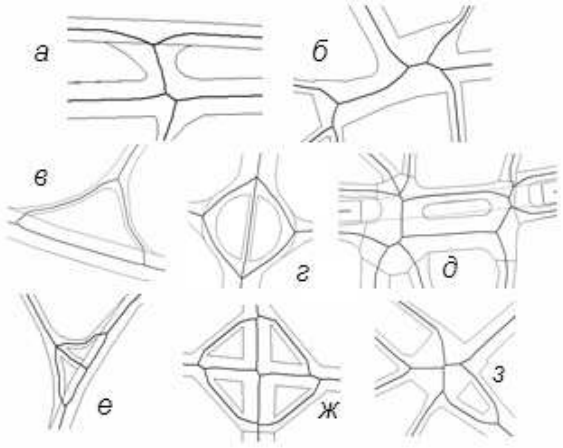


Рисунок 3: Типы одноуровневых перекрестков.

Будем строить сюръективное отображение множества вершин модели на множество перекрестков УДС. Введем три определения.

Будем говорить, что *перекресток H_P является образом вершины P модели УДС*, если H_P соответствует связный подграф модели УДС, содержащий P и не содержащий других вершин степени более двух.

Будем говорить, что *перекресток H_C является образом цикла C модели УДС*, если H_C соответствует связный подграф модели УДС, содержащий C и не содержащий вершин степени более двух, не принадлежащих C, а также не содержащий других циклов, за исключением, быть может, внутри C.

Будем говорить, что *перекресток H_{P_C} является образом «вершины с циклами» P_C модели УДС*, если H_{P_C} соответствует связный подграф модели УДС, содержащий P_C, входящую в состав одного или нескольких циклов $C_1, C_2, \ldots, C_N$, и не содержащий вершин степени более двух, не принадлежащих $C_1, C_2, \ldots, C_N$.

После объединения близких вершин базового скелета (рис.4, рис.2б), мы получим модель, в которой:

- простые перекрестки являются образами вершин модели степени более двух, причем Т-образный перекресток является образом вершины степени 3, крестообразный — вершины степени 4, площадь — вершиной степени 5 и более;
- сложные перекрестки являются образами циклов модели, содержащих по крайней мере две вершины степени более 2, причем сложный Т-образный перекресток является образом цикла, содержащего три вершины степени более 2, и остальные вершины, если они есть, степени 2; сложный крестообразный перекресток — четырьмя вершинами степени более 2; площадь — пятью или более вершинами степени 3, либо содержащим по крайней мере одну вершину степени более 3;
- комбинированные перекрестки являются образами «вершин с циклами» степени более 3, причем комбинированный Т-образный перекресток является образом «верши-

ны с циклами» степени 3, комбинированный крестообразный — «вершиной с циклами» степени 4, комбинированная площадь — «вершиной с циклами» степени 5 или более.

Таким образом, мы определили адекватную сюръекцию множества вершин графа на множество перекрестков (свойство (2)). Модель УДС, к множеству вершин которой применима данная сюръекция, является результатом операции объединения близких вершин базового скелета. В модели можно выделить несколько типов неизоморфных подграфов — *шаблонов перекрестков*. Каждый шаблон характеризуется степенью порождающей вершины и (или) наличием и числом циклов.

Понятие адекватности сюръекции имеет тот смысл, что мы определяем не произвольное отображение одного множества на другое, а закладываем в него интуитивно-понятное соответствие топологии УДС и графа (например, простому перекрестку должна соответствовать одна вершина скелета степени более 2, причем степень соответствует количеству пересекающихся проезжих частей).

Сюръекцию можно определить и по-другому. В ряде случаев удобно использовать модель, в которой детализация организации движения на перекрестке не важна, а важен лишь сам факт наличия перекрестка. В этом случае должно существовать взаимно-однозначное соответствие между множеством перекрестков и множеством вершин скелета степени более двух. Такую модель УДС назовем *макромоделью*.

В следующем разделе подробно опишем все операции над базовым скелетом, а именно: *объединение близких вершин*, *замена цикла вершиной*, *изъятие цикла*,— приводящие к модели и макромодели УДС.

4. ОПЕРАЦИИ НАД БАЗОВЫМ СКЕЛЕТОМ, ПРИВОДЯЩИЕ К МОДЕЛИ И МАКРОМОДЕЛИ УДС

Объединение близких вершин. Участки УДС, соответствующие перегонам между перекрестками имеют протяженность б*о*льшую, чем ширину. В области перекрестков же, напротив, ширина УДС сравнима с длиной. На основе этого сформулируем понятие топологической близости вершин для решаемой задачи: две вершины скелета степени более 2 являются *топологически близкими* в задаче построения модели УДС, если максимальные пустые круги скелета с центрами в этих вершинах пересекаются (рис.4).

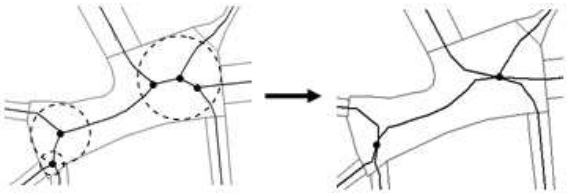


Рисунок 4: Объединение близких вершин.

Такое понятие близости вершин, основанное на топологии скелета, формирует *задачу кластеризации вершин скелета на основе радиальной функции*.

Алгоритм объединения вершин скелета можно найти в [1].

Из-за наличия «дыр» в многоугольной фигуре, описывающей УДС, в модели присутствуют циклы, являющиеся прообразами сложных перекрестков, и содержащие «вершины с циклами», являющиеся прообразами комбинированных перекрестков. Для построения макромодели УДС, необходимо произвести операции замены цикла вершиной и удаления циклов.

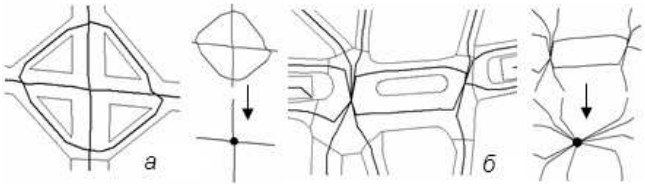


Рисунок 5: Изъятие циклов (а) и замена цикла вершиной (б).

Замена цикла вершиной. Пусть H_C — одноименный сложный перекресток – образ цикла C. Заменой цикла на вершину P_C будем считать удалением всех вершин из базового скелета, входящих в C, введение в полученный граф новой вершины в геометрическом центре цикла и соединение ее с соответствующими висячими ребрами графа (рис.5б). Новая вершина макромодели в данном случае соответствует прообразу простого перекрестка обычной модели УДС того же типа, что и сложный.

Изъятие циклов. Пусть H_{P_C} — комбинированный перекресток – прообраз «вершины с циклами» P_C. Изъятием цикла будем считать удаление всех простых цепочек , не содержащих P_C (рис.5а). Вершина P_C полученной макромодели после такого преобразования соответствует прообразу простого перекрестка обычной модели УДС того же типа, что и комбинированный.

5. ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ

Вычислительный эксперимент построения модели УДС был проведен с помощью ГИС-компонента Mappl [10]. Программный модуль осуществлял автоматическое построение одноуровневой модели фрагментов УДС и выделение перекрестков УДС на основе модели. Для эксперимента были отобраны некоторые сложные одноуровневые фрагменты УДС г. Москвы. На рис.6 изображена построенная модель для участка УДС «Серпуховская площадь и прилегающие улицы» (без тоннеля), а на рис.7 — выделенные перекрестки.

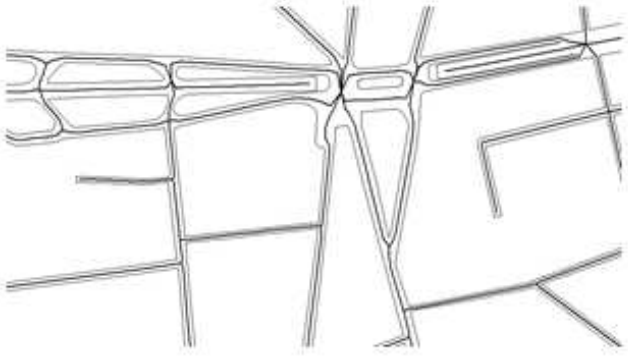


Рисунок 6: Вычислительный эксперимент. Одноуровневая модель фрагмента УДС «Серпуховская площадь и прилегающие улицы» (без тоннеля).

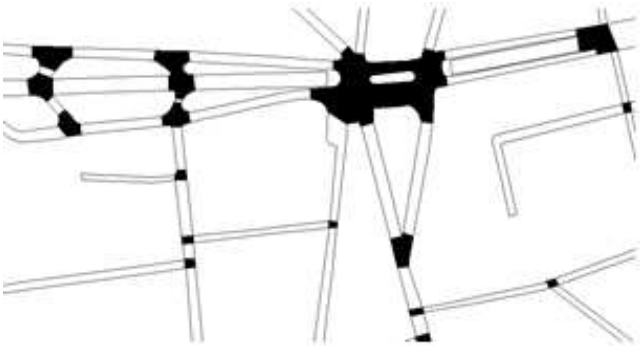


Рисунок 7: Вычислительный эксперимент. Перекрестки, выделенные во фрагменте УДС «Серпуховская площадь и прилегающие улицы» (без тоннеля).

6. ЗАКЛЮЧЕНИЕ

В настоящей работе описана одноуровневая модель дорожной сети на основе скелета многоугольной фигуры. Дана классификация одноуровневых перекрестков на основе топологии модели. Рассмотрены операции над базовым скелетом, приводящие к появлению шаблонов перекрестков в модели и макромодели УДС. Каждый шаблон перекрестка является подграфом модели определенной структуры, причем все шаблоны попарно неизоморфны. Также авторами предложен критерий топологической близости вершин скелета, хорошо применимый к рассматриваемой задаче.

Модель УДС позволяет проводить более детальную классификацию перекрестков и среди изоморфных шаблонов. Так, на (рис.8) слева изображен перекресток с круговым движением, а справа — сложный Т-образный перекресток. Перекрестки имеют изоморфные шаблоны типа «сложный Т-образный перекресток», однако разную геометрическую конфигурацию циклов. Эта задача требует более тонкого анализа геометрии скелета и является предметом будущих исследований.

Также в дальнейшем планируется анализ топологии многоуровневых перекрестков (развязок), а также определение и построения многоуровневой модели УДС.

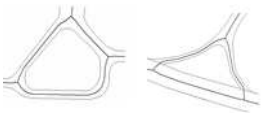


Рисунок 8: Перекресток с круговым движением и сложный Т-образный перекресток: сравнение линейных моделей.

БЛАГОДАРНОСТИ

Работа выполнена при финансовой поддержке РФФИ, проект № 08-01-00670.

АВТОРЫ

Иван Мехедов — аспирант ВЦ РАН.
email: mehedov@mail.ru.

Андрей Козлов — руководитель ООО «Лаборатория транспортных систем».
email: kav57@inbox.ru.

7. СПИСОК ЛИТЕРАТУРЫ

[1] Домахина Л. Г., Охлопков А. Д. Изоморфные скелеты растровых изображений. Труды 18-й международной конференции Графикон-2008, Москва.

[2] Местецкий Л. М. Непрерывная морфология бинарных изображений: фигуры, скелеты, циркуляры. – М.: ФИЗМАТЛИТ, 2009.

[3] Местецкий Л. М. Скелетизация многосвязной многоугольной фигуры на основе дерева смежности ее границы. Сибирский журнал вычислительной математики. – 2006. - Т.9, № 3. - С. 299-314.

[4] Местецкий Л. М., Рейер И. А. Непрерывное скелетное представление изображения с контролируемой точностью. Труды 13-й международной конференции Графикон-2003, Москва.

[5] Aichholzer O., Aurenhammer F. Straight Skeletons for General Polygonal Figures in the Plane. Lecture Notes in Computer Science. Vol. 1090. Springer-Verlag (1996), 117-126.

[6] Haunert J.-H., M. Sester M. Area Collapse and Road Centerlines based on Straight Skeletons. Geoinformatica, Vol.12 є2 (2008), 169-191.

[7] Haunert J.-H., Sester M.. Using the Straight Skeleton for Generalization in Multiple Representation Environment. ICA Workshop on Generalization and Multiple Representation, 2004.

[8] Roberts S. A., Brent Hall G., Boots B. Street Centerlines Пeneration With an Approximated Area Voronoi Diagram. In Fisher P. F. (Ed.), Developments in spatial data handling: 11th International Symposium on Spatial Data Handling. Springer-Verlag (2004), 435-446.

[9] www.egko.ru

[10] www.mappl.ru

ABSTRACT

An approach to the construction of vector map road network model is considered. The model based on a skeleton of polygonal multiconnected domain is a connected graph topologically identical to road network configuration. Based on graph topology, the classification of crossroads is shown. The algorithm of single-level model (all pavements cross at one level) is given.

Keywords: *road network, skeleton, road centerline, crossroad classification.*

Высокочастотный кэш излучения

Константин Востряков
Институт Прикладной Математики им. М.В.Келдыша
vostryakov@gin.keldysh.ru

Аннотация

Разработан новый алгоритм кэширования падающего излучения на основе репроекции. Экстраполяция падающего излучения позволяет существенно сократить количество трассировок луча и вычисления функции отражения. В отличии от других алгоритмов кэширования излучения, предлагаемый алгоритм может работать с высокочастотными функциями отражения и освещением. Алгоритм может быть использован для расчета финального сбора в методах фотонных карт и излучательности. Предлагаемый метод быстрее, чем Монте-Карло трассировка с учетом функции отражения, при этом он конкурирует с ним по точности расчета.

1. ВВЕДЕНИЕ

Постоянно возрастающие требования к сложности сцен вынуждают разрабатывать новые подходы для расчета освещения. Визуальное прототипирование новых продуктов промышленности находит все большее применение и требует не только фотореалистичного, но и физически корректного расчета освещения. Ядром любого алгоритма расчета глобальной освещенности является вычисление интеграла освещения по всем направлениям. Для получения физически корректного решения традиционно применяются методы обратной Монте-Карло трассировки лучей. К сожалению эти методы слишком медленно сходятся. Ради ускорения расчета большинство других алгоритмов накладывают множество ограничений (только ламбертовские поверхности или только плавное освещение). Но большинство реальных объектов имеют сложные ДФО, а падающее излучение неравномерно и имеет большой диапазон. Следовательно возникает потребность в создании алгоритмов, которые позволили бы использовать сцены со сложными ДФО и освещением, но были бы более эффективными, чем традиционные методы обратной Монте-Карло трассировки лучей.

Предлагаемый алгоритм кэширует функцию падающего излучения и переиспользует вычисленные значения в соседних точках поверхности, что позволяет существенно сократить количество трассировок луча и вычислений функции отражения. В отличии от других алгоритмов кэширования излучения, предлагаемый алгоритм может работать с высокочастотными данными. В сравнении с классической реализацией метода Монте-Карло алгоритм дает ускорение в несколько раз при сравнимой точности расчета.

2. ПРЕДЫДУЩИЕ МЕТОДЫ

Для ускорения методов обратной трассировки могут применяться выборка по значимости ДФО. То есть трассировка лучей с плотностью пропорционально функции отражения. Для этого необходимо знать хотя бы приближение подынтегральной функции или уметь обратить функцию распределения, что не всегда возможно выполнить. Адаптивные Монте-Карло алгоритмы могут дать ускорение сходимости, например, Metropolis [1], кэш видимости [2] или многомерная адаптивная трассировка [3], но для их работы необходима регулярная начальная выборка для приближение интегрируемой функции, которая может быть достаточно большой для сложных сцен.

Алгоритмом быстрого глобального освещения [4] генерируется небольшое количество виртуальных точечных источников света, используя метод квази-случайного блуждания. Затем выполняется визуализация сцены с использованием затенения с перемежающейся выборкой виртуальных источников света. Этот алгоритм создает артефакты из-за дискретизации вторичных виртуальных источников и плохо подходит для неламбертовских ДФО и сцен со множеством преград.

Методы, которые основаны на теневых картах [5], имеют аппаратную поддержку в графических процессорах. Затенение вычисляется как доля заграждения геометрии сцены, спроецированной обратно на источник. Методы теневых карт в большинстве случаев пренебрегают физической корректностью. Тем не менее некоторые моменты реализации репроекции предлагаемого алгоритма были почерпнуты из этих методов.

Фотонные карты [6] сохраняют освещение как облако точек столкновения частиц, которые были выпущены из источников, с поверхностями сцены. Однако, оценка освещения, полученная с помощью фотонной карты, имеет низкочастотный шум, и для его устранения используется обратная трассировка по всем направлениям — финальный сбор, который занимает подавляющую часть времени расчета.

Кэш облучения впервые был предложен в работе [7], что позволило в несколько раз ускорить визуализацию, поскольку только в небольшой части точек, видимых через пиксели, облучение вычислялось трассировкой лучей по полусфере, а в остальных точках оно экстраполировалось. Для более лучшего контроля ошибки экстраполяции Krivanek [8] предложил использовать многопроходный адаптивный алгоритм, который, при добавлении новой точки в кэш, сравнивает относительное отклонение облучения в соседних точках, что уменьшает количество артефактов, но приводит к множеству чрезмерных вычислений. Кроме того, в работе [8] было предложено расширить метод Ward'a, чтобы использовать умеренно-зеркальные ДФО, а не только ламбертовские. Это потребовало хранение освещения как функции от направления на полусфере. Для этого использовались сферические гармоники, которые, однако, не применимы к остронаправленным ДФО.

В курсе [6] описывается использование репроекции вместе с кэшем излучения. Точки пересечения лучей и поверхностей сцены, которые были вычислены ранее и записаны в кэш, проецируются на полусферу в новой точке поверхности, в которой необходимо вычислить освещение. Репроекция медленная процедура, кроме того, необходимо использовать z-буфер и что-то делать с пустотами, которые останутся после репроекции. В данной работе сделана попытка устранить недостатки алгоритма репроекции при использовании кэша излучения.

В работе [9] освещение было разделено на ближнее и дальнее. Поскольку дальнее освещение изменяется достаточно плавно, то кэш излучения гораздо более редкий, чем для полного осве-

щения. Для ближнего освещения применялся метод излучательности без учета функции видимости. Ближнее освещение вычисляется очень быстро, но игнорирование видимости дает физически некорректное решение. В предлагаемом алгоритме используется идея разделения на ближнее и дальнее освещение, но в более корректном виде.

3. ВЫСОКОЧАСТОТНЫЙ КЭШ ИЗЛУЧЕНИЯ

За основу нового алгоритма был выбран кэш излучения, поскольку он обладает хорошей масштабируемостью по сложности сцен и точности моделирования. К сожалению он, как и другие алгоритмы, имеет недостатки, которые необходимо устранить. Во-первых, трудно обеспечить заданный уровень ошибки. На практике это преодолевается гораздо большим числом точек в кэше, чем это могло бы быть, и ручным подбором параметров. Один из главных недостатков предыдущего алгоритма кэширования излучения — чрезмерная плотность кэша в зонах, где освещение меняется быстро или ДФО остронаправленная, необходимость многопроходности для точного контроля ошибки. Во-вторых, для того чтобы можно было использовать высокочастотное освещение и ДФО, а также корректно обрабатывать функцию видимости, для базиса нельзя использовать сферические гармоники. Они не могут точно представлять разрывную функцию видимости. В-третьих, использование градиентов для высокочастотного кэша излучения нецелесообразно. Для более точной экстраполяции была выбрана репроекция, которая является более медленной, но позволяет более корректно обрабатывать видимость.

3.1 АЛГОРИТМ ВЕРХНЕГО УРОВНЯ

Предлагается новый алгоритм, основанный на кэше излучения и репроекции. Вначале через все пиксели экрана трассируются лучи. Таким образом находим ближайшие точки пересечения со сценой и сохраняем их в массив. Кроме самой точки, также вычисляются и сохраняются нормали к поверхности и указатели на ДФО. Трассировкой лучей вычисляем первичное освещение от точечных и параллельных источников света. В конце проходим по массиву пикселей и запрашиваем кэш излучения, если в кэше находится подходящая точка для экстраполяции, то экстраполируем (осуществляем репроекцию и заполняем бреши), если нет, то трассируем лучи по полусфере и создаем запись в кэше (сохраняем текстуру падающего излучения и информацию для репроекции). Для удобства хранения падающего излучения в виде квадратной текстуры было выбрано отображение из квадрата на полусферу [10].

3.2 СОЗДАНИЕ ЗАПИСИ КЭША

На вход процедуры, которая вычисляет освещение, подаются координаты точки поверхности и нормаль к поверхности. На выходе имеем текстуру падающего излучения и массив п-элементов, после чего они могут быть записаны в кэш. П-элемент — это сокращение от поверхностный элемент (аналог термина surfel в английском языке, сокращение от surface element). Под этим термином понимается то, что в точке на поверхности имеется некоторая плоская дифференциальная площадка, параллельная поверхности в данной точке.

Вначале трассируем $n \times n$ лучей по полусфере (типичное значение 32×32 или 64×64), находим точки столкновения с поверхностями сцены. Далее вычисляем падающие излучение, которое идет вдоль лучей, сохраняем в текстуру. Вычисленные точки разделяем на слои по глубине. Каждая точка становится п-элементом. Сортировка выполняется по K слоям, что требует $K \cdot O(n \times n)$ времени. Слои выбираются по логарифмической шкале, поскольку ближние п-элементы нужно отсортировать точнее. Внутри слоя объединяем соседние п-элементы имеющие близкую нормаль. Для полученных п-элементов находим угловые точки в uv пространстве (двумерном пространстве полусферы). В конце сохраняем все п-элементы и текстуру падающего излучения в кэш.

3.3 РЕПРОЕКЦИЯ

На вход алгоритма репроекции и заполнения брешей (рисунок 1) подаются координаты точки на поверхности, ее нормаль, ДФО и направление взгляда. А также ближайшая запись из кэша, в которую входит: текстура падающего излучения и массив п-элементов. На выходе имеем исходящее излучение от точки поверхности вдоль направления взгляда. Репроекция осуществляется слой за слоем, начиная от ближнего слоя и заканчивая самым дальним. При этом полноценного Z-буфера не нужно, можно обойтись лишь маской заполнения. Репроецируем все прямоугольные п-элементы в новую точку поверхности, для каждого из них получаем в uv пространстве четыре угловые точки в новой системе координат. При этом по четырем точкам строим прямоугольник, координаты угловых точек которого — это максимальные uv координаты четырех репроецированных точек. Этим мы получаем приближенную проекцию п-элемента. Для п-элемента создаются маски п-элемента и маски неизвестности. Маска п-элемента — это битовая маска, где 1 расположены в тех местах uv пространства, куда п-элемент был спроецирован. Далее следует создание битовой маски следа движения проекции п-элемента. Эта маска вычисляется как объемлющий прямоугольник uv п-элемента в точке из кэша и новой проекции п-элемента. Маска следа позволяет определить область uv пространства, где открылась неизвестная зона (или другими словами зона тени точки из кэша). Маска неизвестности получается как маска следа минус маска заполнения. Маска неизвестности запрещает заполнение более дальних слоев в области неизвестности. Объект, который может находится в такой зоне, может перекрыть фон, видимый из новой точки. Маской неизвестности мы препятствуем тому, что перекрытый фон будет учтен неправильно.

Далее обходим с некоторым шагом k маску заполнения и трассируем лучи там, где есть бреши. На один луч приходится несколько пикселей маски. Добавляем излучение, полученное трассировкой лучей, с весом равным числу незаполненных пикселей маски деленное на число пикселей k^2.

Если после репроекции осталось много брешей, то можно использовать другие соседние записи из кэша. Для этого надо обнулить маску неизвестности, а маску заполнения оставить неизменной. Новый алгоритм корректно поддерживает экстраполяцию освещения от нескольких записей кэша с учетом закрытия одних объектов другими, что предыдущие методы не могли обеспечить.

3.4 РЕАЛИЗАЦИЯ РЕЗУЛЬТАТЫ

Трассировка лучей и вычисление ДФО выполнялись с использованием SSE инструкций [4]. Четыре луча трассировались одновременно. В качестве ускоряющей структуры использовалось kdtree. Репроекция осуществлялась сразу для четырех точках одновременно, также используя SSE2 инструкции. Маски заполнения и неизвестности являются битовыми матрицами 128×128 бит. Любая строка битовой матрицы — это 128-битная переменная. Битовые операции выполняются сразу над всей строкой длиной 128 бит, используя SSE2 инструкции. Перевод п-элемента из одной системы координат в другую может быть выполнен битовыми сдвигами. Репроеция очень похожа на растеризацию. Падающие излучение от п-элемента используется как текстура, а маска п-элемента как альфа-канал

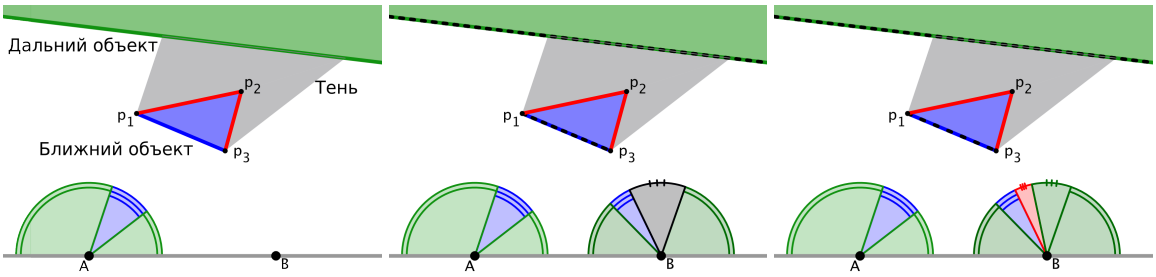


Рисунок 1: Алгоритм репроекциии. а) Падающие излучение в точке A, экстраполируется в точку B б) В точке B освещение рассчитывается репроекцией значений падающего излучения, вычисленных в точке A и сохраненных в кэше. 4 значения излучения репроецируются одновременно используя SSE инструкции, учитывая перекрытие и затенения. в) После репроеции бреши и зона тени заполняются трассировкой лучей.

текстуры.

Рисунок 2: Сравнение качества изображений для выборки пропорционально ДФО 64×64 лучей (первая строка), алгоритма репроекции (вторая строка) и алгоритма репроекции с последующей выборкой по значимости еще 64×64 лучей (третья строка)

Для тестирования метода было выбрано изображение с большим динамическим диапазоном (БДД), которое задавало освещение на сфере. БДД освещения требует большой выборки для интегрирования. За образец для сравнения был выбран метод интегрирования освещенности Монте-Карло с генерацией 64×64 лучей с плотностью пропорционально ДФО, поскольку лишь предлагаемый метод (среди всех методов кэширования) конкурирует с ним по качеству на таких сценах. Метод Монте-Карло сравнивается с предложенным методом репроекции падающего излучения, где кэшируется также 64×64 лучей. Кроме того, после репроекции возможно построить кумулятивную кусочно-постоянную функцию распределения, используя которую можно трассировать лучи более значимым образом и получить несмещенную оценку. Репроекция с последующей выборкой по значимости еще 64×64 лучей также была протестирована.

Все тесты были выполнены для разрешения экрана 512×512 пикселей, один луч на пиксель. Тесты запускались на компьютере: Intel Core2 Quad Q9300 2.5 GHz, используя все 4 ядра процессора. Во всех случаях для уменьшения дисперсии использовалась стратификация базовых случайных величин, которые равномерно распределены в интервале (0, 1). Для Монте-Карло трассировки пропорционально ДФО и при выборке по значимости SSE2 инструкции не использовались.

На рисунке 2 показано сравнение изображений при использовании всех трех методов. Классический контрольный метод Монте-Карло дает шум, который наиболее заметен на сцене shadow, она имеет наиболее остронаправленную зеркальную функцию отражения. При этом детерминизм метода превращает шум в смещение с повторяющимся шаблоном, что визуально менее приемлемо. Но можно использовать метод репроекции вместе с Монте-Карло трассировкой, что дает несмещенную оценку. При этом точность значительно повышается, а шум в освещение снижается. Качество изображение намного выше простого метода Монте-Карло, в котором лучи трассируется с плотностью пропорциональной ДФО. Как видно из таблицы 1 метод репроекции в несколько раз быстрее метода метода Монте-Карло с выборкой пропорционально ДФО, а репроекции с последующей выборкой по значимости показывает лишь немногим большее время расчета, чем базовый метод Монте-Карло. Также можно заметить, что при использовании метода репроекции точность расчета сравнима с методом Монте-Карло (в одной из сцен точность алгоритма репроекции оказывается немного лучше, в остальных немного хуже), при этом скорость в несколько раз больше. Если же после репроекции применяется выборка по значимости, то точность расчета значительно превосходит метод Монте-Карло.

Таблица 1: Время визуализации в секундах и через черту относительная ошибка $L_1 = \frac{\int |L - <L>|}{\int L}$. Сравнение выборки пропорционально ДФО 64×64 лучей, алгоритма репроекции 64×64 лучей и алгоритма репроекции с последующей выборкой по значимости еще 64×64 лучей

Сцена	Кол-во треу-в	Монте-Карло ~ ДФО	Репроекция	Репроекция + Монте-Карло
shadow	110	950.9 / 4.7%	44.0 / 3.4%	961.4/ 1.9%
dragon	871k	310.3 / 2.6%	51.7 / 3.2%	362.9/ 1.8%
armadillo	346k	348.8 / 2.5%	58.1 / 3.3%	411.3/ 1.8%

На основании таблицы 1 точность можно приблизительно считать сравнимой с точностью алгоритма Монте-Карло. При этом можно говорить, что смещенная версия алгоритма репроекции дает ускорение приблизительно от 6 до 21.6 раза. Поскольку метод Монте-Карло сходится как $1/sqrt(N)$, где

N — это количество лучей, то 2 кратное увеличение точности означает 4 кратное ускорение расчета. Используя таблицу 1 можно оценить ускорение, которое дает метод репроекции в несмещенной версии: от 1.92 до 6.1 раз (таблица 2).

Таблица 2: Ускорение (раз) в сравнении с контрольным методом Монте-Карло метода репроекции и метода репроекции с последующей выборкой по значимости (несмещенная версия)

Сцена	Кол-во треу-в	Репроекция	Репроекция + М-К (несмещенная)
shadow	110	21.6	6.12
dragon	871k	6.0	2.09
armadillo	346k	6.0	1.92

В отличии от предыдущих методов кэширования излучения, новый алгоритм репроекции точнее контролирует ошибку экстраполяции, что позволило отказаться от многопроходности, разбив изображение на независимые блоки, которые рассчитываться параллельно в разных потоках. Удалось добиться ускорения расчета вторичного освещения в 3.83 раза при использовании 4 потоков.

3.5 ЗАКЛЮЧЕНИЕ

Разработан алгоритм кэширования и нелинейной экстраполяции падающего излучения. Экстраполяция позволяет переиспользовать вычисленные значения освещения в соседних точках поверхности, существенно сократить количество трассировок луча и вычисления функции отражения. В отличии от других алгоритмов кэширования излучения, предлагаемый алгоритм может работать с высокочастотными функциями отражения и освещением. Более точный контроль ошибки экстраполяции позволяет отказаться от многопроходности и иметь локальный кэш для каждого потока, что ведет к эффективной многопоточной реализации. Алгоритм может быть использован для расчета финального сбора в методах фотонных карт и излучательности. В сравнении с методом обратной трассировки Монте-Карло при сравнимой точности расчета алгоритм работает в несколько раз быстрее. А при использовании вместе с Монте-Карло трассировкой дает несмещенное решение, в несколько раз ускоряя сходимость.

БЛАГОДАРНОСТИ

Работа поддержана грантами РФФИ №07-01-00450 и №09-01-00472.

4. СПИСОК ЛИТЕРАТУРЫ

[1] Abhijeet Ghosh and Wolfgang Heidrich, "Correlated visibility sampling for direct illumination," *Vis. Comput.*, vol. 22, no. 9, pp. 693–701, 2006.

[2] Petrik Clarberg and Tomas Akenine-Möller, "Exploiting Visibility Correlation in Direct Illumination," *Computer Graphics Forum (Proceedings of EGSR 2008)*, vol. 27, no. 4, pp. 1125–1136, 2008.

[3] Toshiya Hachisuka, Wojciech Jarosz, Richard Peter Weistroffer, Kevin Dale, Greg Humphreys, Matthias Zwicker, and Henrik Wann Jensen, "Multidimensional adaptive sampling and reconstruction for ray tracing," *ACM Transactions on Graphics (Proc. of SIGGRAPH)*, vol. 27, 08/2008 2008.

[4] Ingo Wald, *Realtime Ray Tracing and Interactive Global Illumination*, Ph.D. thesis, 2004.

[5] Michael Schwarz and Marc Stamminger, "Bitmask soft shadows," *Comput. Graph. Forum*, vol. 26, no. 3, pp. 515–524, 2007.

[6] Per H. Christensen, Per H. Henrik Wann Jensen, Toskiaki Kato, and Frank Suykens, "A practical guide to global illumination using photon mapping," in *SIGGRAPH 2002 Course Notes*. Association for Computing Machinery, Aug. 2002, ACM SIGGRAPH, Course 43.

[7] Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear, "A ray tracing solution for diffuse interreflection," in *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1988, pp. 85–92, ACM.

[8] Jaroslav Křivánek, *Radiance Caching for Global Illumination Computation on Glossy Surfaces*, Ph.d. thesis, Université de Rennes 1 and Czech Technical University in Prague, December 2005.

[9] Okan Arikan, David Forsyth, and James F. O'Brien, "Fast and detailed approximate global illumination by irradiance decomposition," in *ACM SIGGRAPH 2005 Full Conference DVD-ROM*, Aug. 2005.

[10] Peter Shirley and Kenneth Chiu, "A low distortion map between disk and square," *journal of graphics tools*, vol. 2, no. 3, pp. 45–52, 1997.

High-frequency radiance cache

Abstract

The paper is devoted to a new radiance cache algorithm based on reprojection. By sparsely sampling, caching and reprojecting incident radiance, we significantly reduce number of traced rays and shading calculations. In opposite to previous cache methods, our method is able to handle high-frequency BRDF and lightings. The method is intended for final gathering in radiosity or photon mapping algorithms. Our method is faster than a Monte-Carlo BRDF importance sampling and able to compete with it by quality.

Keywords: *radiance cache, ray tracing, global illumination.*

About the author

Konstantin Vostryakov is a PhD student of the Keldysh Institute for Applied Mathematics RAS. vostryakov@gin.keldysh.ru.

Three-dimensional handwriting visualization method and implementing it software system

Dmitry Shub
Department of Computing Machinery and Systems
Moscow State Institute of Radio engineering, Electronics and Automatics (Technical University), Moscow, Russia
demetryshub@gmail.com

Abstract

In this work a data visualization method and the DA 3D+ Handwriting program system, designed for the on-line analysis of inputted by means of modern input devices (pen tablets, graphic digitizers, etc.) handwriting and signature, are presented. Several new solutions, allowing the increase of the handwriting analysis efficiency, are applied.

Keywords: *visualization, handwriting, signature.*

1. INTRODUCTION

Nowadays various means that allow gaining information about individual characteristics and a state of a person currently attract a lot of attention. One of the reasons for this is the wide application of access control means, data protection means, and diagnostic systems. Abovementioned information can be gained by means of the biometric parameters analysis systems. One of these parameters is handwriting or signature. For years the relationship between the handwriting or the signature and the individual characteristics or a state of a person is used in such fields as the criminalistics (the identification of the person) [1, 10], the graphology (the person character detection) [4, 8, 9]. The signature is one of the common and usual ways to identify the person.

Emersion of computers and modern handwriting input devices resulted in the considerable extension of handwriting research possibilities. The new handwriting input devices allow to gain with a high accuracy not only two-dimensional coordinates of handwriting points, but also hard-to-gain with old ways, using figure on a paper: pen tip pressure value, movement direction, movement speed, and also unavailable with old ways pen slope.

Thus, the modern handwriting input devices provide gaining considerable greater amount of handwriting information that allows to increase performance and reliability of the writer identification and to gain more information about writer state.

Let's note that total replacement of paper documents with the electronic, application of automation in various fields inevitably results in the computerization of the inputting and storing not only printed, but also the handwriting that facilitates and makes usage of the handwriting for mentioned purposes more topical.

New possibilities of handwriting input stimulate a development of new handwriting analysis methods [1, 5, 12, 13, 14] and new tools for the analysis. It could be said that existing methods use tools not aimed for the handwriting analysis that reduces their efficiency.

In this paper, the data visualization method and the DA 3D+ Handwriting program system, designed for the analysis of handwriting inputted by means of mentioned modern input devices, are presented. Several new solutions, allowing to increase the handwriting analysis efficiency, are applied in this system. The program system was designed with state-of-the-art technologies and tools.

Novelty and features of the proposed handwriting visualization and the program system:

- The three-dimensional data visualization method with a usage of the time markers, allowing to display pen tip coordinates, pen tip pressure, the movement speed, measurement time, movement direction simultaneously.
- The natural way of navigation through the text curve, zooming and selection the text curve fragments, by means of a usage of two windows (the preview window and the view window), and a virtual video camera.
- The simultaneous analysis of two and more curves of inputted handwriting with overlapping and scaling support.
- The support of inputted handwriting curves visualization:
    - in an orthogonal projection to view not distorted two-dimensional projection of the text like an image of the text written on a paper, and not distorted curves of dependence of pressure on any of pen coordinates;
    - in a perspective projection to gain an impression of the text curve three-dimensionality on a two-dimensional screen that facilitates sizing up of three-dimensional image depth.
- The possibility to work with most of existing handwriting input devices.
- The support of input, storing and analysis of all inputted handwriting parameters (pen tip coordinates, pen tip pressure, the movement speed, measurement time, movement direction and pen slope) and measurement parameters (time and date of the measurement, information about writer, etc.).
- Expansibility, scalability and the support of upgrade with backward compatibility.
- The usage of the authoring "active" data technology for data storing implementation.
- The usage of state-of-the-art technologies and tools during design of the program system.

2. THE DATA VISUALIZATION METHOD

As it was said, for the handwriting analysis, following parameters are used: pen tip coordinates, pen tip pressure, the movement speed, measurement time, movement direction and pen slope. Most of handwriting researcher does not visualize the gained data or uses traditional methods with usage of two-dimensional plots [1, 5, 11, 12, 13, 14]. Thus the text is represented as two-dimensional image or as a set of two-dimensional plots, corresponding to dependencies of each parameter on a time. In the first case it is possible to visualize time points by means of time markers on displayed curve, but it is difficult to visualize pen pressure and slope in each point of the curve. The pressure visualization by means of change of line width, implemented in some drawing programs (for example, PenSigner, Figure 1), is not convenient as it does not allow to display precisely all of 512 or 1024 pressure levels which is provided by pressure sensors of modern input devices. In the second case (a set of two-dimensional plots, corresponding to dependencies of each parameter on a time) it is difficult to visualize a spatial structure of the text. There is the three-dimensional visualization method applied in ForensicXP-4010 system, in which the image of the text, written on a paper, is represented in the form of a three-dimensional surface (Figure 2), however thus measurement times and pen slopes are not displayed.

Figure 1: The pen pressure visualization by means of a line width change in the PenSigner program.

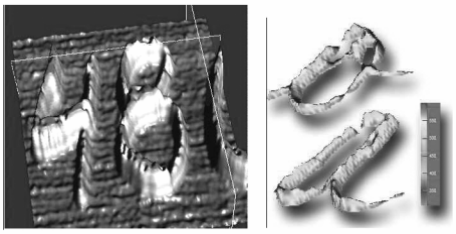


Figure 2: The visualization of a handwriting, written on a paper, in the ForensicXP-4010 system.

Proposed in this work representation of the gained parameters in the form of a three-dimensional curve is more convenient, coordinates of each point of which are corresponded to one of the gained parameters (except measurement time, pen slope and movement direction, which are displayed with the special form of time marker). Implementation of this method is complicated by two-dimensionality of image on the screen of a computer monitor, so the three-dimensional object can be presented on it only indirectly - by means of auxiliary axes, usage of a perspective. Nevertheless this way of representation owing to greater information comprehension is preferable. Implementation of this method is possible by means of some programs which are not specialized for handwriting analysis, for example, computer mathematics and mathematical simulation program systems like Maple, Mathcad, MATLAB, Mathematica. However the handwriting analysis (and especially matching of different samples) by means of these programs is not convenient as they does not support relative moving and scaling of three-dimensional curves directly. Besides in these systems the pen movement directions, pen slope visualization is complicated.

Thus, it is possible to state that the specialized program system for handwriting analysis in which such data visualization method is implemented, not having the abovementioned disadvantages, will increase the analysis efficiency.

The handwriting curve visualized by proposed method is shown on Figure 3. Two curve point coordinates (x, y) correspond to pen tip coordinates, and the third (z) corresponds to pen tip pressure value. The measurement time is represented by means of time markers placed on the gained curve with an equal time period (Figure 4). Thus, the number of markers between readouts corresponds to a time interval between readouts, and the density of markers corresponds to the pen movement speed. Time markers are displayed in the form of pyramids which vertex is directed to the next readouts, i.e. visualize the movement direction.

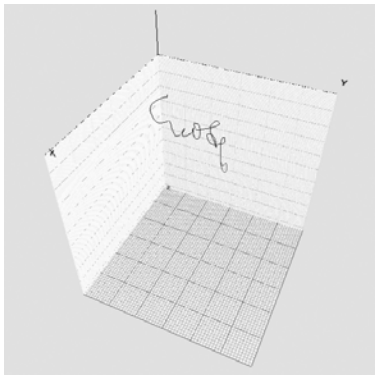


Figure 3: A view of a handwriting curve, the data visualization is implemented by proposed method.

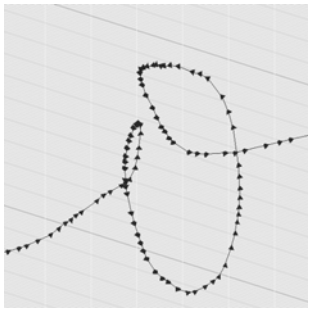


Figure 4: A curve fragment of a handwriting curve.

Measurement of coordinate and pressure values can be performed by means of a millimeter grid (Figure 5) or by means of the special markers, the distance between which is displayed in a corresponding window. The pressure value can be displayed additionally by the line saturation change or the line color transition from the original to inverse

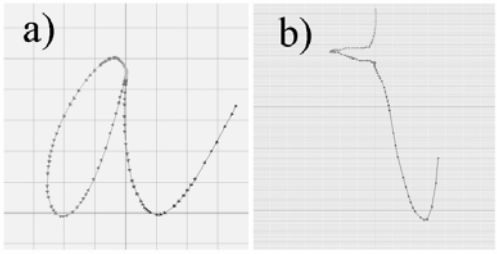


Figure 5: A view of a handwriting curve with a millimeter grid.
a) x and y coordinates, b) pressure and y coordinate

The designed visualization method is realized with a usage of two windows (the preview window and the view window), which are shown simultaneously, and a virtual video camera (further — camera) (Figure 6). In the preview window coordinate axes, a three-dimensional image of the text (x, y – pen coordinates, z - pressure) and the camera connected by a segment with the center of a view area (the camera axis) are displayed; in the view window coordinate axes, the image gained by means of the camera, and millimeter grids in XOY, XOZ, YOZ planes are displayed. The positions of the camera and the center of the view area, the angle between the camera axis and OZ axis, the angle of the camera rotation in the plane XOY, the distance between the camera and the center of the view area (defines image scale in the view window) can be changed. Besides a three-dimensional image of the text can be moved parallel to OX and OY axes and rotated in XOY plane around the text center. The usage of the preview window, in which all considered curves and the camera position being relative to the curve are shown, facilitates the choice of the necessary view of considered curves.

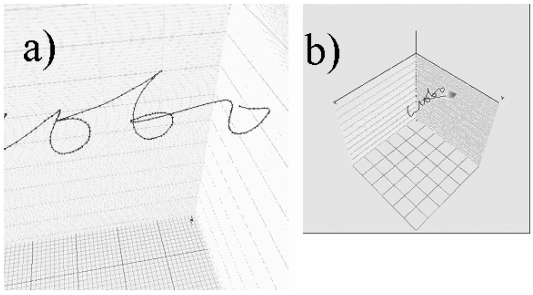


Figure 6: A view of two windows.
a) the view window, b) the preview window

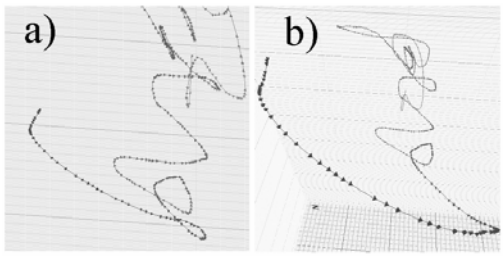


Figure 7: A view of text in the view window.
a) an orthogonal projection, b) a perspective projection

The image, displayed in the view window, can be shown with orthogonal or perspective projection (Figure 7).

The proposed visualization method assumes simultaneous display of two and more different text curves (Figure 8). Thus curves of different texts can be shown with different or same colors that facilitate matching of the curves and visualization of their common characteristics. To support matching of any segments of the curves, independent moving of any curve parallel with OX and OY axes and rotation in XOY plane around center of the curve are provided.

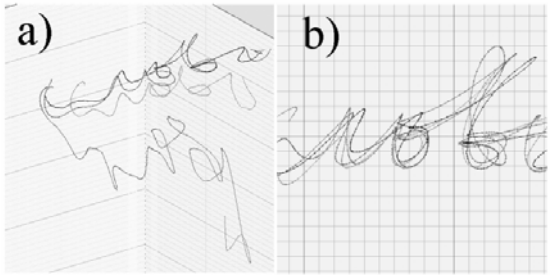


Figure 8: Simultaneous showing different curves of the text.
a) a side view, b) a top view,

3. GAINING UNIFORM READOUTS

Most of input devices (in particular, pen tablets) generates non-uniform readout sequence (in which time interval between adjacent readouts are not equal) while to visualize the time markers the uniform readout sequence are required as the time interval between adjacent time marker must be equal. The readout is a set of several parameters, that is readout corresponds to a vector of handwriting parameters. The values of readouts placed between measurements are gained by means of interpolation. In our case the number of required readouts can attain several thousands.

To solve such problem, a piecewise-linear interpolation and a spline interpolation are applied [3, 6, 7]. In this work, Hermite spline interpolation, providing not only continuous derivative interpolation function but also its (derivative) equality to the given values, is used. As the exact values of curve slope tangents in measurement points, which are necessary for evaluation Hermite spline interpolant, are unknown, it is convenient to use expression Catmull-Rom spline interpolation, being special case of Hermitian spline interpolation. In case of Catmull-Rom spline interpolation it is enough to know readouts between which the interpolated readouts are placed and their previous and next readouts.

4. DA 3D+ HANDWRITING PROGRAM SYSTEM

The complexity of the solution and limitation of allocated for program system development resources demanded to choice the most effective and reliable design technologies and tools. In this case, one of the best solutions was a usage of the Microsoft Visual Studio 2005 IDE, C++ and C# languages, Microsoft .NET technologies, .NET Framework, DirectX9 components.

The decisions, taken during the development of the program system structure, were determined by the chosen handwriting

visualization method and the necessity of support of expansibility, scalability and upgrade with backward compatibility.

To support of mentioned program system quality features following structural decisions has been taken:

1. The system consists of subsystems which supports the basic independent functions.
2. The subsystems interact among each other only by means of a set of interfaces.
3. Interfaces are related with inheritance associations. This and previous decisions allow to realize the subsystem expansibility mechanism which requires to change only the subsystems using new functions.
4. The set of rules, used for creating interfaces, is defined that allows to minimize number of needed for system modification classes.

The program system structure is shown on UML diagram (Figure 9).

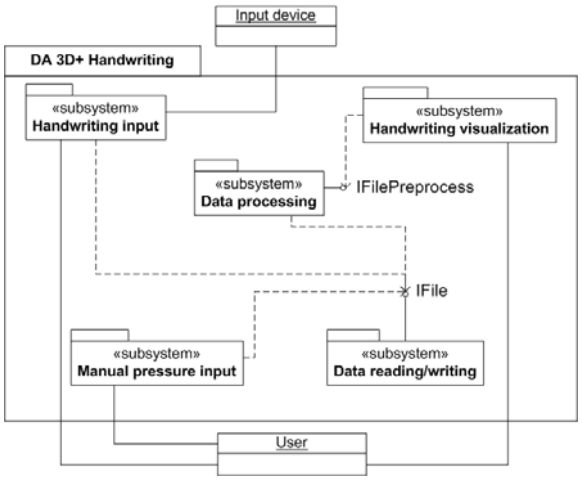


Figure 9: The program system structure.

To support required level of performance during image rendering, two independent sets of graphical resources (the output device objects, vertex buffers, etc.) were used. Thus, to simplify the process of synchronization of images in preview and view windows, a set of the data domain class, performing synchronization and control of graphical resources, was developed. Thus, to each graphical object (the coordinate axes, the millimeter grids, the camera, the three-dimensional handwriting curve, etc.) a special class is corresponded. Design of the graphic interface was performed with usage of standard Windows Forms components, being part of the .NET Framework class library.

5. CONCLUSION

The designed three-dimensional handwriting visualization method and implementing it software system can be used for developing person identification methods, the analysis of person characteristics, the automated identification of the person, applied in electronic document circulation systems, using handwriting or signature analysis, criminalistical expertise of handwriting/signature authenticity.

The proposed program system is tested and used in Institute of Developmental Physiology of Russian Academy of Education for an estimation of children's writing skill development extent, and also for a design of person state analysis, and handwriting identification of person.

6. REFERENCES

[1] Бирюков В.В. Научные и практические основы использования компьютерных технологий для фиксации криминалистически значимой информации: Монография / МВД Украины, Луган. акад. внутр. дел МВД имени 10-летия независимости Украины; – Луганск: РИО ЛАВД, 2002.

[2] Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. – М.: Бином, 1999.

[3] Вержбицкий В.М. Численные методы (математический анализ и обыкновенные дифференциальные уравнения). – М.: Высш.шк., 2001.

[4] Зуев-Инсаров Д. Почерк и личность. Способ определения характера по почерку // Тайна характера. Чтение характера по почерку. - Харьков: Фолио, 1996.

[5] Иванов А.И. Биометрическая идентификация личности по динамике подсознательных движений. – Пенза, Изд-во Пензенского ГУ, 2000.

[6] Каханер Д., Моулер К., Нэш С. Численные методы и программное обеспечение. – М.: Мир, 2001.

[7] Корн Г., Корн Т. Справочник по математике для научных работников и инженеров. – М.: Наука, 1974.

[8] Лисиченко В.К., Барабашина В.М., Варфоломеева Т.В. К вопросу об определении по почерку типологических особенностей личности // Материалы Всесоюзной научной конференции. - М., 1972.

[9] Моргенштерн И. Психографология // Тайна характера. Чтение характера по почерку. - Харьков: Фолио, 1996.

[10] Погибко Ю.Н., Сидельникова Л.В., Хускивадзе Т.Х. Предпосылки разработки научных основ и создания методик криминалистической психолого-почерковедческой экспертизы // Актуальные вопросы судебно-почерковедческой экспертизы: Сб.- М.: ВНИИСЭ, 1985.

[11] Alston J., Taylor J. Handwriting: Theory, Reseach and Practice.- London.: Routlegde, 1987.

[12] Klassen E., Srivastava A., Mio W., Joshi S. (2004), “Analysis of planar shapes using geodesic paths on shape spaces,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 26.

[13] Varga T., Bunke H., ”Offline Handwriting Recognition Using Synthetic Training Data Produced by Means of a Geometrical Distortion Model”, Int. Journal of Pattern Recognition and Artificial Intelligence, Vol. 18, No. 7, 2004.

[14] Vignolo L.A. Modality-specific disorders of writing languages.// Localization in neuropsychology.- N-Y., 1983.

About the author

Dmitry Shub is a Ph.D. student at Moscow State Institute of Radio engineering, Electronics and Automatics (Technical University), Department of Computing Machinery and Systems. His contact email is demetryshub@gmail.com.

Алгоритм рендеринга по методу Монте-Карло петрографических шлифов одноосных кристаллов

Дмитрий Козлов
Новосибирский государственный университет, Новосибирск, Россия
kozlov@oapmg.sscc.ru

Аннотация

Данная работа посвящена задаче рендеринга петрографических шлифов одноосных кристаллов. Описывается специфическая трехмерная сцена – модель петрографического микроскопа. Особенности алгоритма: а) физически корректная модель взаимодействия света с элементами сцены; б) рендеринг поляризованным лучом; в) учет интерференционных эффектов; г) применение интерференционной карты – аналога карты освещенности Арво. В работе приводятся результаты численных экспериментов и обсуждаются полученные изображения.

Ключевые слова: *Фотореалистический рендеринг, кристаллы, петрографические шлифы, двойное лучепреломление, поляризация, дисперсия, интерференция.*

1. ВВЕДЕНИЕ

Кристаллы в общем случае являются анизотропными средами, которые проявляют ряд оптических свойств, несвойственных изотропным средам, например, двойное лучепреломление и плеохроизм [1]. В работах [2-4] рассмотрены алгоритмы рендеринга с учётом поляризации света, а в работе [5] предложен физически корректный алгоритм рендеринга одноосных кристаллических агрегатов.

В обычных сценах интерференция не оказывает существенного влияния, однако, в таких специфичных, как сцены со шлифами, этим эффектом пренебрегать нельзя. В связи с необходимостью учёта интерференции отметим работу [6], где рассматривается реалистический рендеринг при наличии на объектах сцены тонкоплёночных покрытий.

2. ВИЗУАЛИЗАЦИЯ ШЛИФОВ

2.1 Изучение шлифов

Петрографический шлиф – это тонкая пластинка горной породы толщиной 10-30 мкм. К обеим сторонам такой пластинки с каждой стороны приклеиваются две тонких стеклянных пластинки, отшлифованных с внешней от шлифа стороны. Пластинки приклеиваются с помощью канадского бальзама, который имеет оптические характеристики близкие к характеристикам стекла. Размер стандартного шлифа примерно 2x4 см.

При изучении минералов в проходящем свете лучи света проходят через скрещенные поляризаторы (направления пропускания поляризаторов перпендикулярны друг другу), расположенные параллельно шлифу по разные стороны от него [7]. Видимая окраска шлифа определяется возникающими в срезе интерференционными эффектами. Различают два способа изучения шлифов в проходящем свете: в параллельном (ортоскопия) и сходящемся (коноскопия) свете. В первом случае все лучи падают на шлиф перпендикулярно, а во втором случае – конусом с вершиной внутри среза.

2.2 Схема распространения лучей

Рассмотрим схему распространения лучей в случае ортоскопии. Источник белого света (CIE D65) расположен в верхней части сцены (верхняя жирная линия на рис. 1).

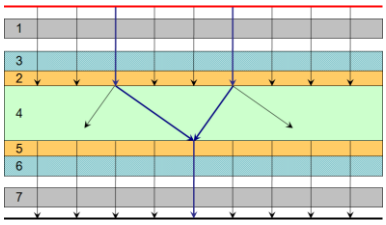


Рис. 1. Схема распространения лучей при изучении шлифа методом ортоскопии

Луч, испущенный источником, обладает естественной поляризацией, т.е. не поляризован.

1. При прохождении поляризатора (1) луч становится линейно поляризованным.
2. Далее луч проходит через слой стекла (3) и слой канадского бальзама (2).
3. При преломлении на верхней границе кристалла (4) луч расщепляется на два луча (обыкновенный и необыкновенный). Пусть при входе в кристалл фаза равна нулю.
4. Распространяясь в кристалле (4) по различным направлениям и с различной скоростью, лучи на выходе приобретают разность фаз. Поскольку плоскости границ кристалла параллельны друг другу, то после преломления на второй границе лучи приобретают направление, которое было до преломления на первой границе, т.е. перпендикулярно кристаллической пластинке.
5. Далее лучи проходят слои канадского бальзама (5) и стекла (6).
6. Проходя через второй поляризатор (направление пропускания поляризатора перпендикулярно первому), лучи приобретают одинаковое направление поляризации.
7. Набор всех лучей представляет собой набор пар с фиксированной разностью фаз, а, следовательно, при выходе из поляризатора они интерферируют между собой, образуя устойчивую интерференционную окраску шлифа.

Изображение формируется на матрице “фотоаппарата”, которая расположена сразу за поляризатором. На рис. 1 матрица отмечена жирной линией снизу.

Для расчета изображений шлифов с помощью метода коноскопии использовалась схема, приведённая на рис. 2. В сцену добавлены две линзы: собирающая (8) и рассеивающая (9). В результате лучи проходят через кристаллическую пластинку

(4), фокусируясь в её центре, что и происходит при исследовании шлифов по методу коноскопии. Сцена существенно отличается от реальной сцены петрографического микроскопа, но она реализует суть метода исследования. В реальности используется значительно более сложная система линз и её более точное моделирование входит в дальнейшие планы нашей работы.

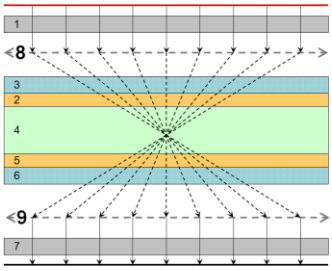


Рис. 2. Схема распространения лучей при изучении шлифа методом коноскопии. Распространение лучей внутри кристалла не показано

3. АЛГОРИТМ РЕНДЕРИНГА ШЛИФОВ

Для рендеринга шлифов в работе был выбран подход, основанный на прямой лучевой трассировке по методу Монте-Карло.

3.1 Представление поляризации света

Для учёта эффекта поляризации света в компьютерной графике необходимо иметь возможность представлять состояние поляризации луча в удобной для компьютерных расчётов форме. Кроме того, необходимо определить набор правил, позволяющих пересчитывать представление поляризации луча при различных изменениях состояния поляризации луча.

В данной работе используется подход основанных на матрицах когерентности и модификаторах матриц когерентности. Этот подход был ранее использован нами для фотореалистичной визуализации одноосных кристаллов подробнее см. в [5]. В этой работе и в работе [4] описаны все необходимые правила преобразования матриц когерентности при взаимодействии света с полупрозрачными одноосными кристаллическими агрегатами.

3.2 Расчёт интерференции в шлифе

При преломлении в одноосном кристалле образуется один обыкновенный и один необыкновенный луч. Поскольку свет падает перпендикулярно к поверхности шлифа, обыкновенный луч продолжит своё распространение по прямой, совпадающей с направлением распространения падающего луча, а необыкновенный луч отклонится в сторону и будет лежать в плоскости, определяемой перпендикуляром к поверхности шлифа и направлением оптической оси, проведённой через точку падения [7], см. рис. 3.

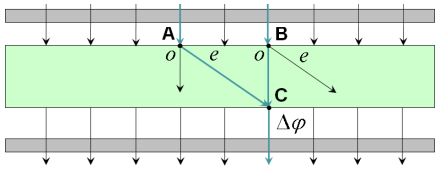


Рис. 3. Схема визуализации шлифа одноосного кристалла. Здесь o и e обозначают соответственно обыкновенный и необыкновенный лучи

Матрица когерентности не содержит полной информации о фазе волны, поэтому для расчёта эффекта интерференции необходимо дополнить описание луча информацией о его фазе, а, именно, информацией о сдвиге фазы относительно какой-либо точки. В данном случае целесообразно хранить информацию о сдвиге фазы относительно точек A и B для обыкновенного и необыкновенного луча соответственно, то есть относительно тех моментов времени, когда лучи зарождаются в кристалле. Таким образом, зададим каждый луч как:

$$R \equiv \{R_0, R_d, CM, \Delta\varphi\}, \quad (1)$$

здесь R_0 – точка начала луча, R_d – единичный вектор распространения луча, CM – матрица когерентности, а $\Delta\varphi$ – сдвиг фазы относительно выбранной точки.

Будем считать, что значения фазы в точках A и B равны 0. Вычислим направления лучей, их состояния поляризации и коэффициенты преломления для обыкновенного и необыкновенного лучей. Тогда на второй границе кристалла фазы для обыкновенного и необыкновенного лучей равны:

$$\varphi_o = \frac{2\pi}{\lambda} n_o d \quad \text{и} \quad \varphi_e = \frac{2\pi}{\lambda} n_e d, \quad (2)$$

где n_o, n_e – коэффициенты преломления обыкновенного и необыкновенного лучей, d – толщина кристаллической пластинки в шлифе, λ – длина волны в вакууме.

После прохождения поляризатора лучи приобретают одну плоскость поляризации и интерферируют. Их общая интенсивность определяется следующим выражением:

$$I = I_o + I_e + 2\sqrt{I_o I_e}\cos(\varphi_o - \varphi_e), \quad (3)$$

где I_o, I_e – интенсивности обыкновенного и необыкновенного лучей, соответственно, после прохождения поляризатора.

3.3 Карта интерференции

Поскольку интерферируют только лучи, распространяющиеся по одной прямой, поляризованные в одной плоскости и имеющие одинаковую длину волны, на матрице изображения необходимо сохранять интенсивности, а также направления и состояния поляризации каждого из попавших в камеру лучей. Для решения этой проблемы нами была разработана *карта интерференции* – аналог карт освещённости [8].

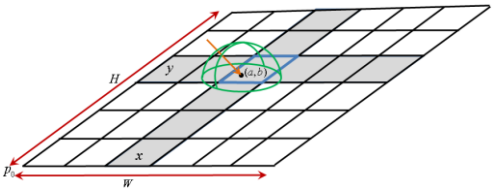


Рис. 4. Схема интерференционной карты

На рис. 4 приведена схема интерференционной карты. В каждой ячейке интерференционной карты хранится информация о лучах, попавших в эту ячейку. Поскольку все лучи, падающие на интерференционную карту, прошли через поляризатор, все они имеют одинаковое состояние поляризации, что позволяет нам не хранить это состояние в интерференционной карте. Для каждого луча хранится информация о его направлении, длине волны, интенсивности и сдвиге фазы. Лучи

с одинаковыми длиной волны, сдвигом фазы и направлением объединяются в один, а их интенсивности суммируются.

Для более экономного хранения лучей в карте, значения длины волны, направления распространения луча и фазового сдвига дискретизируются. Длина волны представляется индексом сампла (из спектра). Фазовый сдвиг определяется в интервале от 0 до 2π с точностью до двух знаков после запятой.

В каждой ячейке интерференционной карты для каждой длины волны (номера сампла) хранится *карта направлений*. Направление луча нормализуется и сохраняется в следующем виде:

$$r_d = (\cos\varphi, \sin\varphi, z). \quad (4)$$

В интерференционной карте сохраняются значения φ и z с точностью 2 знака после запятой.

Для каждого дискретного направления хранится *список сдвигов фаз* и соответствующих им интенсивностей.

Рассмотрим алгоритм сохранения луча в интерференционной карте и алгоритм расчёта спектрального изображения по заполнённой карте.

После попадания луча на карту интерференции производятся следующие действия:

- в ячейке, в которую попал луч, выбирается карта направлений, соответствующая длине волны луча (номеру сампла);
- дискретизируется направление по формуле (4);
- интенсивность пришедшего луча добавляется в список сдвигов фаз соответствующего направления.

При расчёте спектрального изображения интерферируются только лучи, попавшие в одну ячейку интерференционной карты, в одну карту направлений и в одно направление карты направлений. Их общая интенсивность вычисляется по формуле [7]:

$$I = \sum_{m=1}^{n}\sum_{k=1}^{n}\sqrt{I_m I_k}\cos\delta_{mk}, \delta_{mk} = \Delta\varphi_m - \Delta\varphi_k, \quad (5)$$

где I_m, I_k – интенсивности m-ого и k-ого лучей, а φ_m, φ_k – их сдвиги фаз.

Полное значение интенсивности для ячейки интерференционной карты и фиксированного номера сампла вычисляется как сумма интенсивностей по всем направлениям соответствующей карты направлений.

В результате “сбора” изображения с карты интерференции путем вычисления по формуле (5) интенсивности для каждой ячейки и каждой длины волны мы получаем спектральное изображение.

3.4 Рендеринг

Рассмотрим алгоритм рендерига шлифов по методу ортоскопии. В качестве источника будем использовать источник со спектром излучения D65, при этом излучение источника является плоско поляризованным в плоскости, перпендикулярной плоскости источника. Спектр источника лежит в видимом диапазоне волн (380 – 780 нм). Алгоритм:

1. Выбираем случайную точку (x, y) на прямоугольной поверхности источника, расположенного в плоскости XOY, и случайную целочисленную величину χ (номер сампла) из диапазона от 0 до n.

2. Выпускаем луч из этой точки перпендикулярно к поверхности источника в направлении оси аппликат с длиной волны

$$\lambda = 380 + \chi \cdot \frac{400}{n}. \quad (6)$$

Матрица когерентности для луча равна

$$CM_R = \begin{pmatrix} D65(\lambda) & 0 \\ 0 & 0 \end{pmatrix}, \quad (7)$$

здесь $D65(\lambda)$ – спектр источника типа D65.

3. Верхняя граница кристаллической пластинки параллельна плоскости источника света и расположена в плоскости $z = z_0$. Луч пересекает границу в точке (x, y, z_0).

4. Рассчитываем направления распространения преломленных обыкновенного и необыкновенного лучей и направления их векторов рефракции [5,9] и вычисляем координаты их пересечения со второй границей кристаллической пластинки, т.е. с плоскостью:

$$z = z_1 = z_0 + d, \quad (8)$$

где d – толщина кристаллической пластинки. Это будут точки (x, y, z_1) для обыкновенного луча и (x', y', z_1) для необыкновенного. Векторы рефракции обоих лучей совпадут с направлением обыкновенного луча, а их точка пересечения со второй границей пластинки будет (x, y, z_1). Коэффициенты преломления для каждого из лучей равны n_o и n_e, а фазовые сдвиги вычисляются по формуле (2).

5. Вычисляем направления преломленных на второй границе лучей. В нашем случае они будут параллельны испущенному лучу. Их фазовые сдвиги равны $\varphi_1 = \varphi_o$ и $\varphi_2 = \varphi_e$, а матрицы когерентности CM_1 и CM_2.

6. При прохождении через анализатор (нижний поляризатор) направления лучей остаются прежними, а матрицы когерентности лучей пересчитываются по следующей формуле:

$$CM_i^* = M_p CM_i M_p^\dagger, \; i \in \{1, 2\}, \quad (9)$$

где M_p – матрица поворота на угол между осью абсцисс системы координат, связанной с лучом, и осью пропускания анализатора, которая перпендикулярна оси пропускания первого поляризатора, см. [4].

7. Добавляем оба луча в интерференционную карту, расположенную под анализатором, по алгоритму, описанному выше.

8. После генерации источником заданного количества лучей по заполненной карте интерференции рассчитывается спектральное изображение.

4. ЗАКЛЮЧЕНИЕ

На рис. 5 приведены рассчитанные изображения коноскопии шлифа кальцита толщиной 30 мкм. Оригинальные размеры каждого из приведённых изображений составляют 64 на 48

пикселей, именно такого размера использовалась карта интерференции. Размер карты интерференции в памяти составил порядка 900 мегабайт, это соответствует 100 миллионам испущенных лучей для сцены с вертикальным расположением оптической оси и 10 миллионам лучей для сцены с наклонным расположением оси.

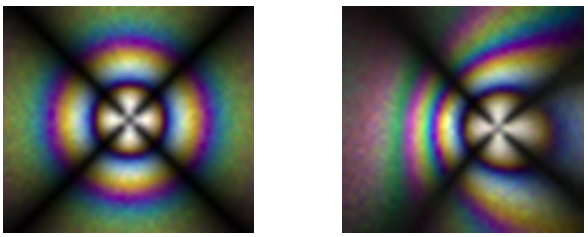


Рис. 5. Рассчитанное изображение коноскопии шлифа кальцита (слева – ось перпендикулярна плоскости среза, справа – несколько наклонена относительно вертикали)

Классический петрографический объект клин представляет собой тонкую вытянутую пластинку некоторого минерала, срезанную под небольшим углом. При просвечивании клина по методу ортоскопии образуется набор интерференционных полос. На рис. 6 приведено изображение клина (угол среза порядка 3 градусов). Оригинальные размеры изображения 200 на 50 пикселей. Излучено 30 миллионов лучей.

Рис. 6. Рассчитанное изображение клина кальцита

В данной точке нашего исследования сравнение рассчитанных изображений с реальными фотографиями оказалось затруднено, тем не менее, оно было проведено с некоторой "натяжкой". На рис. 7 приведено **рассчитанное** изображение коноскопии шлифа кальцита с вертикальной оптической осью, и фотография шлифа берилла (http://jm-derochette.be), **полученная** с помощью петрографического микроскопа. Как видно, алгоритм правильно ухватывает основные особенности интерференционной окраски. К сожалению, нам не удалось найти фотографию шлифа кальцита.

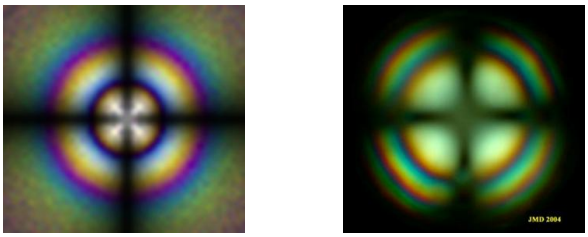


Рис. 7. Сравнение рассчитанного изображения коноскопии шлифа (слева) и фотографии

5. БЛАГОДАРНОСТИ

Я хочу выразить благодарность научному руководителю Дебелову Виктору Алексеевичу за постановку задачи и постоянное внимание к данной работе.

Работа выполнена при поддержке РФФИ по грантам № 06-07-89216а и № 09-07-00237а.

6. БИБЛИОГРАФИЯ

[1] Дебелов В.А., Саттаров М.А. *Модели и алгоритмы для фотореалистической визуализации сцен с кристаллами. //Тр. 15-й международной конференции по компьютерной графике и зрению Графикон-2005, Новосибирск.*

[2] Guy S., Soler C. *Fast and Physically-Based Rendering of Gemstones. // Comp. Graphics (SIGGRAPH '03 Proceedings), ACM Press.*

[3] Дебелов В.А., Козлов Д.С., Саттаров М.А. *Рендеринг монокристаллов с учетом поляризации луча // Proceedings of 6th International Workshop on Virtual Environment on PC Cluster, VEonPC'2006 – Изд-во ИФТИ, Протвино. – С. 55-65.*

[4] Tannenbaum D.C., Tannenbaum P., Wozny M.J. *Polarization and birefringency considerations in rendering. // Comp. Graphics (SIGGRAPH '94 Proceedings), ACM Press, 221-222.*

[5] Козлов Д.С. *Алгоритм фотореалистичного рендеринга одноосных полупрозрачных кристаллических агрегатов. // Программа и тезисы докладов IX Всероссийской конференции молодых ученых по математическому моделированию и информационным технологиям. Кемерово, 2008.* http://www.ict.nsc.ru/ws/YM2008/14308/Kozlov_YM2008.pdf

[6] Yinlong Sun, F. David Fracchia, Thomas W. Calvert, and Mark S. Drew. *Deriving Spectra from Colors and Rendering Light Interference // IEEE Computer Graphics and Applications. – 1999. – No. 4. – P. 1-8*

[7] Борн М., Вольф Э. *Основы Оптики. Москва, "Наука", 1973.*

[8] Arvo J. *Backward Ray Tracing. Developments in Ray Tracing // ACM Siggraph Course Notes 12 – 1986 – P. 259-263.*

[9] Фёдоров Ф.И., Филиппов В.В. *Отражение и преломление света прозрачными кристаллами. Минск, "Наука и техника", 1976.*

Об авторах

Дмитрий Сергеевич Козлов – аспирант Новосибирского государственного университета.

E-mail: kozlov@oapmg.sscc.ru.

Monte Carlo rendering of thin sections of uniaxial crystals

The given report is devoted to the rendering of thin petrographic sections of uniaxial crystals. A specific 3D-scene of petrographic microscope is considered. Features of the algorithm: a) physically correct model of interaction between light and objects; b) rendering with polarization properties; c) interference phenomenon is taken into account; d) using of an interference map similar to Arvo's illumination maps. The paper contains results of numerical experiments and their analysis.

Keywords: *Photorealistic rendering, crystal, thin section, birefringence, polarization, optical dispersion, interference.*

About the authors

Dmitry S. Kozlov is a Ph.D. student at Novosibirsk State University. His contact email is kozlov@oapmg.sscc.ru.

Визуализация данных об экспрессии генов на поверхности коры и гиппокампа мозга мыши

Ольга Сенюкова
Лаборатория Компьютерной Графики и Мультимедиа,
Факультет Вычислительной Математики и Кибернетики,
Московский Государственный Университет, Москва, Россия
olsen222@yandex.ru

Аннотация

Задача исследования экспрессии генов в мозге чрезвычайно важна для понимания развития, функционирования и заболеваний мозга. Одним из этапов анализа экспрессии гена является визуализация. В данной работе визуализируется экспрессия генов в церебральной коре и гиппокампе, которая связана с механизмами памяти.

Исходными данными для визуализации являются трехмерная полигональная модель поверхности мозга (гиппокампа), набор трехмерных точек экспрессии некоторого гена, распределенных по объему коры (гиппокампа), а также атрибуты точек активности – сила экспрессии. Трехмерная визуализация экспрессии гена не всегда дает наглядное представление, поэтому рассматривается альтернативный подход – двумерная визуализация на поверхности анатомических структур. Точки экспрессии гена из указанного слоя (задается глубиной относительно поверхности и шириной) проецируются на поверхность мозга (гиппокампа).

Как правило, поверхностная визуализация применяется для данных функциональной МРТ и ПЭТ. В данной статье предлагается адаптация этих методов для визуализации данных об экспрессии генов.

Также предложен новый ускоренный алгоритм центрального проецирования точек экспрессии гена на поверхность.

Ключевые слова: *Экспрессия гена, Визуализация мозга, Карта мозга, Центральная проекция, Ортогональная проекция.*

1. ВВЕДЕНИЕ

Экспрессией гена называется процесс синтеза белка за счет активности генов в составе ДНК, содержащейся в ядре клетки. В каждой клетке экспрессируется лишь часть генов, которые и определяют ее структуру и функции. В мозге, в отличие от других органов, экспрессируются более 60% всех генов генома. Поэтому исследование активности генов в мозге является чрезвычайно важной задачей для понимания механизмов его работы.

Для исследования экспрессии определенного гена проводится эксперимент над лабораторным животным (обычно лабораторная мышь), в результате которого в его мозге начинает экспрессироваться данный ген. По окончании эксперимента у животного извлекают мозг и замораживают. Главная цель – определить, где и насколько активно экспрессировался исследуемый ген.

Для анализа используются тонкие срезы мозга (порядка 20 микрон). На срезы наносится специфический для данного гена молекулярный зонд. Визуализация зонда выявляет те клетки, где экспрессировался рассматриваемый ген.

После того, как получены срезы, они приводятся к каноническому виду – сопоставляются со специальным атласом мозга мыши (например, [1]). Затем по ним восстанавливается трехмерная модель мозга и трехмерные точки экспрессии гена.

Как правило, используется воксельная визуализация экспрессии генов [2]. Но такой способ визуализации не всегда позволяет качественно оценить данные из-за того, что экспрессия гена – в большом количестве отдельных точек, распределенных по объему структуры (Рисунок 1).

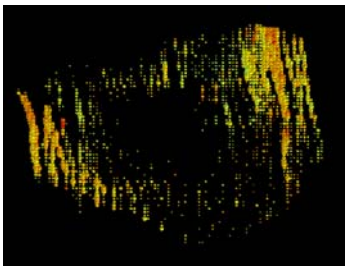


Рисунок 1: Объемная визуализация экспрессии гена.

Поэтому удобно использовать проекции данных экспрессии гена, находящиеся в указанном слое (задается глубиной относительно поверхности и шириной) на поверхность анатомических структур. Рассматривается два метода проецирования – центральное и ортогональное.

Результатом работы должна являться трехмерная модель поверхности мозга (гиппокампа) с нанесенной на нее текстурой – картой активности гена. Текстура поверхности генерируется таким образом, чтобы она была сглаженная, несмотря на дискретность исходных данных, и наглядно отображала картину экспрессии исследуемого гена.

Статья имеет следующую структуру. Часть 2 представляет собой краткий обзор существующих методов визуализации различных экспериментальных данных на поверхности мозга. В части 3 описаны алгоритмы проецирования точек экспрессии гена на поверхность заданной структуры (коры или гиппокампа). Часть 4 посвящена методам раскраски поверхности модели структуры по спроецированным на нее данным. Часть 5 посвящена описанию реализации и тестирования. Приведены примеры результатов визуализации. В заключительной 6 части указаны направления дальнейших исследований.

2. ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ

В [3] описано построение плоских карт для данных авторадиографии мозга крысы. Для каждого среза строится разбиение на сектора с помощью лучей, идущих от центра среза. Для каждого сектора в каждом срезе рассчитывается среднее значение данных. Эти значения последовательно наносятся на двумерную карту. Если каждое значение рассматривать как цвет, то полученную карту можно использовать как текстуру модели мозга.

Авторы [4] предлагают методы поверхностной визуализации данных функциональной МРТ для мозга человека. Объемные экспериментальные данные (массив вокселей со значениями-цветами) визуализируются на полигональной модели поверхности мозга. Каждой вершине присваивается цвет по одному из двух методов. Цвета граней интерполируются по вершинам. Первый метод предполагает нахождение для каждой вершины поверхности мозга вокселя, в который она попадает, и присвоение вершине цвета этого вокселя. При отсутствии для данной вершины соответствующего вокселя используется трилинейная интерполяция ближайших вокселей для определения цвета вершины. Таким образом, этот метод позволяет визуализировать только те данные, которые находятся на поверхности мозга. Второй метод заключается в построении луча, идущего из центра мозга к заданной вершине, и присвоении вершине цвета, равному среднему арифметическому цветов вокселей, через которые проходит этот луч.

В [5] рассматривается проецирование различных видов экспериментальных данных (таких, как функциональная МРТ, позитронно-эмиссионная томография) на поверхность мозга человека. Данные представляются дискретно – в виде набора точек, распределенных по объему. В таком же виде (каждая точка – отдельным кружком) они и визуализируются на поверхности после проецирования. Метод проецирования заключается в поиске для каждой точки ближайшей вершины поверхности – по сути, приближенное ортогональное проецирование.

Таким образом, схема раскраски текстуры при поверхностной визуализации, как правило, соответствует виду начальных данных. Если данные непрерывные – в виде массива вокселей, то и текстура строится сглаженная. Если данные дискретные – в виде набора точек, то они и визуализируются на поверхности по отдельности.

В данной статье предлагается метод, который позволяет генерировать сглаженную текстуру поверхности по дискретным данным экспрессии гена.

3. АЛГОРИТМЫ ПРОЕЦИРОВАНИЯ ДАННЫХ НА ПОВЕРХНОСТЬ

Для построения текстуры точки экспрессии гена сначала проецируются на поверхность одним из двух методов – центральным (для мозга) или ортогональным (для мозга или гиппокампа). Попутно предложен новый метод центрального проецирования, который позволяет значительно ускорить процесс.

3.1 Центральное проецирование

При центральном проецировании данных на поверхность структуры в качестве центра проекции выбирается точка, расположенная в центре исследуемой структуры. Проекцией точки экспрессии гена на поверхность считается пересечение луча, идущего из центра проекции и проходящего через данную точку, с поверхностью (Рисунок 2). Принадлежность точки указанному слою для проецирования будет определяться ее расстоянием до поверхности вдоль этого луча.

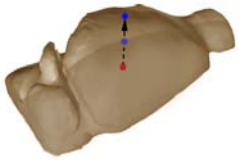


Рисунок 2: Схема центрального проецирования.

Для поиска проекции точки экспрессии гена на поверхность мозга проводится перебор не всех треугольников модели, а только треугольников из некоторой области. Это дает значительный прирост по скорости.

Область поиска нужного треугольника определяется следующим образом. Вводится эллиптическая система координат, начало которой помещается в центр мозга – центр проекции. Таким образом, каждую вершину поверхности мозга и каждую точку экспрессии гена можно задать в этой системе координат тройкой (λ, φ, ρ), где λ и φ - углы, а ρ - расстояние до начала координат. Параметр ρ в данном алгоритме не используется, поэтому каждая точка задается только парой углов (λ, φ) (см. Рисунок 3)

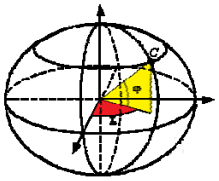


Рисунок 3: Эллиптическая система координат.

Пространство углов разбивается на ячейки – как показано на Рисунке 4.

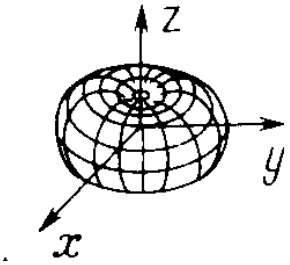


Рисунок 4: Разбиение пространства углов на ячейки.

Искомую область поиска составляют треугольники, вершины которых попадают в ту же ячейку, что и проецируемая точка экспрессии гена. Если в данной ячейке не найден треугольник, на который попадает проекция, то

рассматриваются ячейки, соседние с данной, затем – соседние с ними и т.д., пока не будет найдена проекция.

3.2 Ортогональное проецирование

При ортогональном проецировании для каждой точки экспрессии гена ищется ближайшая точка на поверхности (Рисунок 5).

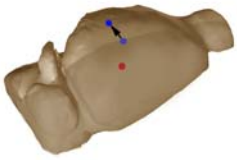


Рисунок 5: Схема ортогонального проецирования.

В данной работе используется приближенная схема ортогонального проецирования, аналогично [5], где для каждой точки исходных данных ищется не ближайшая точка, а ближайшая *вершина* поверхности. Принадлежность точки указанному слою для проецирования будет определяться ее расстоянием до этой вершины.

4. ВИЗУАЛИЗАЦИЯ ТРЕХМЕРНОЙ МОДЕЛИ

4.1 Визуализация экспрессии гена по точкам

Один из способов визуализации данных об экспрессии гена, спроецированных на поверхность мозга (гиппокампа) заключается в том, что каждая точка экспрессии гена изображена отдельным кружком, цвет которого зависит от силы экспрессии гена в данной точке (Рисунок 6(а-б)). Метод визуализации напоминает [5], но в [5] цвет не используется для кодирования значения данных в этой точке.

4.2 Сглаженная визуализация экспрессии гена

Предлагаемый метод сглаженной визуализации данных об экспрессии гена на поверхности мозга (гиппокампа) позволяет построить по спроецированным отдельным точкам сглаженную текстуру – картину экспрессии.

При построении сглаженной текстуры каждой вершине определенным образом присваивается цвет, чтобы затем цвета граней интерполировать по вершинам (как в [4]). Алгоритм присвоения цвета вершине зависит от метода, которым они были спроецированы точки экспрессии гена.

Если точки были спроецированы методом центрального проецирования, то каждой вершине присваивается цвет, соответствующий суммарной экспрессии гена в спроецированных точках, для которых эта вершина – ближайшая.

При приближенном ортогональном проецировании несколько точек могут проецироваться в одну и ту же вершину. Сила их экспрессии суммируется.

Таким образом, текстура поверхности получается сглаженная (Рисунок 6(б),7).

Данный способ визуализации позволяет наглядно выявлять области повышенной экспрессии гена, в то время как способ, описанный в п. 4.1 больше ориентирован на выявление местоположения всех точек экспрессии.

5. РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ

Алгоритмы реализованы в среде MATLAB.

5.1 Замеры производительности

Проведены замеры производительности предложенного алгоритма центрального проецирования. Для модели из 11948 треугольников и для 6304 точек экспрессии гена алгоритм работает 17 секунд. Для сравнения, алгоритм центрального проецирования с полным перебором всех треугольников модели работает около 30 минут. Таким образом, достигается значительный прирост производительности. Это весьма существенно для разработки интерактивного приложения, в котором пользователь может менять различные параметры и необходимо проводить проецирование заново.

Ускоренный алгоритм поиска пересечения луча с полигональной сеткой [7] работает 1 минуту 15 секунд, т.е. медленнее предложенного алгоритма более чем в 4 раза.

5.2 Результаты

В рамках данной работе визуализировались гены, экспрессирующиеся в церебральной коре мозга и гиппокампе – структуре в мозге, отвечающей за формирование памяти.

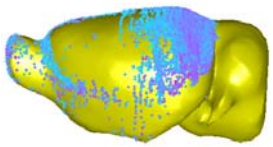


Рисунок 6(а): Визуализация экспрессии гена C-Fos в коре по точкам (метод проецирования - центральное).

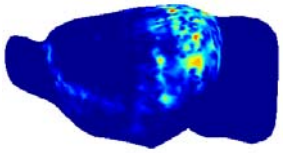


Рисунок 6(б): Сглаженная экспрессии гена C-Fos в коре по точкам (метод проецирования – приближенное ортогональное).

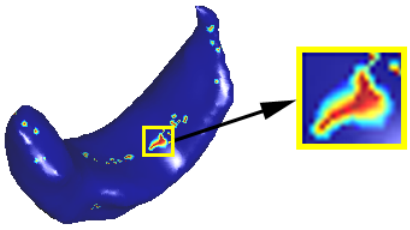


Рисунок 7: Сглаженная визуализация экспрессии гена Crlf1 в гиппокампе.

5.3 Интерактивное приложение

В среде MATLAB также разработано интерактивное приложение для визуализации данных об экспрессии генов на поверхности мозга и гиппокампа. Разработанный прототип обладает следующими возможностями:

- отображение трехмерной модели со спроецированными на нее точками активности гена (модель можно интерактивно вращать)
- выбор рассматриваемой структуры (кора или гиппокамп)
- выбор режима раскраски точек активности гена в зависимости от расстояния до поверхности или силы экспрессии
- задание процента отображаемых точек активности гена
- задание слоя для проецирования с помощью двух параметров – глубины (расстояния до поверхности) и ширины
- отображение исходных точек экспрессии гена (показать или скрыть).

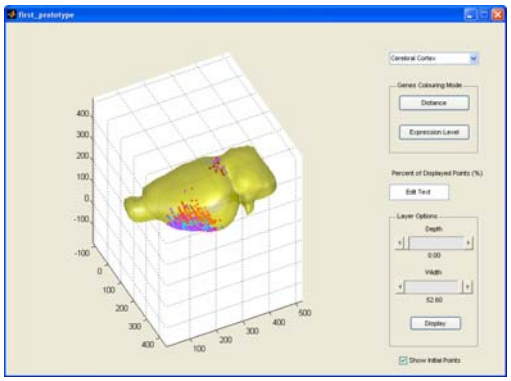


Рисунок 8: Прототип приложения

6. ДАЛЬНЕЙШИЕ ПЛАНЫ

В качестве ближайших планов по данной работе планируется построение плоской развертки структуры (мозга или гиппокампа) со спроецированными на нее данными – карты активности гена. Также планируется разработка методов анализа и сравнения данных. Помимо ортогонального и центрального проецирования будет рассматриваться метод проецирования вдоль колонок в мозге, которые будут распознаваться по данным оптической томографии.

7. БЛАГОДАРНОСТИ

Автор выражает благодарность Антону Конушину и Дмитрию Ветрову за помощь, оказанную во время работы. Автор также благодарит Константина Анохина за ценные консультации.

8. БИБЛИОГРАФИЯ

[1] Allen Reference Atlas. ARA Coronal Atlas Images

http://mouse.brain-map.org/atlas/ARA/Coronal/browser.html

[2] C. Lau, L. Ng, C. Thompson, S. Pathak, L. Kuan, A. Jones, M. Hawrylycz. *Exploration and visualization of gene expression with neuroanatomy in the adult mouse brain.* BMC Bioinformatics, 2008

[3] D.P. Holschneidera, O.U. Scremine, D.R. Chialvog, B.P. Kayd, J.-M. I. Maarekd. *Flattened cortical maps of cerebral function in the rat: A region-of-interest approach to data sampling, analysis and display.* Neuroscience Letters, Vol. 434, Issue 2, pp.179-184, 2008

[4] P. Lincoln. *Surface projection method for visualizing volumetric data.* 2006 http://sigpubs.biostr.washington.edu/archive/00000193

[5] D. C. Van Essen, H. A. Drury, S. Joshi, M. I. Miller. *Functional and structural mapping of human cerebral cortex - Solutions are in the surfaces.* Proceedings of the National Academy of Sciences of the United States of America, Vol. 95, Issue 3, pp. 788-795, 1998

[6] H. Samet. *The Design and Analysis of Spatial Data Structures.* Addison-Wesley, Reading, MA, 1990. ISBN 0-201-50255-0.

[7] K. Eaton. "boundary" class v2.1: a wrapper for surface objects

http://www.mathworks.com/matlabcentral/fileexchange/20637

Об авторе

Ольга Сенюкова – аспирант лаборатории Компьютерной Графики и Мультимедиа факультета Вычислительной Математики и Кибернетики Московского Государственного Университета им. М.В. Ломоносова. Ее электронный адрес - olsen222@yandex.ru.

Visualization of gene expression data on the surface of mouse brain cerebral cortex and hippocampus

Abstract

Exploring of gene expression in brain is very important for understanding of brain development, disease, and function. One of the stages of gene expression analysis is visualization. In this work genes expression in cerebral cortex and hippocampus which is essential for memory function, are visualized.

3D polygonal model of brain (hippocampus) surface, a set of expression points of a particular gene and points' attributes (expression scores) is initial data for visualization. 3D visualization is not always convenient. An alternative method – 2D visualization on the surfaces of anatomical structures – is regarded here. Gene expression points from a user-defined layer (which is set by depth – distance from the surface, and width) are projected onto the surface of the brain (hippocampus).

Usually surface-based visualization is used for fMRI and positron emission tomography. In this paper there is proposed an adaptation of these methods to visualization of gene expression data.

There is also proposed a novel fast algorithm of central projection.

Key words: *Gene expression, Brain visualization, Surface-based visualization, Brain map, Central projection, Orthogonal projection.*

About the author

Olga Senyukova is a PhD student at Moscow State University, Department of Computational Mathematics and Cybernetics, Graphics and Media Lab. Her email is olsen222@yandex.ru.

Исправление перспективных искажений с помощью зеркала

Павел Михайлов
Омский государственный университет им. Ф.М. Достоевского, Омск
cs11mih@mail.ru

Аннотация

Целью данной работы является улучшение существующих систем видеонаблюдения за счет уменьшения перспективных искажений, появляющихся на изображении из-за расположения камеры под углом к объекту наблюдения. Предлагается исправлять возникающие искажения с помощью зеркала особой формы. Получено дифференциальное уравнение для поверхности зеркала, обладающего требуемыми свойствами, уравнение решено численно. Свойство зеркала устранять перспективные искажения проверено моделированием рассматриваемой оптической системы методом трассировки лучей. Обозначены границы применимости полученной оптической системы и ее возможные улучшения.

Ключевые слова: коррекция перспективы, трассировка лучей, зеркало, аберрации оптических систем.

1. ВВЕДЕНИЕ

Перспективные искажения обычно возникают в случае, если плоскость изображения и плоскость предмета не параллельны между собой. Например, при фотографировании высоких объектов с уровня земли (архитектурных сооружений) линии, вертикальные на фотографируемом объекте, на изображении получаются наклонными. На рис.1(слева) представлено расположение камеры над серединой фотографируемого объекта (в случае фотографирования архитектурных сооружений – посередине высоты здания). В этом случае перспективные искажения отсутствуют, черно-белые полосы на изображении одинаковы по ширине.

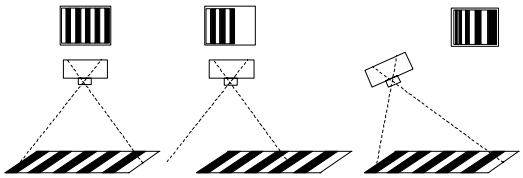


Рис. 1 – взаимное расположение предмета и камеры

Но, как правило, не удается расположить камеру точно над объектом наблюдения. Здания приходится фотографировать с земли, камера наблюдения располагается не над улицей, а, например, на стене одного из зданий (рис. 1, по центру). В этом случае, если плоскости изображения и предмета параллельны, перспективных искажений нет, но не весь предмет помещается в кадре. Поэтому камеру приходится наклонять, чтобы в кадр попал весь предмет (рис. 1, справа). В этом случае черные и белые полосы имеют разную ширину, проявляются перспективные искажения. Чем дальше область предмета расположена от плоскости изображения, тем меньше ее ширина на изображении.

Существуют различные способы исправления перспективных искажений. Программные методы коррекции [6] устраняют видимые искажения (вертикальные на предмете, но сходящиеся на изображении линии после коррекции становятся вертикальными), но при этом расположенные дальше от плоскости изображения области предмета не становятся лучше различимыми.

Для исправления перспективных искажений во время съемки используются специальные шифт-объективы. Оптическая ось объектива проходит не посередине пленки (светочувствительной матрицы) (рис. 2, слева), и при съемке в кадр попадает весь предмет. Существующие шифт-объективы позволяют сдвигать оптическую ось объектива на расстояния до 20 мм, исправляя угол наклона до 25^{o} (имеется в виду угол между перпендикуляром к поверхности предмета и оптической осью камеры). Кроме того, в случае сдвига объектива в сторону от центра пленки свет падает на объектив под большим углом, и, значит, сильнее сказываются оптические недостатки объектива.

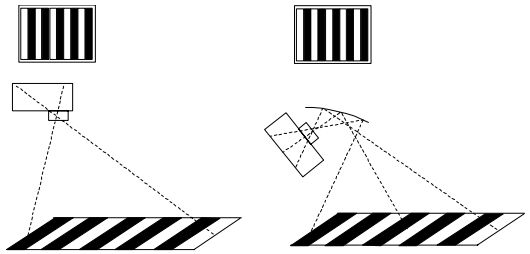


Рис. 2 – исправление перспективных искажений

В данной работе предлагается иной подход. Исправлять перспективные искажения предлагается с помощью зеркала особой формы, расположенного так, как показано на рис.2, справа. Зеркало будет вносить определенные искажения в получаемое изображение, но с его помощью можно исправлять искажения от больших углов наклона, чем с помощью шифт-объектива. Кроме того, для систем видеонаблюдения требуется не столько высокая детализация (здесь и далее под детализацией изображения в какой-либо его точке будем понимать минимальный угловой размер различимых деталей предмета, процедура проверки различимости будет описана ниже) изображения, сколько возможность фиксации и идентификации происходящих с наблюдаемым объектом событий. При этом допустимы некоторые искажения получаемого изображения. Если камера системы наблюдения расположена вдоль стены здания или вдоль забора, то ближние к камере области будут получаться с большей детализацией, а удаленные области – с меньшей детализацией, чем нужно для выполнения поставленных перед такой камерой задач. С помощью зеркала можно ценой вносимых им искажений добиться одинаковой детализацией как удаленных, так и близких областей предмета. В некоторых случаях возникающими искажениями можно будет пренебречь

2. ХАРАКТЕР ПЕРСПЕКТИВНЫХ ИСКАЖЕНИЙ

Рассмотрим, как именно меняется изображение предмета в случае, если камера расположена под углом к плоскости изображения (рис. 3).

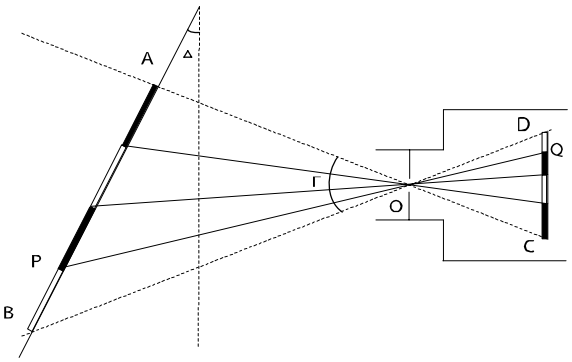


Рис. 3 – формирование изображения предмета, расположенного под углом к плоскости изображения

Камера с углом изображения Γ направлена на расположенный под углом Δ к плоскости изображения предмет AB. На плоскости изображения формируется изображение предмета DC, при этом точка P предмета переходит в точку Q изображения. Рассмотрим функциональную зависимость $q = CQ / DC, q \in [0,1]$ от $p = AP / AB,\ p \in [0,1]$

$$q = \frac{p\cos\left(\Delta + \frac{\Gamma}{2}\right)\cos\left(\frac{\Gamma}{2}\right)}{\cos\left(\frac{\Gamma}{2}\right)\cos\left(\Delta - \frac{\Gamma}{2}\right) - p\sin\Gamma\sin\Delta} \qquad (1)$$

График зависимости p от q для $\Gamma = \pi/4$ и для различных значений $\Delta \in \{0, \pi/8, \pi/6, \pi/4, \pi/3\}$ представлен на рис. 4.

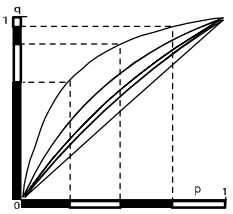


Рис. 4 – график зависимости q от p

На рис. 4 вдоль осей координат показаны предмет (горизонтально) и соответствующее ему искаженное изображение (вертикально) для угла наклона $\Delta = \pi/3$. Зависимость p от q показывает, как растягивается или сжимается изображение в зависимости от угла наклона Δ.

3. ЗЕРКАЛО ДЛЯ КОРРЕКЦИИ ПЕРСПЕКТИВНЫХ ИСКАЖЕНИЙ

Рассмотрим случай, когда предмет находится на значительном расстоянии от камеры (в «бесконечности»). В этом случае можно считать, что интенсивность точки Q на изображении (рис. 5) определяется лучами света, идущими в направлении PO.

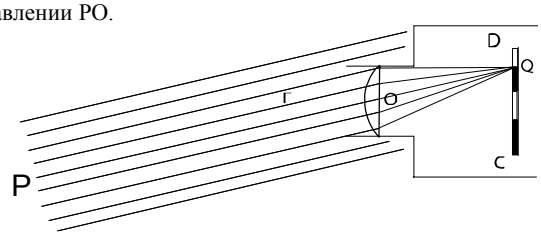


Рис. 5 – предмет расположен на значительном удалении от камеры

Расположим на пути лучей света зеркало M (рис. 6). Если зеркало велико по сравнению с камерой, то интенсивность точки изображения в направлении CD будет определяться узким пучком лучей направления BCO. Необходимо найти зеркало такой формы, чтобы изображение наклоненного предмета получалось бы как от предмета, расположенного без наклона. Пусть $\angle CDE = \varphi, \angle AOB = \xi$, $\gamma(\varphi)$ - угол наклона касательной в точке C

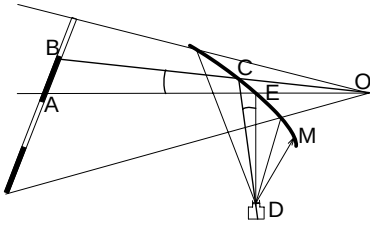


Рис. 6 – схема расположения зеркала между предметом и камерой

Тогда

$$tg(\gamma(\varphi)) = \frac{\sin(\varphi + \xi) - 1}{\cos(\varphi + \xi)} \qquad (2)$$

Зная угол наклона касательной к поверхности зеркала в любой точке, можно записать уравнение поверхности зеркала в полярных координатах с центром в точке D:

$$r'(\varphi) - \frac{r(\varphi)}{tg(\gamma(\varphi))} = 0 \qquad (3)$$

Преобразуя данное уравнение, получаем:

$$\frac{1}{tg(\gamma(\varphi))} = \frac{1 - tg\varphi \cdot tg\xi}{tg\varphi + tg\xi + \frac{\sqrt{tg\xi^2 + 1}}{\cos\varphi}};$$

$$r(\varphi) = e^{-\int \frac{d\varphi}{tg(\gamma(\varphi))}} \cdot r(0); \qquad (4)$$

Это уравнение решалось численно. В случае $\Delta = 0$ решением данного уравнения является прямая (плоское зеркало).

4. ПРОВЕРКА СВОЙСТВ ЗЕРКАЛА МЕТОДОМ ТРАССИРОВКИ ЛУЧЕЙ

Свойство найденной поверхности исправлять перспективные искажения проверялись методом трассировки лучей. Моделирование проводилось в предположениях геометрической оптики [1]. Предмет, расположенный под углом к плоскости изображения состоит из n плоских

ламбертовских [4] источников света (рис. 9), каждый источник света испускает m лучей. Лучи отражаются от зеркала и попадают в камеру. Камера состоит из непроницаемого для лучей корпуса, объектива и «светочувствительной матрицы». Далее лучи преломляются на объективе камеры и попадают на «светочувствительную матрицу», разбитую на k светочувствительных ячеек.

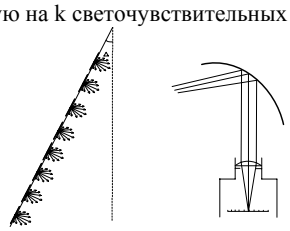


Рис. 9 – взаимное расположение предмета, зеркала и камеры для проверки свойств зеркала методом трассировки лучей

Интенсивность изображения i-го пикселя формируется в зависимости от количества лучей, пересекшихся с рассматриваемой ячейкой номер i. В качестве объектива камеры использовались (рис. 10): камера-обскура (отверстие в непрозрачной пластинке), объектив из одной гиперболической линзы, описанной в [3], а также фотографический объектив И-10 [5].

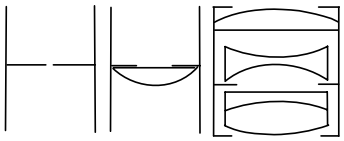


Рис. 10– различные объективы, использовавшиеся для получения изображения методом трассировки лучей

В качестве предмета использовалось одномерное изображение размером 128 пикселей (рис. 11). «светочувствительная матрица» камеры состояла из 256 пикселей. Предмет расположен на значительном удалении от камеры, камера сфокусирована на «бесконечность»

Рис. 11–предмет для проверки свойств зеркала методом трассировки лучей

На рис. 12 показано изображение этого предмета, полученное для случая угла наклона между плоскостью предмета и плоскостью изображения $\Delta = \pi/4$, предмет расположен в «бесконечности», камера без оптических недостатков (рис. 10, слева). В левой части изображения заметна повышенная детализация (при съемке элементы этой области видны под большим углом), справа же заметно понижение детализации – белые полосы в правой части предмета уже не различимы. Цель исправления перспективных искажений состоит в том, чтобы добиться одинаковой детализации по всему изображению.

Рис. 12 – Изображение предмета при наклоне камеры, без зеркала

Расположим на пути лучей зеркало так, как показано на рис. 9 В качестве характеристики относительных размеров зеркала и камеры будем использовать отношение ширины апертурной диафрагмы к расстоянию от диафрагмы до зеркала в виде 1:d.

Уравнение поверхности зеркала получено при допущении, что камера очень мала по сравнению с зеркалом, и на интенсивность каждой точки изображения влияют лучи одного соответствующего направления. Если же размеры камеры (точнее, ширина апертурной диафрагмы) сравнимы с размерами зеркала, то на интенсивность точки изображения будут влиять лучи разных направлений, внося таким образом искажения в изображение предмета. Чем больше размер зеркала по сравнению с камерой, тем искажения меньше. Искажения носят характер «размытия» изображения. За счет оптических свойств зеркала повышается детализация участков предмета, видимых под меньшим углом, и одновременно понижается за счет вносимых зеркалом искажений.

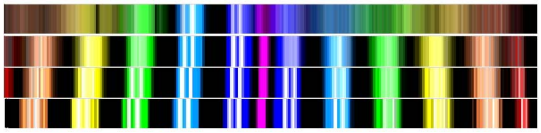


Рис. 13– изображения, полученные с использованием корректирующего зеркала, для разных соотношений размеров зеркала и камеры

На рис. 13 показаны изображения предмета, полученного после расположения зеркала найденной ранее формы между предметом и камерой, для разных относительных размеров зеркала и камеры - 1:6, 1:25, 1:50, 1:100. На рис. 13 внизу (соответствующему большему относительному размеру зеркала) белые полосы по всему предмету различимы, кроме того, заметна одинаковая ширина цветных полос в сравнении с рис.12. При уменьшении относительных размеров зеркала (рис. 13, вверху) заметны вносимые зеркалом искажения – изображение полос более размыто и о повышении детализации в этом случае говорить нельзя.

5. ДЕТАЛИЗАЦИЯ ПОЛУЧАЕМЫХ ИЗОБРАЖЕНИЙ

Для определения детализации получаемого изображения использовались «миры». Предмет, рассматриваемый с помощью оптической системы, состоял из двух светлых полосок определенной ширины, разделенных темной полоской той же ширины. Последовательно рассматривая миры увеличивающегося размера, фиксировалась мира, на изображении которой были различимы две светлые полоски. В качестве критерия различимости использовалась следующая процедура. На изображении выделялись два максимума интенсивности – слева и справа от темной полоски (рис. 14). Две соседние полоски разрешаются, если минимум интенсивности между ними достаточно мал, чтобы его разглядеть. Если отношение интенсивности в максимумах к интенсивности посередине между максимумами было больше заданной константы $\alpha = 0.5$, то полоски считались различимыми.

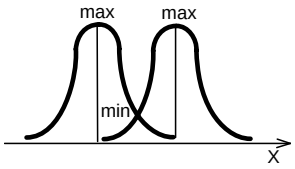


Рис 14 – определение различимости

На рис. 15 приведены графики значений детализации (углов, под которыми видна различима мира минимального размера) в различных частях изображения, полученных для случая предмета без наклона (горизонтальная прямая) и с наклоном $\Delta = \pi/4$ плоскости предмета по отношению к плоскости изображения.

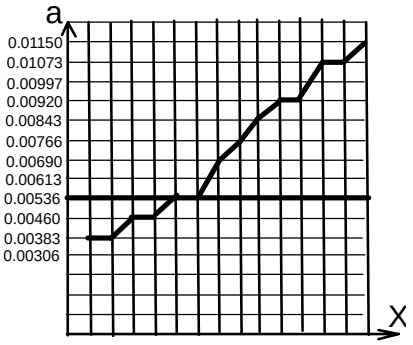


Рис. 15 – детализация изображений без использования корректирующего зеркала

Заметно повышение детализации в левой части изображения и понижения в правой (по сравнению детализацией предмета без наклона). Это обуславливается тем, что детали в левой части изображения видны под большим углом, чем детали правой части.

На рис. 16 приведены графики детализации изображений, полученных с использованием корректирующего зеркала для различных относительных размеров зеркала и диафрагмы. Для небольших относительных размеров зеркала и диафрагмы заметно значительное понижение детализации за счет вносимых зеркалом искажений. С увеличением размеров зеркала детализация растет и после некоторого порога (предельного размера зеркала) «кружок рассеяния» становится меньше чем размер светочувствительной ячейки камеры. После этого дальнейшее увеличение зеркала не будет приводить к повышению детализации.

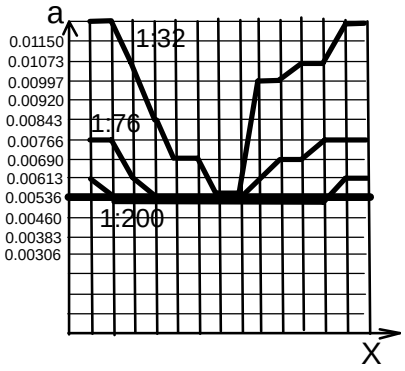


Рисунок 16 – детализация изображений, полученных с использованием корректирующего зеркала.

6. УСТРАНЕНИЕ ИСКАЖЕНИЙ, ВНОСИМЫХ ЗЕРКАЛОМ

Возникает вопрос, можно ли каким-либо образом устранить искажения, вносимые зеркалом в случае, когда относительные размеры зеркала и объектива меньше предельного (т.е. такого порогового значения, после которого увеличение зеркала не будет приводить к увеличению детализации). Пусть функции $S(x)$ и $L(x)$ задают соответственно предмет и изображение. Эти функции связаны уравнением:

$$\int K(x,t)S(t)dt = L(x) \qquad (5)$$

Задача нахождения $S(x)$ (предмета) по $L(x)$ (изображению) является некорректной (некорректно поставленной), т.е. малые ошибки в $L(x)$ и ядре $K(x,t)$ могут приводить к большим ошибкам в решении [2]. В дискретном случае, когда $S(x)$ и $L(x)$ заменяются на векторы, а уравнение (5) – на систему линейных уравнений. Эта система оказывается разрешимой, и по известному вектору изображения (с искажениями, вносимыми зеркалом) удается восстановить предмет (рис. 17). Сверху – изображение с искажениями, снизу – результат восстановления.

Рис. 17 – устранение искажений, вносимых зеркалом

7. ЗАКЛЮЧЕНИЕ

С помощью зеркала и последующей обработки полученного изображения удается устранить перспективные искажения в изображении предмета. Рассмотренная выше оптическая система, состоящая из камеры и корректирующего зеркала, может найти применение в системах видеонаблюдения, где решающим является не качество изображения, а возможность идентифицировать происходящие на наблюдаемом объекте события.

8. ЛИТЕРАТУРА

1. **Бегунов, Б.Н.** Геометрическая оптика / Б.Н. Бегунов, М:.1966.
2. **Верлань А.Ф. ,Сизиков В.С.** Методы решения интегральных уравнений с программами для ЭВМ / А.Ф. Верлань, В.С. Сизиков – Киев «Наукова думка»,1978. 292с.
3. **Декарт, Р.** Рассуждение о методе с приложениями: диоптрика, метеоры, геометрия / Р.Декарт, - М:.1953
4. **Родионов С.А.** Основы оптики. Конспект лекций.– / С.А. Родионов - СПб: СПб ГИТМО (ТУ), 2000.
5. **Яковлев А.Ф.** Каталог "Объективы" / А.Ф. Яковлев - Л., ГОИ, 1970 г.
6. **Hartley, R.I. and Zisserman, A.** Multiple View Geometry in Computer Vision - Cambridge University Press, 2004.

Excluding cascading classifier for face detection

Ivan A. Krestinin, Oleg S. Seredin
Department of Automation and Remote Control
Tula State University, Tula, Russia
ivan@crown-s-soft.com, oseredin@yandex.ru

Abstract

In the article, is discussed the methodology allowing to accomplish the task of learning by instruction in conditions of large training sets. The essence of the methodology comes to introducing of a preliminary cascading classifier removing objects which present "simple" cases from a training selection.

Keywords: *machine learning, large scale train set, cascading classifier, object detection.*

1. INTRODUCTION

When accomplishing a task of searching for a face or another object in an image, a procedure generating a set of fragments is usually used, for example, the sliding-window method or the special-point method [1]. After that, each of the fragments is classified by the binary classifier, which refers an object either to the "face" class or to the "background" class ("non-face"). Thus, the quality of the search is basically defined by the quality of the binary classifier used for this.

However, when creating a quality binary classifier, the following problems arise: a large amount of training material; the large dimensionality of the attribute space; a considerable difference between the number of objects of the "face" class and the number of objects of the "non-face" class. Similar problems arise not only when detecting objects in images; thus, some procedures of solving single-class tasks presuppose generating the second set, which evenly fills the entire object space [2]. It's obvious that in this case, the number of objects in the generated set is likely to be considerably greater than that of the objects in the original set.

2. EXCLUDING CLASSIFIER TRAINING

In practical tasks of training a binary classifier, we can often make an assumption about the compactness of one of the sets. To identify it, let us call its objects "positive", and objects of the second class, which is less compact, we'll call "negative". Quite often, the number of objects of the positive class turns out to be considerably less than that of negative set objects. In those tasks when there's no opportunity to make a preliminary judgment on objects' compactness level, a set containing a greater number of objects or an arbitrary set can be considered "negative".

An example of the described situation is given in Figure 1.

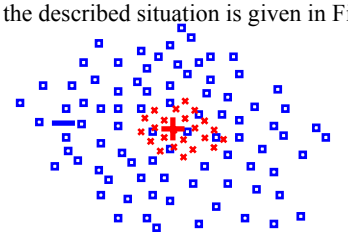


Figure 1: Objects of positive and negative sets.

With such a classifier, let us try to sort out a number of objects, which present "simple" cases. For this, let us train a classifier with the zero error rate for positive objects and, as far as possible, with a low error rate for negative objects (Figure 2).

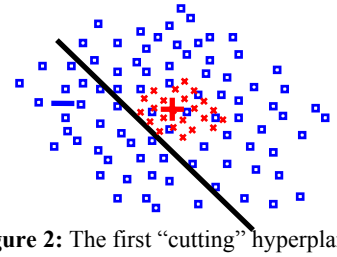


Figure 2: The first "cutting" hyperplane.

After we get the classifier with the zero error rate for positive objects, let us remove the objects, which were unambiguously referred to the negative set by the classifier, from the training set and train another classifier with the zero error rate for positive objects (Figure 3).

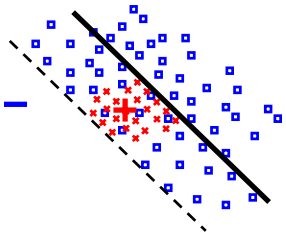


Figure 3: The second "cutting" hyperplane.

We'll continue following the specified procedure until the number of objects cut off by every new hyperplane is small enough. As a result, we'll get the situation shown in Figure 4.

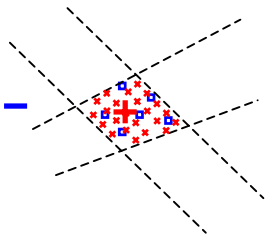


Figure 4: The training set after introducing four cutting hyperplanes.

After the procedure has been completed, we can unite the modeled hyperplanes on the principle of "logic AND", that is, refer an object to the positive set only if **all** of the classifiers treated the object as positive.

3. ANALYSIS OF TRAINING PROCEDURE RESULTS

It's quite possible, that the classifier we got in this way is already of a high quality level – in this case, it can be used independently.

The high quality level of such a classifier indirectly indicates that that the hypothesis about the compactness of the positive set, which was originally put forward, is true. In case of negative results, perhaps, it's reasonable to make an assumption about the compactness of the negative set and repeat the procedure interchanging the sets.

Negative results of the procedure can also indicate the high level of noise in data. In this case, is recommended analyzing the positive objects which are most close to the hyperplanes (when using an SVM classifier, frame objects). Such an analysis may help to define the noise source.

4. CLASSIFIER'S CHARACTERISTICS

We can try to improve the results by using the classifier we got as a "preliminary filter" for objects. To do this, the objects of the test set, which were not referred to the negative set by any of the hyperplanes, should be split with a more complex classifier, for example, by moving the objects to a space of larger dimensionality or by using a non-linear nuclear for the support vector method. Here, a procedure similar to that described above can also be used.

Considerably reducing the number of objects in the negative set allows us to use more complex recognition procedures at the next stage of training. Using such procedures for the original training set would be extremely difficult or even impossible due to the large number of objects.

Another advantage of the resultant set of hyperplanes is the ability to check an object incompletely at the recognition stage; thus, if the next hyperplane referred an object to the negative set, checking the object with other cascades doesn't make sense as it can be immediately referred to the negative set.

Among the disadvantages of the algorithm, we should note its extremely high susceptibility to noise in data, which virtually unambiguously defines the sphere of its usage as a preliminary sorting out of "simple" cases before using a complex classifier.

Another disadvantage is that the positive set must be a highly completed set as the decision rule is entirely defined by the hull of the set, which can lead to a high error rate if there are no objects close to a section of the positive set's real hull.

Also, it should be noted that the algorithm is somehow similar to boosting procedures [5]. The algorithm is not a subset of these procedures as the final classifier is trained not as a weighted total of individual classifiers' results, but as uniting them by the "logic AND" operation. However, in some respect, the algorithm can be regarded as a certain ultimate case of a boosting procedure. As compared with this group of algorithms, the described one advantageously differs from them as instead of assigning low importance to the successfully classified objects, it completely excludes them, which positively influences the time performance of the training procedure and of the recognition procedure as well.

5. EXPERIMENTAL RESULTS

To test the described procedure, we used a set of images (objects) of two classes: "faces" and "non-faces". We made an assumption about the compactness of the faces set. The number of positive objects (faces) is 540 thousand, the number of negative objects - 22 mln. The basic problem of this task was that a large number of objects in the negative set did not allow using the procedure of training an SVM classifier in a space with 1200 attributes (due to unreasonable memory and CPU time requirements). The dimensionality of the attribute space for the preliminary cascading classifier was chosen to be 270 attributes. Thus, using 36 hyperplanes at the training stage, we have got a classifier with the zero error rate for the faces set and with the error rate of 2.25% for the non-faces set. Thus, the number of objects in the negative set was reduced from 22 mln. to 510 thousand, which allowed using a classifier with the dimensionality of 1200 attributes at the next stage of training.

6. CONCLUSION

The described methodology of uniting classifiers into a cascade can be highly useful in terms of large amount of the training material, and also, to optimize time performance of complex classifiers by reducing the number of objects being classified at the expense of introducing an additional cascade of simple classifiers passing judgments in simple cases, which are prevailing. This methodology can also be used as an independent classifier, provided that the compactness of objects of one of the classes is high.

7. REFERENCES

[1] Крестинин И. А., Середин О. С. Метод особых точек в задачах поиска лиц на графических изображениях // Известия ТулГУ, серия «Технические науки», Вып. 3 – Тула: ТулГУ, 2008, с. 218-227.

[2] Roberts, S., Tarassenko, L., Pardey, J., and Siegwart, D. A validation index for artificial neural networks. In Proceedings of Int. Conference on Neural Networks and Expert Systems in Medicine and Healthcare, 1994, pages 23–30.

[3] Vapnik, V.: Statistical Learning Theory. Wiley, New York, 1998.

[4] Veropoulos K., Cristianini N. and Campbell C. Controlling the sensitivity of support vector machines. Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI99), Stockholm, Sweden, 1999.

[5] R. Polikar. Ensemble Based Systems in Decision Making. IEEE Circuits and Systems Magazine, vol.6, no.3, pp. 21-45, 2006, http://engineering.rowan.edu/~polikar/RESEARCH/PUBLICATIONS/csm06.pdf

About the authors

Ivan A. Krestinin is a Ph.D. student at Tula State University, Department of Automation and Remote Control. His contact e-mail is ivan@crown-s-soft.com.

Oleg S. Seredin, was born on 18.11.1974. In 1998, obtained the M.S. degree at Tula State University, Department of Automation and Remote Control. In 2001, received the Ph.D. degree at Computing Center of the Russian Academy of Sciences, Moscow (Thesis: Methods and algorithms of featureless pattern recognition). Since 2002, he has been an Associate Professor at Tula State University, Department of Technical Cybernetics. Lecture Courses: Object-oriented Programming, Data Mining. Theoretical Research: featureless pattern recognition, linear methods of data analysis. His contact e-mail is oseredin@yandex.ru.

Real-Time Object Detection in Video Streams on Low Performance Embedded Systems

Karpov Alexey*
Moscow State University,
Department of Mechanics and Mathematics

Abstract

Object detection on images is an important task of computer vision. Images are often extracted from continuous video streams, since many real-life applications of object detection are various control systems. This article presents the way to utilize continuity of a video stream in order to speed up object detection by Viola-Jones-like [1] [2] algorithms.

Keywords: *object detection, face detection, embedded system.*

1. INTRODUCTION

If one looks closely at the real-file applications of object detection, such as various surveillance and control systems, human-computer interaction systems, internet conferencing, one can see that those applications require analysis of a continuous video stream. Moreover the analysis has to be done in a real-time, hence object detection also has to be done in a real-time. At the same time computers used for such tasks have to be compact, power efficient, able to work in a harsh environment, therefore usually compromising its speed. The algorithm proposed in the article is required to work in a real-time (more than 1 frames per second performance) on a TMS320 based embedded system.

2. THE ALGORITHM

The algorithms allowing (with modifications) to achieve acceptable real-time performance on the embedded system are Viola-Jones type boosted cascade algorithms [1] [2]. However these methods do not take into account that object detection is performed in a video stream. The following modifications are proposed so as to increase the performance of the base object detection algorithm.

2.1 Related Methods

The problem of object detection in a video stream is not new. There are comprehensive taxonomies [3] and papers [4] [5] which give object detection algorithms using movement detection and segmentation. Therefore such algorithms try to detect objects through evaluation and analysis of the movement regions and object tracking. These algorithms are conceptually different than the algorithm presented in this article, since the object detection itself is done using rather effective boosted cascade algorithm. Furthermore, not all of the algorithms can be implemented for a real-time environment while maintaining satisfactory detection quality. Other algorithms [6] [7] attempt to use probabilistic approach for object detection in a video stream. Such algorithms try to achieve better detection quality other than processing speed. However, recognizing the quality of object detection of Viola-Jones based algorithms as satisfactory, this article concentrates on increasing of object detection speed in a video stream.

The idea behind modification is that consecutive frames in a video stream generally tend to have regions that differ insignificantly. Hence, the object detection should be performed only in changing regions or, more precisely, the object detection should not be performed in regions where has been no changes. The algorithm tries to find only one object in a frame of a video stream (the extension for multiple objects is not difficult to implement).

2.2 Frame Differencing

The methods described in [3] [4] [5] perform analysis of each pixel in a video stream to determine if it has sufficiently changed. Thus such analysis requires significant amount of time. In order to quickly yet reliably estimate region of variance of two consecutive frames, the following method is proposed. One of the algorithms for image segmentation described in [8] uses column and row sums of the image to extract foreground. The technique is called amplitude projection segmentation. If one applies the technique to the difference of two frames, one gets the estimation of variance region for the frames. That is to say, let I_{n-1}, I_n – two consecutive frames from a video stream, then

$$HSumDiff(i) = \left| \sum_{j=1}^{width} I_{n-1}(i,j) - \sum_{j=1}^{width} I_n(i,j) \right|,$$

$$VSumDiff(j) = \left| \sum_{i=1}^{height} I_{n-1}(i,j) - \sum_{i=1}^{height} I_n(i,j) \right|.$$

Let $Hthr$ and $Vthr$ are the thresholds to control sensitivity of variances in rows and columns, respectively. Then

$$imin = \min_i \{i \mid HSumDiff(i) > Hthr\},$$

$$imax = \max_i \{i \mid HSumDiff(i) > Hthr\},$$

$$jmin = \min_j \{j \mid VSumDiff(j) > Vthr\},$$

$$jmax = \max_j \{j \mid VSumDiff(j) > Vthr\}.$$

Finally the region of variance is a rectangular area with upper left vertex at $(imin, jmin)$, width of $jmax - jmin$ and height of $imax - imin$. It should be noted that the sums $HSumDiff(i)$ and $VSumDiff(j)$ can be easily calculated from data of the base object detection algorithm as difference of the integral sums. Also the algorithm does not require to store the previous frame, but just the sums.

*karpov@fit.com.ru

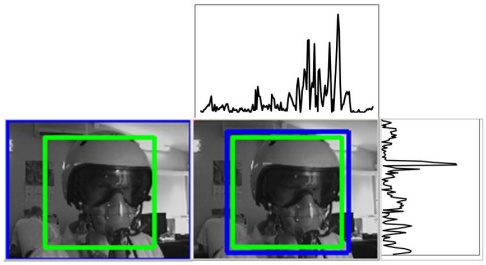


Figure 1: *Amplitude projections differencies example*

2.3 Region of Interest Computation

The object detection is performed in the region of interest (ROI) – specific rectangular area. Below is given the algorithm for determining of the region of interest. To begin with, let the *active region* (AR) is an area of frame, where the object may occur. At first the active region is the whole frame. Consider the following:

- Calculate region of variance (ROV)
- Consider the following cases
  1. If the object has been detected on the previous frame (in the rectangular area $ROBJ$), then
     (a) if ROV is too small (practically less than half of minimum possible object size), return $ROBJ$
     (b) if $ROV \bigcap ROBJ$=Ø, then $AR = AR \bigcup ROV$, return $ROBJ$
     (c) if $ROV \bigcap ROBJ \neq \emptyset$, then
        i. $ROI = ROBJ \bigcup ROV \bigcup AR$,
        ii. $ROBJ = \mathrm{DetectObject}(ROI)$,
        iii. if object is found then $AR = ROBJ$, return $ROBJ$ else $AR = ROI$, return $NOOBJFOUND$
  2. If the object has not been detected on the previous frame, then
     (a) if ROV is too small, then return $NOOBJFOUND$
     (b) if ROV is big enough, then
        i. $ROI = ROV \bigcup AR$,
        ii. $ROBJ = \mathrm{DetectObject}(ROI)$
        iii. if the object is found then $AR = OBJ$, else $AR = ROI$. Return $ROBJ$.

The result of operation of joining ($\bigcup$) two rectangular areas is a bounding box area of these rectangular areas.

2.4 Results

The proposed algorithm was tested on video streams with different frame rates (see tables 1, 2 for results). As one should expect, the more frames per second has a video stream, the more increase of processing speed is achieved by using the modified algorithm. It is important to note, that the quality of object detection is almost the same. The algorithm can be implemented as a real-time algorithm, since one can derive formulae showing upper bound for processing time for given parameters for the base algorithm and ROV calculation complexity is very low and depends only on height and width of a frame.

Video frame rate	Original algorithm	Modified algorithm
∼1 fps	18.5	21.9
4 fps	73	42.5
10 fps	430	225

Table 1: *Processing time (seconds) for the sequences*

Video frame rate	Original algorithm	Modified algorithm
∼1 fps	0.76 / 0.025	0.74 / 0.03
4 fps	0.93 / 0.001	0.9 / 0.001
10 fps	0.83 / 0.003	0.84 / 0.01

Table 2: *Performance comparison (hit rate/false alarms rate)*

3. FUTURE WORK AND CONCLUSION

The article presents method for object detection in video streams in a real-time. The method is implemented on the embedded systems based on TMS320 CPU. The method is part ASKPM (railroad safety) system which is put on approve test on Russian Railways locomotive ChS2K type. The method is also part of BASBP (aircraft safety) system which is put on approve test on planes produced by the Russian Aircraft Corporation MiG, Sukhoi design bureau and Yakovlev design bureau.

In the future, various improvements are planned for the algorithm, such as more precise region of variance determination by using more localized features, deeper feature history.

4. REFERENCES

[1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," 2001.

[2] "Rainer Lienhart et al, "Empirical analysis of detection cascades of boosted classifiers for rapid object detection.," in *"DAGM-Symposium"*, 2003, pp. "297–304".

[3] Jacinto C. Nascimento and Jorge S. Marques, "Performance evaluation of object detection algorithms for video surveillance," *IEEE Transactions on Multimedia*, vol. 8, no. 4, pp. 761–774, 2006.

[4] Rita Cucchiara et al, "Detecting moving objects, ghosts, and shadows in video streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1337–1342, 2003.

[5] Robert T. Collins et al, "A system for video surveillance and monitoring," 2000.

[6] Jacek Czyz, "Object detection in video via particle filters," *Pattern Recognition, International Conference on*, vol. 1, pp. 820–823, 2006.

[7] Kenji Okuma et al, "A boosted particle filter: Multitarget detection and tracking," in *In ECCV*, 2004, pp. 28–39.

[8] William K. Pratt, *Digital Image Processing: PIKS Inside*, John Wiley & Sons, Inc., New York, NY, USA, 2001.

SSIM modification for video error concealment methods quality assessments*

Dmitiry Kulikov
Department of Computational Mathematics and Cybernetics
Moscow State University, Moscow, Russia
dkulikov@graphics.cs.msu.ru

Abstract

The typical problem in video streaming is a packet-loss during signal transmission. Due to this fact different error concealment methods are developed. The best possible way for comparing processing quality of these methods is subjective quality assessments. But due to fact this type of estimation is very difficult and expensive to perform objective quality metrics are widely used. The most well-known are PSNR and SSIM, last one is closer to human perception. The main drawback for this metrics in video quality assessments is per-frame metrics value that does not take into account any temporal artifacts which could be more sensible to the human than static artifacts. This article is devoted to new metric development based on SSIM, that could estimate temporal artifacts as well as spatial. This new metric was tested and compared to original PSNR and SSIM for different error concealment methods with help of SAMVIQ methodology for subjective quality assessments.

Keywords: *video, error concealment, quality assessments, SSIM modification.*

1. INTRODUCTION

The task of automatic quality estimation for video error concealment methods is very important, because subjective quality assessments performing is very expensive and time-consuming task. But this task could not be solved solely with mathematical tools, because many methods during processing do not recover lost information, but change it to most similar in the frame or sequence. And the task of quality assessments has an addition not only to estimate the closeness of the original and processed video sequences, but also take into account visually noticeable artifacts. Quality measurement algorithms are based on human visual system imitation[1]. But human visual system is not fully investigated yet for truly adequate automatic estimation algorithms construction [1, 2]. This situation forces to develop new quality estimation algorithms and to find approaches to increase adequateness of known metrics. The typical work-flow for video processing algorithm quality assessment is to compute some "spatial" metric value for each frame and than to average per-frame values for the whole video sequence. The main quality metrics are PSNR and SSIM[3].

PSNR (Peak to Signal Noise Ratio) estimates inverse mean square error in logarithmic scale actually. The main idea of SSIM (Structural Similarity Index Measure) computation is to consider three main types of distortion: luminance distortions, contrast distortions and structure distortions. Every component is very essential for human perception and its combination could approximate human video or image estimation. Many experiments show that SSIM is more adequate to the human perception than PSNR and it is preferable to use while performing quality assessments tasks. But for video error concealment methods quality estimation even SSIM is not always adequate. To estimate metric adequateness the experiment described in Sec. 3. was performed. To etsimate metric correlation to subjective marks Kendall's τ was used. It shows that when average $\tau = 0.67$ for some test vectors $\tau = 0.42$. The main feature of these vectors is slow motion, processed vectors have artifacts in temporal area, e.g. not monotonous motion of some processed parts of the frames. This low correlation is explained by the fact that per-frame SSIM metric could not estimate such artifacts, but human eye is very sensible to it and sometimes temporal artifacts are more important than spatial. And because of it new metric is proposed in this article to take into account temporal artifacts.

2. PROPOSED METHOD DESCRIPTION

2.1 Algorithm schema

The main idea of proposed algorithm is to perform concealment area characteristics consistency analysis in addition to standard SSIM computation. This analysis is done by the motion vector in concealed blocks deviation power estimation comparing to the reference frame of the source and processed video sequences.

The algorithm schema is shown at Fig. 1. First of all SSIM calculation for two frames (source F_t and processed G_t) is performed. The first intensional step of algorithm is motion vector field homogeneity analysis performed for the source(F_t and F_{t-1}) and processed (G_t and G_{t-1}) video sequences. The second step is distortion coefficient computation. And the final step is SSIM temporal modification (SSIMt) metric value calculation for the whole frame.

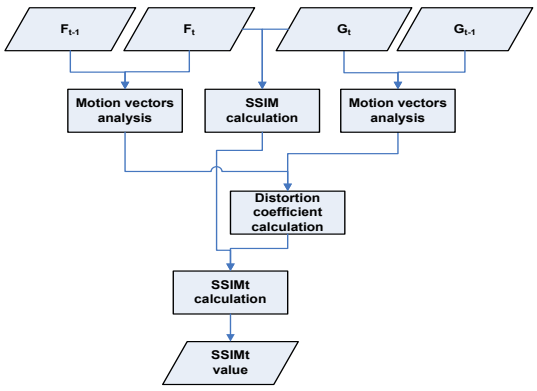


Figure 1: SSIMt calculation schema

*This work was supported by RFBR grant 07-01-00759-a.

2.2 Motion vector field homogeneity analysis

Motion homogeneity for video sequence could be performed by means of motion vector mechanism with the following approach. The homogeneity coefficient is computed as a ratio of average motion vector consistency (comparing to the neighbors) to similar value for motion vector of corresponding block in reference frame. It could be done using next algorithm:

1. Adaptive threshold calculation for horizontal and vertical components of the motion vector. This threshold is proportional to neighbors vectors value dispersion:
$\sigma_c^{TH} = \sqrt{\sum_{mv^j \in \theta(mv^i)} \left(mv_c^{AVG} - mv_c^j\right)^2} / \sum_{mv^j \in \theta(mv^i)} 1$
where c=x or y, mv_x and mv_y – horizontal and vertical components of motion vectors, mv^{AVG} – average value of the corresponding components of vectors, θ – neighborhood of motion vector.

2. Motion vector matching value computation comparing to its neighbors using calculated thresholds:
$$\beta(mv_i) = \sqrt{\beta_x^2 + \beta_y^2}, where \quad (1)$$
$$\beta_x = \sigma_x^{TH} - \alpha|mv_x^{AVG} - mv_x^i| \quad (2)$$
$$\beta_y = \sigma_y^{TH} - \alpha|mv_y^{AVG} - mv_y^i|, \quad (3)$$
where α – weighting coefficient.

3. Motion vector matching coefficients for the current and reference frame ratio calculation:
$$\beta = (\beta(mv_i(t)) + \varepsilon) \,/\, (\beta(mv_i(t-1)) + \varepsilon), \quad (4)$$
where ε – small coefficient to prevent devision by zero.

2.3 Distortion coefficient and metric value calculation

During concealed area (block) temporal visibility estimation the analysis have to be performed in comparison to original undistorted video sequence because of the fact that even strong differences in temporal area could be internal video sequence characteristics and do not connect with video processing algorithm.

And because of it distortion coefficient is calculated as a ratio between accordance coefficients for source video sequence B_f and reference – B_g:

$$\lambda(B) = (\min(\beta(B_f), \beta(B_g)))/(\max(\beta(B_f), \beta(B_g))) \quad (5)$$

SSIMt metric value calculation is performed using SSIM value and calculated distortion coefficient:

$$SSIMt(B) = \alpha SSIM(B) + (1-\alpha)\lambda(B), \quad (6)$$

where α – empiric weighting coefficient.

For metric value for the whole frame average values for the all blocks are used and for the video sequence metric value average per-frame values are used.

3. EXPERIMENTAL RESULTS

To perform metrics adequateness estimation comparing to subjective marks the next experiment was carried out. The test set of video sequences was chosen, it contains 4 different video sequences with different motion patterns, each sequence was corrupted and restored using 7 different video error concealment methods. Thus there are 32 test vectors including original ones. 10 experts participate the experiment.

The typical situation during the experiment is the next: two different methods A and B results are very close visually and by objective metrics at one frame, but result of method B is temporal inconsistent, for example it produces block flicking. These changes are not quite big and not estimated by frame by frame comparison, but produce strong temporal visual artifacts.

For SSIMt metric adequateness to human perception and comparing it to original SSIM and PSNR the next approach is used – for every test vector subjective mark is received using SAMVIQ methodology for subjective quality assessments from European Broadcasting Union[4]. To measure the degree of correspondence between objective and subjective marks the Kendall τ rank correlation coefficient is used. The average τ coefficients for three different are $\tau_{PSNR} = 0.6786$, $\tau_{SSIM} = 0.6964$ and $\tau_{SSIMt} = 0.7143$, comparing to average subjective marks received during test. The proposed approach leads to rank coefficient increase that means more adequate metric in the terms of correlation between objective and subjective tests.

4. CONCLUSION AND FUTURE WORK

In this article the new approach for quality metric is proposed. It is based not only on spatial metrics calculation, but also on temporal artifacts estimation using motion vector field analysis. This approach leads to more adequate metrics value in the terms of human perception (using subjective tests) comparing to original SSIM and PSNR and could be used in different video processing methods quality estimation, video error concealment methods. On of future work directions is to perform complex comparison to other motion-based SSIM modifications like [5]. The goal of new metric construction is to produce metric for video error concealment methods evaluation that is close to subjective marks. And proposed approach could be a good basis for new metrics.

5. REFERENCES

[1] Marcus J. Nadenau, David Alleysson, and Murat Kunt, "Human vision models for perceptually optimized image processing – a review," in *submitted to Proc. of the IEEE*, September 2000.

[2] Bernd Girod, "What's wrong with mean-squared error?," *Digital images and human vision*, pp. 207–220, 1993.

[3] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, 2004.

[4] Steinmann V. Sunna P. Wyckens E Kozamernik, F., "Samviq – a new ebu methodology for video quality evaluations in multimedia," *SMPTE motion imaging journal*, vol. 114, no. 4, pp. 152–160, 2005.

[5] Qiang Li and Zhou Wang, "Video quality assessment by incorporating a motion perception model," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, 2007, vol. 2, pp. 173–176.

ABOUT THE AUTHOR

Dmitriy Kulikov is a Ph.D. student at Moscow State University, Department of Computational Mathematics and Cybernetics. His contact email is dkulikov@graphics.cs.msu.ru.

Тестирование вычислительных возможностей метода MAC для моделирования динамики жидкости и газа в компьютерной графике

Спижевой А.С., Турлапов В.Е.
Нижегородский Государственный Университет им. Н.И. Лобачевского
alexey.spizhevoj@gmail.com, vadim.turlapov@gmail.com

Abstract

Paper describes new interactive implementation of known MAC-method algorithms. Performances of this implementation are investigated on the popular tasks: fluid-solid coupling; gas-solid coupling; and melting undo heat propagation. Implicit surface function for quality visualization is built. Real time of these simulations on one-core CPU is achieved for 32x32x64 cells and 30000 markers.

Key words: computer graphics, fluid modeling, Marker-And-Cell, CPU, real time

Математическая модель для жидкости

Цель данной работы: реализация на центральном процессоре метода Marker-And-Cell (MAC) и тестирование его вычислительных возможностей на популярных в компьютерной графике задачах моделирования жидкости и газа со свободными границами и их взаимодействия с твердыми телами. В литературе имеются алгоритмы и примеры решения таких задач [1-4], однако, как правило, не в реальном времени. Наша цель – интерактивные модели. При реализации мы следовали алгоритмам работ [1-3].

Основой моделирования динамики несжимаемой жидкости является уравнения Навье-Стокса:

$$\mathbf{u}_t = -(\mathbf{u}\cdot\nabla)\mathbf{u} + \nu\nabla^2\mathbf{u} - \frac{1}{\rho}\nabla \mathrm{p} + \mathbf{g} \qquad (1.1)$$

$$\nabla\cdot\mathbf{u} = 0\,, \qquad (1.2)$$

где $\mathbf{u} = (u, v, w)$ – вектор скорости в точке.

Граничные условия: $\mathbf{u}\cdot\mathbf{n} = 0, (x, y, z) \in \partial ModelingArea$

В качестве области моделирования принят прямоугольный параллелепипед, со сторонами кратными размеру ячейки h. Следуя работам [1-2], используем регулярную MAC-сетку.

Для соблюдении граничных условий, как и в работе [1], вводится дополнительный слой ячеек (пристеночный слой). На тех гранях ячеек сетки, которые граничат со стенами, устанавливаются скорости равными нулю. На остальных гранях ячеек сетки значения устанавливаются таким образом, чтобы обеспечивалось свободное скольжение вдоль стен. Для этого необходимо скопировать в грани ячеек приграничного слоя, расположенные перпендикулярно стенам, значения из граней ближайших ячеек.

Процесс интегрирования

1. Поле скоростей задано из начальных условий или из предыдущего шага. Шаг интегрирования выбирается из условия Куранта-Фридрихса-Леви $\Delta t < \frac{h}{\max(|u|,|v|,|w|)}$, и сходимости численных методов $\Delta t < \frac{h^2}{6\nu}$.

2. Осуществляем обновление координат маркеров методом Рунге-Кутты второго порядка (для дыма подходит и метод Эйлера). При интерполяции скорости в точках близких к поверхности жидкости, возникает необходимость использовать скорости в ячейках, которые не были вычислены при интегрировании на предыдущем шаге. В таком случае в ячейке вычисляется среднее значение необходимой компоненты скорости по соседним, содержащим маркеры ячейкам.

3. Во вновь заполненных ячейках скорости в центре каждой грани вычисляются на основе скоростей маркеров содержащихся в них. Для этого берется среднее арифметическое соответствующих компонент скоростей всех частиц, попавших в кубик со стороной h, центр которого находится в той точке, скорость в которой определяется. Если для не найдется частиц попавших в кубик, то берется ближайшая частица.

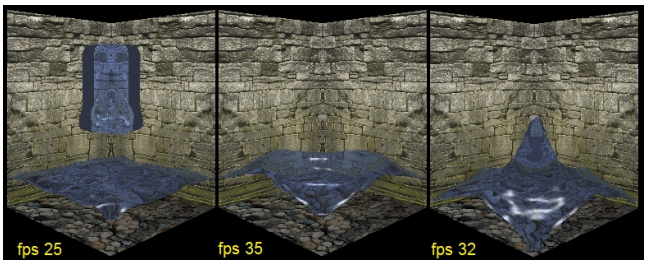
4. Решение уравнений (1.1) производится отдельно для каждого члена. Скорости обновляются на гранях ячеек, имеющих жидкость по обе стороны. Вычисляется $\mathbf{u}^1 = \mathbf{u} + \Delta t(-(\mathbf{u}\cdot\nabla)\mathbf{u} - \nu\nabla^2\mathbf{u} + \mathbf{g})$ с использованием semi-Lagrangian техники для конвекции [3] и метода Эйлера для остальных членов.

5. Обеспечиваем выполнение граничных условий у стен.

6. $\mathbf{u}^1$ не удовлетворяет уравнению несжимаемости (1.2). Для компенсации этого вычисляется $\mathbf{u}^2 = \mathbf{u}^1 - \frac{\Delta t}{\rho}\nabla p$, где p находится из условия (1.2). На данном шаге, при нахождении давления, возникает уравнение Пуассона, которое в дискретной форме представляет собой систему линейных уравнений с разреженной, симметричной и положительно определенной матрицей. Решение может быть найдено с помощью метода сопряженных градиентов [1]. $\mathbf{u}^2$ – есть новое корректное поле скоростей.

7. Обрабатываем поверхность жидкости (границу жидкость-воздух). В таких точках мы не можем воспользоваться уравнением (1.1), поэтому скорости в ячейках на поверхности устанавливаются непосредственно из условия (1.2) в разностной форме. Его решение не единственно, когда рассматриваемая ячейка (содержащая хотя бы один маркер, но граничащая с воздухом) имеет более чем одну грань, граничащую с воздухом. В таком случае используются эвристики [1].

8. Повторяем все сначала.



Моделирование дыма

Дополнительно вводятся два скалярных поля: плотность дыма и его температура, динамика которых в простейшем случае (только конвекция, без диффузии) описывается следующими уравнениями:

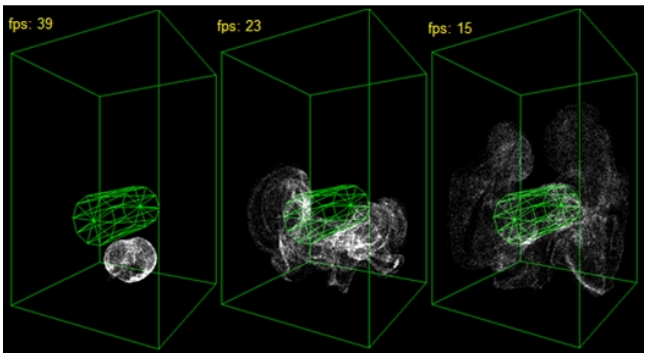
$$\mathrm{d}_t = -(\mathbf{u}\cdot\nabla)\mathrm{d},\ \mathrm{T}_t = -(\mathbf{u}\cdot\nabla)\mathrm{T}$$

Для учета влияния плотности и температуры на динамику дыма используется следующая модель:

$$\mathbf{f}_{smoke} = (0, -\alpha \mathrm{d} + \beta(\mathrm{T} - \mathrm{T}_{ambient}), 0)$$

Как и давление, данные величины хранятся в центрах ячеек. Для их обновления полей на каждом шаге используется semi-Lagrangian техники. Как известно semi-Lagrangian техника может приводить к возникновению диссипативных эффектов [2]. В частности при моделировании дыма было отмечено, что суммарная плотность и температура со временем уменьшаются, чего не должно быть согласно математической модели. Предполагая, что это изменение невелико, мы просто умножаем значение плотности и температуры в ячейках на отношение заданного суммарного значения к текущему суммарному значению. Артефакты в результате использования предложенного алгоритма замечены не были. Для компенсации диссипации скорости вводится дополнительная сила [2], не дающая возникающим вихрям быстро затухать:

$\mathbf{f}_{vorticity} = \gamma h(normalize(\nabla|\boldsymbol{\omega}|) \times \boldsymbol{\omega})$, где $\boldsymbol{\omega} = \nabla \times \mathbf{u}$.



Твердые тела

Мы рассматриваем одностороннее взаимодействие: когда твердое тело влияет на жидкость или газ, а они на твердое тело не влияют.

Множество ячеек, которые пересекаются границей твердого тела, обрабатываются также как и другие, но перед передвижением маркеров в них производится коррекция скоростей. Она производится таким образом, чтобы нормальная составляющая относительной скорости в центре граничных ячеек была направлена от границы твердого тела:

1. Вычисляется вектор скорости $\mathbf{u}_c$ в центре ячейки.
2. $\mathbf{u}_c^{new} = \mathbf{u}_c + \max(0, (\mathbf{u}_0 - \mathbf{u}_c) \cdot \mathbf{n}_c)\, \mathbf{n}_c$, где $\mathbf{u}_0$ – скорость тела, $\mathbf{n}_c$ – усредненная внешняя нормаль.
3. Во все шесть граней данной ячейки записывается вычисленное значение $\mathbf{u}_c^{new}$.

Для эффективного поиска ячеек пересекаемых телом, мы используем следующий алгоритм: перед началом вычислений мы набрасываем на каждый полигон (используются треугольники) тела случайным образом точки. Их число в одной ячейке ограничиваем сверху некоторой константой, далее эти точки считаются “приклеенными” к твердому телу и движутся вместе с ним.

Пусть нам дан треугольник $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$. Точка, лежащая на нем, может быть представлена в виде:

$\boldsymbol{p}_{triangle} = \alpha_1 \cdot (\boldsymbol{b} - \boldsymbol{a}) + \alpha_2 \cdot (\boldsymbol{c} - \boldsymbol{a})$,

$\alpha_1 \geq 0, \alpha_2 \geq 0, \alpha_1 + \alpha_2 \leq 1$.

Таким образом, остается сгенерировать необходимое число пар (α_1, α_2) удовлетворяющих вышеприведенному условию.

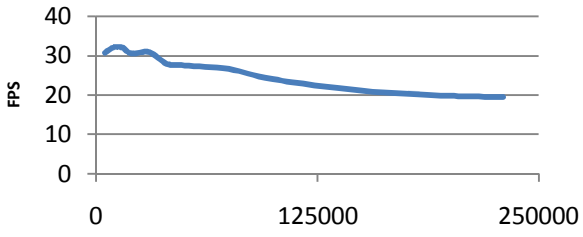
Пусть тело пересекает только те ячейки, в которых есть его представители – набросанные точки. На регулярной сетке индекс ячейки, в которой лежит заданная точка, может быть вычислен следующим образом: $(i, j, k) = \left(\left\lfloor \frac{x}{h} \right\rfloor, \left\lfloor \frac{y}{h} \right\rfloor, \left\lfloor \frac{z}{h} \right\rfloor\right)$, где i, j, k – индексы ячейки, x, y, z - координаты точки, h - размер ячейки.

Нормаль $\mathbf{n}_c$ вычисляется, как средняя по нормалям исходящим из точек-представителей (в них нормаль определяется однозначно по треугольникам, которым они принадлежат).

Весь нижний уровень вычислений векторизован.

Результаты[1]

Достигнута интерактивность реализации MAC-метода для 32x32x64 ячеек и 30000 маркеров, т.е. на уровне и выше [4]. Для визуализации жидкости использовался ray tracing и photon mapping на GPU (предоставленные на Graphicon2009 Денисом Боголеповым). Дым визуализировался сглаженными точками большого размера с использованием OpenGL.



Литература

[1] Carlson, M. Rigid, *Melting, and Flowing Fluid* / PhD thesis, Georgia Institute of Technology. 2004.

[2] Fedkiw, R., Stam, J., and Jensen, H. W. *Visual simulation of smoke.* In Proceedings of the 28th annual conference on Computer graphics and interactive techniques, ACM Press, p.15–22. 2001.

[3] David, C., David C. and Parris, K. E. *Fluid Flow for the Rest of Us: Tutorial of the Marker and Cell Method in Computer Graphics* / Brigham Young University.

[4] Batty, Ch. , Bertails, F., Bridson, R. *A Fast Variational Framework for Accurate Solid-Fluid Coupling* / University of British Columbia, p.1-7, In Proceeding of SIGGRAPH 2007.

1 Intel® Core™ 2 Duo CPU P8400, RAM 3.0GB, NVIDIA GeForce 9600M GS, Windows Vista™ Home Basic

Проблемы выделения объектов в компрессированном потоке изображений

П. В. Харебов, С. В. Новиков
Физический факультет ПГУ
Пермский Государственный Университет, Пермь, Россия
kharebov.p@yandex.ru, novikov@networkvideo.ru

Аннотация

Обнаружение и отслеживание объектов требует устойчивой работы алгоритмов моделирования фона. Устойчивость необходима как к шуму, так и к изменениям освещения. Кроме того, стандарты компрессии, используемые в системах охранного телевидения, вносят дополнительные артефакты, затрудняющие анализ видеопотока.

Для отслеживания объектов используется техника «вычитания фона». Для автоматического выбора порога, устойчивого к артефактам, вносимых сжатием, используется метод блочного анализа шума.

Ключевые слова: *Видеонаблюдение, вычитание фона, автоматический выбор порога.*

1. ВВЕДЕНИЕ

Компрессия видеопотоков в системах охранного телевидения неизбежна из-за необходимости хранения и передачи большого объема данных. Так, например, IP камеры компрессируют видеопоток внутренними средствами перед отправкой по сети.

Существуют две категории стандартов сжатия: покадровые методы (MJPEG) и межкадровые, основанные на оценке движения (MPEG-2, MPEG-4, H.264). Каждый из алгоритмов вносит характерные артефакты в изображение.

Стандарт MJPEG сжимает каждый кадр при помощи алгоритма JPEG. Сжатие приводит к потере мелких деталей изображения и сглаживанию шума.

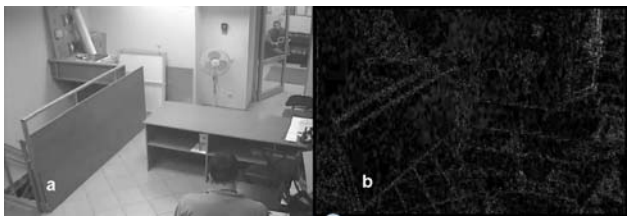


Рис. 1: Разность между последовательными кадрами.

Например, на рис. 1(a) показан JPEG-сжатый кадр, полученный с камеры охранного телевидения. На рис. 1(b) – разница между двумя последовательными JPEG-сжатыми кадрами. Рис. 1(b) дает характерное представление о шумах при JPEG сжатии – отклонение значений пикселов на границах больше чем на однородных областях.

Для видео, сжатого с использованием *компенсации движения*, кроме описанных выше шумов на краях изображений, появляющихся при JPEG-сжатии, добавляется еще один характерный артефакт – «пульсирующий шум», вызванный сглаживанием движения в пределах группы кадров. Разница максимальна между последним разностным кадром в группе и новым опорным кадром.

Решаемая нами задача – выделение объектов и отслеживание их перемещений в компрессированном видеопотоке. Пусть дан поток изображений $I^t(i,j)$, где $t = 0,1,2..$ - порядковый номер, i, j – координаты пиксела в данном изображении. Элемент изображения (пиксел) – задается значением яркости (уровень серого) в данной точке.

Техника «вычитания фона» широко распространена для выделения объектов в видео [1]. При этом каждый пиксел изображения классифицируется как принадлежащий переднему или заднему плану путем сравнения текущего изображения $I^t(i,j)$ с фоном $B^t(i,j)$. Главным критерием выбора метода «вычитания» была скорость вычисления, что критично для систем видеонаблюдения, работающих в реальном времени и обрабатывающих несколько потоков одновременно. Как показано в [2], наиболее быстрым является метод, при котором фон $B^t(i,j)$ представляет собой усредненное по времени значение $I^t(i,j)$. Пиксель (i,j) считается принадлежащим переднему плану, если $\left|I^t(i,j) - B^t(i,j)\right| > \tau$, где τ - некоторый порог.

Возникает проблема выбора порога сравнения. Использование глобального для изображения порога, не позволяют артефакты, вносимые сжатием – порог сравнения τ должен быть своим у отдельного пиксела. Кроме того, шумы не постоянны, что приводит к необходимости изменения порога τ не только в пространстве, но и во времени.

Цель данной работы – разработка метода автоматического выбора порога τ для каждого пиксела, учитывающий историю шума.

2. АНАЛИЗ ШУМА

Шумовые отклонения текущего значения пиксела от фонового значения, присутствуют всегда. Для поиска амплитуды шума изображение разбивается на блоки размером $P * P$ пикселов. В каждом блоке ищется максимальное отклонение между двумя последовательными кадрами:

$$D^t(k,l) = \max_{i,j}\left(\left|I^t(i,j) - I^{t-1}(i,j)\right|\right),\ i = \overline{k,k+P},\ j = \overline{l,l+P}$$

В течение N кадров ищется максимальное значение пиксела блока:

$$M^f(k,l) = \max_{v}\left(D^v(k,l)\right),\ v = \overline{t,t-N+1},$$

причем, для учета «пульсирующего шума», N должно быть не меньше числа кадров в группе N_1 : $N \geq N_1$.

Т.к. отклонение может быть вызвано движением, слишком большие значения $M^f(k,l)$ обрезаются так, что если $M^f(k,l) > \frac{3}{2}H^{f-1}(k,l)$, то $M^f(k,l) = \frac{3}{2}H^{f-1}(k,l)$.

Обновляется матрица истории шума $H^f(k,l)$:

$$H^f(k,l) = \beta M^f(k,l) + (1-\beta)H^{f-1}(k,l).$$

Где β – параметр, отвечающий за скорость обновления $H^f(k,l)$. $H^f(k,l)$ характеризует возможное отклонение пиксела от фонового значения.

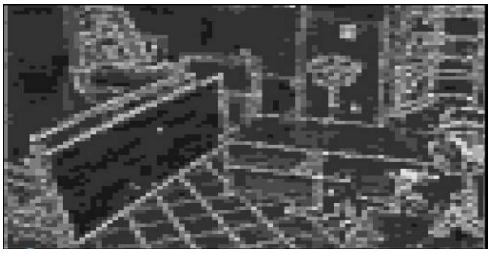


Рис. 2: История шума $H^f(k,l)$.

На рис. 2 показана история шума при $P = 8$. Для удобства восприятия значение каждого пиксела умножено на 20. Кадр соответствует рис. 1 (a). Видна неравномерность шума значений пикселов. Использование глобального фиксированного порога, при сравнении текущего кадра с фоном, привело бы к ложным срабатываниям в блоках с большим шумом и пропущенными срабатываниями в не шумящих блоках.

3. ВЫДЕЛЕНИЕ ПЕРЕДНЕГО ПЛАНА

Классификация пикселов на передний и задний план обозначается с помощью маски пикселов переднего плана $F^t(i,j)$:

$$F^t(i,j) = \begin{cases} 1, \text{ если } \left|I^t(i,j) - B^{t-1}(i,j)\right| > aH\left(\frac{i}{P}, \frac{j}{P}\right) + b \\ 0, \text{ иначе} \end{cases}$$

Где параметры a, b выбираются таким образом, чтобы при заданных N и P обеспечить необходимый уровень верных срабатываний.

$F^t(i,j)$ сглаживается с помощью операций математической морфологии.

Рис. 3: Выделение переднего плана.

На рис. 3(a) показано исходное изображение и выделенный объект, рис. 1(b) – анализируемое изображение $I^t(i,j)$, рис. 1(c) – фон $B^t(i,j)$, рис. 1(d) – маска пикселов переднего плана $F^t(i,j)$.

Последний шаг – обновление фона в пикселах заднего плана:

$$B^t(i,j) = \alpha B^{t-1}(i,j) + (1-\alpha)I^t(i,j)$$

4. ЗАКЛЮЧЕНИЕ

Предложенный метод устойчиво работает не только в пространстве яркости, но и с обработанными градиентными методами изображениями, что позволяет отслеживать объекты и в условиях сильно флуктуирующего освещения.

Реализована на практике система выделения и отслеживания объектов в компрессированном потоке изображений. Система работает в реальном времени (25 к/с) для восьми камер с разрешением 640x480 на компьютере с процессором Intel Pentium Core 2 Quad.

Достигнут компромисс между устойчивостью вычислительно сложных методов вычитания фона (например, моделирование с помощью гауссовских распределений[1,2]) и скоростью выполнения простейших методов.

5. СПИСОК ЛИТЕРАТУРЫ

[1] R.J. Radke, A. Srinivas, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: a systematic survey. IEEE Trans. Image Processing, vol. 14, no. 3, pp. 294–307, Mar. 2005.

[2] M. Piccardi. Background Subtraction Techniques: A Review. IEEE SMC/ICSMC, vol. 4, pp. 3099–3104, 2004.

Об авторах

Харебов Петр – аспирант кафедры компьютерных систем и коммуникаций ПГУ. E-mail: kharebov.p@yandex.ru

Новиков Сергей – к.т.н., руководитель проекта «Domination». E-mail: novikov@networkvideo.ru

The problems of object tracking in compressed video stream.

Objects detection and tracking demands a steady work background modeling algorithms. Robustness is necessary both for noise, and for illumination changes. Besides, the compression standards used in surveillance systems create the additional artifacts complicating the video stream analysis.

In this paper the method of the noise block analysis is presented. The technics of background subtraction is used for objects tracking.

Keywords: video surveillance, background subtraction, automatic thresholding.

About the authors

Petr Kharebov – PhD student of computer systems and telecommunications department of Perm State University. E-mail: kharebov.p@yandex.ru.

Sergey Novikov – PhD, the «Domination» project head. E-mail: novikov@networkvideo.ru.

Методы моделирования «живого» персонального почерка[1]

А.Б. Семенов
Факультет прикладной математики и кибернетики,
Тверской государственный университет, Тверь, Россия
semenov@tversu.ru

Аннотация

В статье представлены способы получения персонального рукописного шрифта для моделирования «авторского» письма. Приводятся методы «оживления» почерка, характеризующегося варьируемыми начертаниями одинаковых символов.

Ключевые слова: *векторный рукописный шрифт, жирная B-сплайновая кривая, след пера, персональный «живой» почерк.*

1. ВВЕДЕНИЕ

Основным отличием компьютерного рукописного шрифта от всех остальных является потребность в его оживлении при выводе на печать. В силу того, что природа рукописного почерка предполагает существенную изменчивость символов шрифта (человеку достаточно сложно воспроизвести две абсолютно одинаковые буквы при письме), возникает задача моделирования «авторского» письма с изменчивыми формами одинаковых букв. Подобный механизм оживления может быть получен путем внесения небольших изменений в форму каждого отдельного символа, свойственных живому почерку человека. Ставшие де факто самые популярные шрифтовые форматы Microsoft TrueType и Adobe Postscript используют контурное описание символов с помощью отрезков прямых линий и составных кривых Безье, что оказывается не всегда удобным, так как не позволяет осуществлять вариацию ширины штрихов, изменение формы пера, сложную деформацию символа. Тем не менее, решение данной задачи становится вполне допустимым, если рукописные символы представить в виде однопараметрических семейств кругов переменного радиуса с центрами на гладких кривых, так называемых жирных линий. Жирная линия представляет собой гораздо более адекватную модель следа пера с переменной шириной, чем контур границы символа. Модель жирной линии позволяет достаточно просто варьировать траекторией и шириной пера, сохраняя при этом гладкость оси и границы штриха (рис. 1). Способы получения рукописных символов в виде семейства жирных линий рассматривались в предыдущих работах автора. Так, восстановление формы букв в работе [] осуществлялось из сканированного образца почерка (рукописного алфавита) с использованием непрерывного скелетного графа растрового бинарного изображения. Другой подход в конструировании рукописных символов был предложен в работе [3], где формирование шрифта производится при непосредственном вводе данных с помощью чувствительного к силе нажатия пера графического планшета. В настоящей работе рассматриваются способы «оживления» почерка через внесение небольших динамических вариаций в ширину и форму штрихов отдельных рукописных символов.

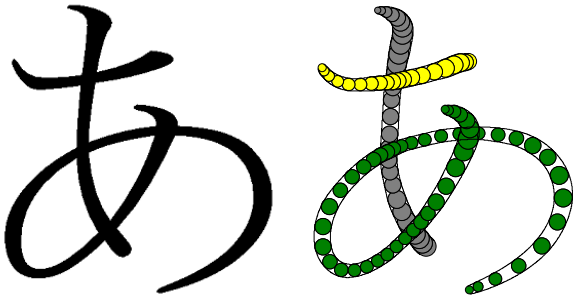


Рисунок 1: Рукописный символ.

2. МЕТОДЫ РЕДАКТИРОВАНИЯ ЖИРНОЙ ЛИНИИ

Редактирование формы жирной линии может быть достигнуто двумя основными операциями над ней – изменение положения центра и величины радиусов ее контрольных кругов. Такая модель деформации жирных линий позволяет реализовывать довольно сложные эффекты «случайности» для формы рукописных символов.

Первый из рассматриваемых способов «оживления» почерка заключается в добавлении (вычитании) небольших случайных величин к значениям радиусов контрольных кругов жирных линий: $H_{ir} = H_{ir}(1+\xi_i)$, где H_{ir} – радиус контрольного круга H_i, ξ_i – случайная величина, имеющая, например, равномерное распределение в некотором заданном интервале. Второй способ «случайной» деформации формы жирной линии заключается в небольших изменениях положения центров контрольных кругов. Здесь в качестве параметра метода выступает вектор направления и величина смещения каждого контрольного круга, значения которых могут быть получены случайным образом. Возможно также совместное применение двух указанных методик деформации. Рассмотренные способы «случайного» преобразования жирных линий нацелены на изменение положения центра и величины радиуса каждого контрольного круга в отдельности. Возможен иной композитный подход к преобразованию жирной линии как единого целого. Если рассмотреть жирную линию как семейство контрольных кругов, «нанизанных» на общую цепочку, звенья которой являются отрезками неизменной длинны, шарнирно соединенными между собой, то при перемещении одного контрольного круга в новую позицию все остальные круги также переместятся и займут новое положение. Подобная

[1] Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проекты 08-01-00670, 08-07-00270)

жесткошарнирная конструкция позволяет обеспечить простоту и удобство преобразования жирными линиями. Ниже (рис. 2) показан пошагово алгоритм композитного перемещения цепочки, состоящей из пяти точек: $P_1,...,P_5$ (P_3 – точка смещения).

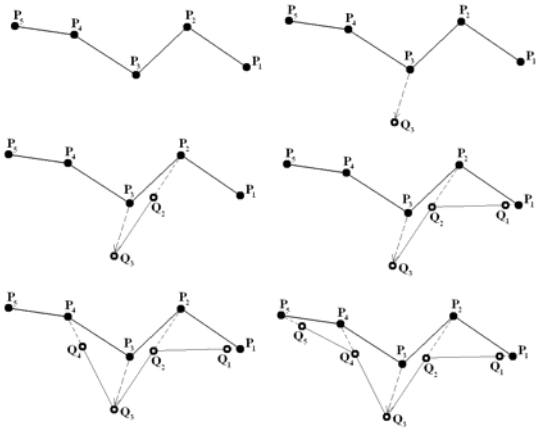


Рисунок 2: Композитная деформация жирной линии.

Предложенный метод преобразования также может быть использован при моделировании «случайности» жирной линии путем произвольного выбора одного контрольного круга, направления и величины композитного смещения. Причем подобный шаг может быть повторен произвольное число раз для большей «зашумленности» формы. Следующий способ преобразования жирной линии также оперирует только положениями контрольных кругов, не изменяя при этом значений радиусов. Метод основан на построении нелинейного преобразования двух четырехугольников (рис. 3). Формально задача трансформации фигур выглядит следующим образом:

$$\begin{cases} u = a*x*y + b*x + c*y + d \\ v = e*x*y + f*x + g*y + h \end{cases}, \text{ где}$$

(x, y) – точки четырехугольника-прообраза (слева),

(u, v) – точки четырехугольника-образа (справа),

$A_i = (x_i, y_i)$ –вершины четырехугольника-прообраза,

$B_i = (u_i, v_i)$ –вершины четырехугольника-образа,

a, b, c, d, e, f, g, h – неизвестные, однозначно определяющие преобразование, при котором внутренние точки четырехугольника A_1, A_2, A_3, A_4 трансформируются во внутренние точки четырехугольника B_1, B_2, B_3, B_4. Используя подобную схему преобразования можно форму символа подвергнуть довольно сложной и замысловатой деформации. Характер изменчивости символа будет зависеть от случайной величины смещения вершин исходного прямоугольника. Чем больше эти значения, тем более отличимым от прямоугольника будет получаемый четырехугольник, а соответственно и сами символы будут стремиться к большей «непохожести». Пример подобного преобразования формы символа через деформацию объемлющего прямоугольника представлен ниже (рис. 3).

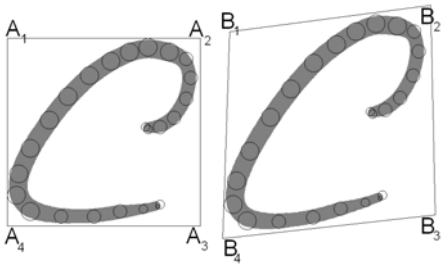


Рисунок 3: Нелинейное преобразование жирной линии.

При реализации описанных методов была реализована возможность вывода на принтер произвольного текста со «случайной» деформацией формы символов. Ниже показан пример подобной реализации (рис. 4).

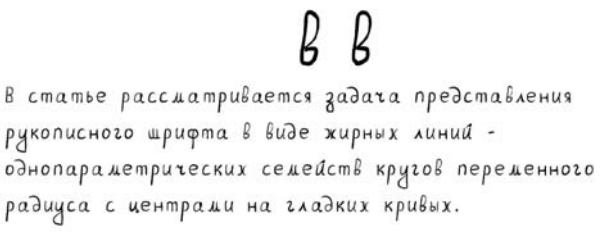


Рисунок 4: Вывод текста на печать.

3. ЗАКЛЮЧЕНИЕ

Рассмотренные в работе методы получения «случайных» форм рукописных символов дают возможность моделировать «живое» письмо при выводе (печати) произвольного текста на принтер. Идея заключается в том, что при формировании очередного символа на канве принтера «включается» алгоритм модификации формы со случайным набором значений параметров этого метода. И в этом случае даже одни и те же буквы алфавита будут иметь небольшие визуальные различия в форме символов, как и в настоящем живом почерке человека.

4. ЛИТЕРАТУРА

[1] Adobe Type Library Reference Book (3rd Edition), Adobe Press; 2007.

[2] Каров П. Шрифтовые технологии. Описание и инструментарий. М.: Мир, 2001.

[3] Семенов А.Б. Новый подход в конструировании рукописных шрифтов. // Труды 18 международной конференции ГРАФИКОН-2008. Москва, МГУ, 2008.

[4] Таранов Н.Н. Рукописный шрифт. Вища школа, 1986.

Авторы

Семенов Андрей Борисович, кандидат физико-математических наук, доцент кафедры информационных технологий Тверского государственного университета.

E-mail: semenov@tversu.ru

Моделирование каустик на графическом процессоре

Денис Боголепов, Татьяна Удалова, Вадим Турлапов
Нижегородский государственный университет им. Н.И. Лобачевского
Факультет вычислительной математики и кибернетики, Нижний Новгород, Россия
bogdencmc@inbox.ru, tauname@rambler.ru, vadim.turlapov@cs.vmk.unn.ru

Аннотация

The goal of this work is adaptation of the photon mapping method for caustics modeling in real time. The hybrid algorithm executed both on CPU and GPU is proposed. Building of the photon map and ray tracing are performed on GPU. To improve performance preliminary sorting of the photon map on CPU is used. Performance estimation is presented.

Keywords: *Interactive Graphics, GPU Ray Tracing, GPU Photon Mapping, Caustics, GPGPU.*

1. ВВЕДЕНИЕ

Целью настоящей работы является адаптация *метода фотонных карт* [1] для исполнения на графическом процессоре в реальном масштабе времени, что позволит корректно моделировать отражающие и прозрачные объекты.

2. РЕАЛИЗАЦИЯ МЕТОДА ФОТОННЫХ КАРТ

2.1 Простейший подход к моделированию каустик

Для моделирования каустик на основе метода фотонных карт можно предложить следующий двухпроходный алгоритм. При первом проходе в соответствии с расположением и ориентацией источника света генерируются фотоны света, число которых соответствует числу текселей в текстуре фотонной карты `PhotonMapTexture`. В простейшем случае можно рассматривать *точечные* (испускают фотоны равномерно по всем направлениям) или *прямоугольные* (из произвольной точки поверхности испускают фотоны в произвольном направлении полупространства) источники света.

Рисунок 1. Примеры источников света

Для каждого фотона света прослеживается траектория его распространения и возможные взаимодействия с объектами сцены. Если фотон соударяется с *диффузным* объектом сцены, в качестве результата фрагментный шейдер возвращает *координаты* найденной точки соударения. Если объект обладает *отражающими* или *преломляющими* свойствами, отслеживаются дальнейшие взаимодействия фотона вдоль направления *отражения* или *преломления* соответственно. Наконец, если фотон покинул сцену без соударения с объектами, в качестве результата возвращаются заведомо *недопустимые* координаты. Интенсивность всех фотонов полагается одинаковой и задается параметром I_{photon}. Данный подход не обеспечивает моделирования *хроматических аберраций*. Однако это ограничение может быть снято путем генерации фотонов отдельно для каждого канала RGB и использования дополнительной текстуры для хранения интенсивностей.

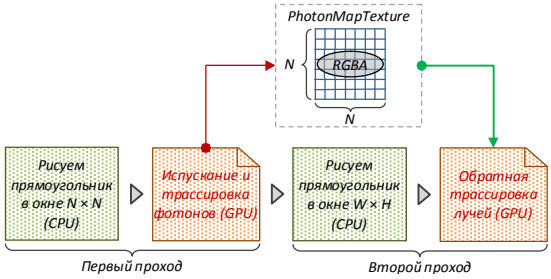


Рисунок 2. Простейшая реализация метода фотонных карт

При втором проходе выполняется традиционный алгоритм обратной трассировки лучей. При этом к вычисленной в точке соударения освещенности прибавляется интенсивность фотонов из некоторой окрестности, которые доступны посредством обращения к текстуре фотонной карты. В данной работе использовалась следующая формула для вычисления итоговой освещенности точки:

$$I = I_{ray\ tracing} + I_{photon} \times \sum_{i=0}^{N^2-1} \max\left\{0,\ 1 - \frac{|p - p_i|}{\varepsilon}\right\},$$

где p – координаты точки соударения, в которой вычисляется освещенность, а p_i – координаты фотона, записанного в i-ый тексель текстуры фотонной карты. Параметр $\varepsilon > 0$ задает радиус окрестности, с которой производится сбор фотонов. Значение данного параметра подбирается эмпирически для конкретной сцены.

2.2 Развитие подхода

Описанный подход имеет низкую эффективность, поскольку во время второго прохода вычисление освещенности каждой точки соударения требует полного перебора всех элементов текстуры фотонной карты. Эффективным способом повышения производительности является предварительная *сортировка* данной текстуры, что позволит фрагментному шейдеру обратной трассировки использовать алгоритм бинарного поиска для определения фотона с *минимальными* и *максимальными* координатами в ε-окрестности точки соударения. Если N_1 и N_2 – индексы фотонов с минимальными и максимальными координатами соответственно, то формула для вычисления итоговой освещенности точки соударения перепишется следующим образом:

$$I = I_{ray\ tracing} + I_{photon} \times \sum_{i=N_1}^{N_2} \max\left\{0,\ 1 - \frac{|p - p_i|}{\varepsilon}\right\}.$$

Таким образом, суммирование производится не по всем элементам текстуры фотонной карты, а лишь по некоторому

промежутку текстурных координат. Модифицированная схема визуализации выглядит следующим образом:

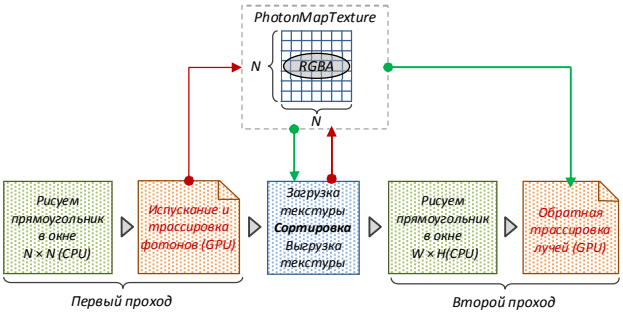


Рисунок 3. Улучшенная реализация метода фотонных карт

Указанная сортировка может быть выполнена как на *центральном*, так и на *графическом* процессоре. Сортировка на графическом процессоре является весьма затратной: обработка текстуры из миллиона элементов требует выполнения сотен проходов рендеринга [3]. С другой стороны, использование центрального процессора требует загрузки текстуры в системную память и последующей выгрузки в память графического ускорителя. Данные операции нагружают системную шину и могут служить лимитирующим фактором для производительности. Тем не менее, при использовании не слишком больших фотонных карт (до миллиона элементов) данный подход может оказаться вполне эффективным и используется в данной работе. Сортировка проводилась с помощью стандартной библиотечной функции *qsort* языка C.

3. ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ

Оценка производительности рассмотренных алгоритмов проводилась на примере визуализации водной поверхности.

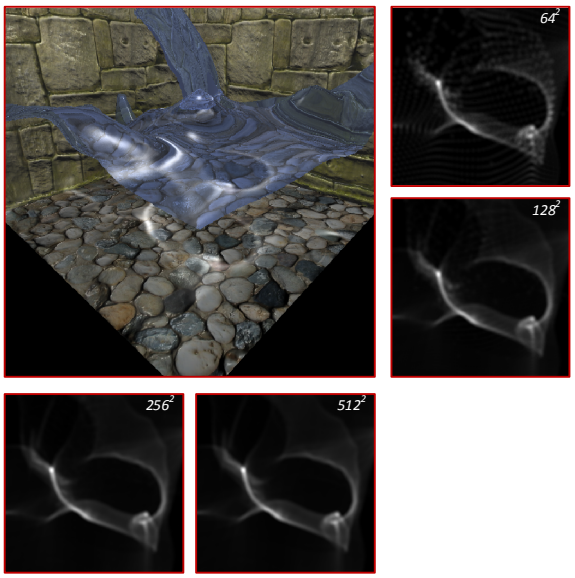


Рисунок 3. Изображения каустик для фотонных карт различного размера (*64 × 64 – 512 × 512*)

На основе физических расчетов формируется неявно заданная функция трех переменных $\varphi(x,y,z)$, нулевое значение уровня которой соответствует водной поверхности.

Для проведения эксперимента использовался графический ускоритель AMD Radeon 4850 и центральный процессор AMD Phenom II X3 710. Размер окна визуализации составлял 512 × 512 точек, значение параметра ε выбиралось равным ~1,5% от размера сцены.

В следующей таблице представлены результаты замеров производительности.

Таблица 1. Результаты замеров производительности

Размер фотонной карты	*Время загрузки и выгрузки текстуры (мс)*	*Время сортировки фотонной карты (мс)*
64 × 64	*1*	*1*
128 × 128	*3*	*5*
256 × 256	*7*	*22*
Размер фотонной карты	*Простейший подход (к/с) для глубины трассировки 1/2/3/4*	*Улучшенный подход (к/с) для глубины трассировки 1/2/3/4*
64 × 64	*9/6/5/4*	*52/40/32/26*
128 × 128	*2/1/0/0*	*24/18/15/12*
256 × 256	*0/0/0/0*	*15/11/8/4*

4. ЗАКЛЮЧЕНИЕ

Предложен упрощенный вариант метода фотонных карт для исполнения на графическом процессоре в реальном масштабе времени. Данный метод позволяет корректно моделировать кустики – области с резко возрастающей интенсивностью светового поля, возникающие при взаимодействии света с прозрачными или отражающими объектами сцены. Для повышения эффективности визуализации предложенный алгоритм выполняет предварительную сортировку фотонной карты на центральном процессоре. Данный подход обеспечивает приемлемую скорость работы при использовании фотонных карт размера 256 × 256 элементов.

5. ЛИТЕРАТУРА

[1] Henrik Wann Jensen. "*Realistic Image Synthesis Using Photon Mapping*". AK Peters. Ltd., Massachusetts. 2001.

[2] Tin-Tin Yu, John Lowther, Ching-Kuang Shene. "*Photon mapping made easy*". ACM SIGCSE Bulletin, v.37 n.1, 2005.

[3] Timothy J. Purcell1, Craig Donner, Mike Cammarano, Henrik Wann Jensen and Pat Hanrahan. "*Photon mapping on programmable graphics hardware*". Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference, 2003.

[4] Jag Mohan Singh, P. J. Narayanan, "*Real-Time Ray-Tracing of Implicit Surfaces on the GPU*". IEEE Transactions on Visualization and Computer Graphics, 2009.

Построение параметрической модели головы человека по набору изображений

Максим Федюков
Факультет вычислительной математики и кибернетики
Московский государственный университет им. М. В. Ломоносова
mfedyukov@graphics.cs.msu.ru

1 Введение

Задача моделирования головы человека является актуальной во многих областях: идентификации, отслеживании, видеокодировании на базе трехмерных моделей, моделировании виртуального присутствия и других. В зависимости от области применения моделирование может производиться с помощью трехмерного лазерного сканера [1, 2], по стереопаре [3, 4], по одной фотографии [5, 6], либо по набору изображений: фотографиям или кадрам видеопоследовательности [7, 8]. При постановке задачи реконструкции всей модели головы (но не только модели лица) с минимальными требованиями к входным данным оптимальным является последний подход. Также методы моделирования можно классифицировать по типу выходных данных. В некоторых работах ими является облако точек [9]. Во многих работах реконструированная форма представляется в виде полигональной модели [3, 5, 6, 8]. Отдельной задачей, ставшей актуальной в последние годы с развитием многопользовательских виртуальных сред является реконструкция модели, представляющей собой компактный набор параметров, описывающий морфологические характеристики головы человека.

2 Постановка задачи

Везде, где не оговорено обратное, будем понимать под *полигональной моделью* u вектор $(u_1, u_2, ..., u_N)$, $\forall i = 1, ..., N, u_i \in \mathbb{R}^3$, состоящий из радиус-векторов вершин. Множество граней фиксировано, и потому не рассматривается с целью упрощения модели. На отдельных же этапах, касающихся получения и работы с текстурой, будем рассматривать *текстурированную полигональную модель* $\tilde{u} := (\tilde{u}_1, \tilde{u}_2, ..., \tilde{u}_N), \forall i = 1, ..., N, \tilde{u}_i \in \mathbb{R}^5$, $\tilde{u}_i = (\tilde{u}_{i,x}, \tilde{u}_{i,y}, \tilde{u}_{i,z}, \tilde{u}_{i,u}, \tilde{u}_{i,v})$, $\tilde{u}_{i,x}, \tilde{u}_{i,y}, \tilde{u}_{i,z}$ — пространственные координаты вершины, $\tilde{u}_{i,u}, \tilde{u}_{i,v}$ — текстурные координаты вершины. Под *параметризованной полигональной моделью* a будем подразумевать $a := b + Sp$, где $b = (b_1, b_2, ..., b_N)$ — *базовая* полигональная модель, задающая координаты вершин по умолчанию, $S := [s_{ij}]$ $(i = 1, 2, ..., N;\ j = 1, 2, ..., K;\ s_{ij} \in \mathbb{R}^3)$ — матрица *смещений*, $p := (p_1, p_2, ..., p_K)$ $(j = 1, 2, ..., K; p_j \in \mathbb{R}, -1 \le p_j \le 1)$ — вектор *параметров* модели, N — число вершин полигональной модели, K — число параметров. Далее такую модель a для краткости будем называть просто *параметрической моделью*. Базовая полигональная модель b и матрица смещений S считаются фиксированными. Решаемая задача заключается в разработке системы, определяющей значения всех параметров p и текстуру по набору из 1—4 фотографий (анфас, в профиль слева, в профиль справа и сзади).

3 Предлагаемый метод

Схема работы системы представлена на рис. 1. Базовая полигональная модель предварительно размечена. *Универсальная разметка* $T := (t_1, t_2, ..., t_M)$ представляет собой набор векторов $t_i := (t_1^i, t_2^i, ..., t_{L_i}^i)$ $(\forall i, j, t_j^i \in \mathbb{N}, t_j^i \le N;\ i = 1, 2, ..., M;\ j = 1, 2, ..., L_i;\ M$ — число векторов, L_i — число элементов векторов, N — число вершин полигональной модели), элементами которых являются индексы t_i^i вершин базовой полигональной модели. Таким образом, вектор $q_i = (q_1^i, q_2^i, ..., q_{L_i}^i)$, где $\forall i, j, q_j^i = b_{t_j^i})$ задает ломаную, узлами которой являются вершины базовой полигональной модели. Далее $(q_1, q_2, ..., q_M)$ будем обозначать как Q. Каждая ломаная описывает одну характерную черту лица человека, такую как контур брови, носа, или всей головы. Разметка (ломаные) заданы для каждой проекции — анфас, в профиль слева, в профиль справа и сзади: T_f (Q_f), T_l (Q_l), T_r (Q_r) и T_b (Q_b), соответственно. Однако, в силу того, что работа с разметкой и ломаными на большинстве этапов происходит одинаково для каждой проекции, для упрощения записи, где это возможно, будем писать T и Q, не уточняя индекс проекции.

3.1 Предобработка

На каждой фотографии распознаются характерные точки лица, такие как уголки глаз или кончик носа. На каждую фотографию проецируется соответствующий набор ломаных: $\Omega = \mathbf{P}Q$, где $\mathbf{P} : \mathbb{R}^3 \to \mathbb{R}^2$ — оператор ортогонального проецирования на координатную плоскость. Узлы спроецированных ломаных, соответствующие найденным характерным точкам, перемещаются в соответствующие координаты. По ним вычисляется матрица аффинного преобразования, с помощью которой перемещаются все остальные узлы каждой ломаной. После этого дополнительно скорректировать положение узлов спроецированных ломаных Ω можно вручную. Далее фотографии, как текстурированные прямоугольники, располагаются в $\mathbb{R}^3$, корректируются их положение, поворот и масштаб. Полагаем, что плоскость Oxy расположена горизонтально, ось Y направлена на наблюдателя, ось X — вправо, ось Z — вверх. Сначала определяется положение фотографии анфас. Она располагается в плоскости Oxz и поворачивается вдоль оси Y таким образом, чтобы ось симметрии ломаных Ω_f на ней совпадала с осью Z. Затем фотография в профиль располагается в плоскости Oyz и ее положение, поворот, и масштаб определяются при помощи минимизации среднеквадратичного отклонения вершин, входящих одновременно как в набор ломаных для фотографии анфас Q_f, так и в набор ломаных для фотографии анфас Q_r.

3.2 Получение формы

На этапе построения параметрической модели положение узлов ломаных Ω считается фиксированным. Цель этапа заключается в подгонке проекций ломаных, задаваемых параметрической моделью, к ломаным, полученным на предыдущем этапе. Формально она может быть сформулирована как нахождение значений параметров p, при которых функционал $\mathcal{F} = \sum_{i=1}^{M} \sum_{j=1}^{L_i} (\omega_j^i - \omega'^i_j)^2$ принимает минимальное значе-

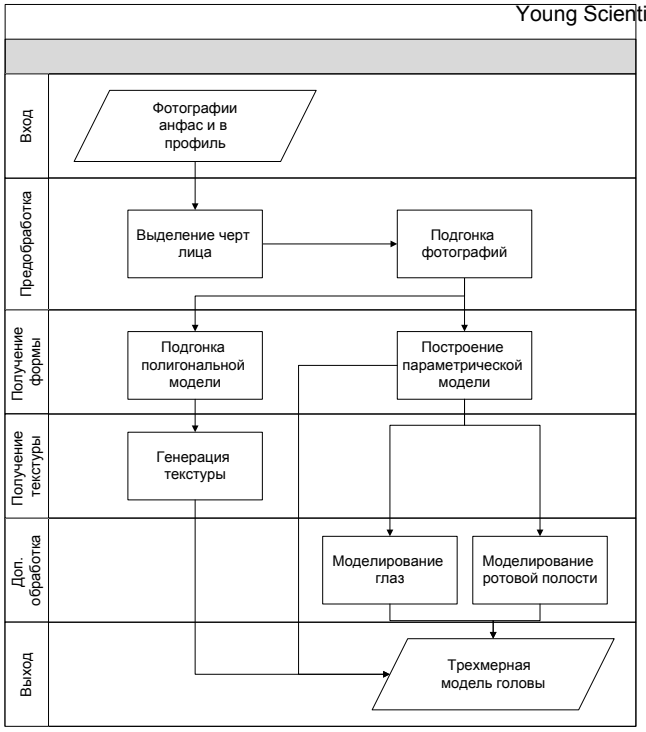


Рис. 1: *Схема работы системы моделирования головы человека по набору фотографий.*

ние. Здесь ${\omega'}^i_j$ — узлы ломаных $\omega' = \mathbf{P}Q$, а так как ломаные Q проходят по вершинам параметрической модели $a(p)$, $\forall i, j, {\omega'}^i_j = {\omega'}^i_j(p)$. Минимизация осуществляется методом градиентного спуска. В зависимости от заданной матрицы смещений S значения функционалов невязки G_m могут быть достаточно большими и получаемая полигональная модель a может быть недостаточно точной для генерации по ней тестуры. Для решения этой проблемы создается текстурированная полигональная модель $\tilde{c}$ — копия базовой текстурированной модели $\tilde{b}$. Затем ломаные на фотографиях Ω проецируются оператором $\mathbf{P}^{-1}$ из $\mathbb{R}^2$ в $\mathbb{R}^3$ (получаемые ломаные будем обозначать $\hat{Q}$): $\forall i, j, \hat{q}^i_{j,x} = \omega^i_{j,x}$, $\hat{q}^i_{j,y} = \omega^i_{j,y}$, а значение z-компоненты восстанавливается из исходной модели: $\hat{q}^i_{j,z} = q^i_{j,z}$. Далее подгонка полигональной модели $\tilde{c}$ осуществляется методом свободных деформаций Дирихле [8].

3.3 Получение текстуры

Для генерации текстуры создается текстурированная полигональная модель $\tilde{d}$. В случае фотографии анфас, значения элементов $\tilde{d}$ задаются следующим образом: $\tilde{d}_x = 0$, $\tilde{d}_y = \tilde{c}_u$, $\tilde{d}_z = \tilde{c}_v$, $\tilde{d}_u = \mathbf{P}\tilde{c}_y$, $\tilde{d}_v = \mathbf{P}\tilde{c}_z$, где $\mathbf{P}$ — оператор ортогонального проецирования. Полученная плоская полигональная модель визуализируется в текстуру (в данном случае, фронтальную). Аналогичным образом генерируется текстура по фотографии в профиль. Затем по заранее заданной маске все текстуры смешиваются.

3.4 Дополнительная обработка

Если система, в которой будет использоваться полученная модель головы, предполагает анимацию (в частности, моргание глаз и открывание рта), то исходная базовая модель головы не включает себя модели глаз, зубов, языка и полости рта, и их моделирование является дополнительной задачей. Модель каждого глаза, представляющая собой объединение двух сфер (глазное яблоко и зрачок) переносится и масштабируется в соответствии с положением ломаной $\hat{Q}$, описывающей контур данного глаза. Усредненный цвет радужки глаза определяется по фотографии анфас. Затем в пространстве HSL соответственно изменяется цвет радужки на предварительно подготовленной развертке фотографии глаза. Модели полости рта, а также языка и зубов, подгоняются аналогично моделям глаз. Текстура, полученная с фотографий, смешивается с предварительно полученной разверткой фотографий ротовой полости, включающей зубы и язык.

4 Заключение и предстоящая работа

В статье представлен разработанный метод построения параметрической модели головы человека по набору изображений. Данный подход применим к любым параметрическим моделям, обладающим описанными свойствами. В рамках дальнейшей работы планируется как улучшение предложенных алгоритмов, так и добавление новых этапов, таких как фильтрация текстуры и моделирование прически в соответствии с входными фотографиями.

Список литературы

[1] C. Xu, L. Quan, Y. Wang, T. Tan, and M. Lhuillier. Adaptive multi-resolution fitting and its application to realistic head modeling. In *Proceedings of Geometric Modeling and Processing*, 2004.

[2] Frank Hülsken, Christian Eckes, Roland Kuck, Jörg Unterberg, and Sophie Jörg. Modeling and animating virtual humans for real-time applications. In *The International Journal of Virtual Reality, 2007, 6(4):11-20*, 2007.

[3] G. Galicia and A. Zakhor. Depth based recovery of human facial features from video sequences. In *Image Processing, 1995. Proceedings., International Conference on*, volume 2, 1995.

[4] R. Lengagne, P. Fua, and O. Monga. 3D stereo reconstruction of human faces driven by differential constraints. *Image and Vision Computing*, 18(4):337–343, 2000.

[5] R. Dovgard and R. Basri. Statistical symmetric shape from shading for 3D structure recovery of faces. *Lecture Notes in Computer Science*, pages 99–113, 2004.

[6] W.A.P. Smith and E.R. Hancock. A Model-Based Method for Face Shape Recovery. In *Pattern Recognition and Image Analysis: Second Iberian Conference, IbPRIA*, pages 268–276. Springer, 2005.

[7] P. Fua. Regularized bundle-adjustment to model heads from image sequences without calibration data. *International Journal of Computer Vision*, 38(2):153–171, 2000.

[8] VG Zhislina, DV Ivanov, VF Kuriakin, VS Lempitskii, EM Martinova, KV Rodyushkin, TV Firsova, AA Khropov, and AV Shokurov. Creating and Animating Personalized Head Models from Digital Photographs and Video. *Programming and Computer Software*, 30(5):242–257, 2004.

[9] Y. Xu, C.S. Xu, Y.L. Tian, S.D. Ma, and M.L. Luo. 3D face image acquisition and reconstruction system. In *IEEE Instrumentation and Measurement Technology Conference, 1998. IMTC/98. Conference Proceedings*, volume 2, 1998.

Index of Authors

О Microsoft

Основанная в 1975 году, корпорация Microsoft является мировым лидером в производстве программного обеспечения, предоставлении услуг и разработке интернет-технологий для персональных компьютеров и серверов.

Корпорация Microsoft разрабатывает и выпускает широкий спектр программных продуктов. В их число входят настольные и сетевые операционные системы, серверные приложения для клиент-серверных сред, настольные бизнес-приложения и офисные приложения для пользователей, интерактивные программы и игры, средства для работы в сети интернет и инструменты разработки. Кроме того, Microsoft предлагает интерактивные (online) услуги, издает книги по компьютерной тематике, производит периферийное оборудование для компьютеров, занимается исследовательской деятельностью и разработкой новых компьютерных технологий. Продукты Microsoft продаются более чем в 80 странах мира, переведены более чем на 45 языков (в том числе — на русский) и совместимы с большинством платформ персональных компьютеров.

С ноября 1992 года в России действует представительство Microsoft (с июля 2004 года — ООО «Майкрософт Рус»), которое занимается продвижением программного обеспечения, развитием партнерской сети, внедрением продуктов, локализацией передовых решений и развитием рынка информационных технологий России.

Сегодня лицензионное программное обеспечение Microsoft используется на сотнях тысяч рабочих мест в России. На основе продуктов Microsoft ведутся проекты по внедрению мощных информационных систем в крупнейших отечественных коммерческих компаниях и государственных организациях.

Более подробно о примерах внедрения ПО Microsoft:

— в государственных организациях - www.microsoft.com/rus/government/

— в коммерческих компаниях - www.microsoft.com/rus/business/

Microsoft имеет все необходимые программные продукты для реализации концепции Microsoft .NET — надежные и масштабируемые серверные операционные системы и приложения Microsoft являются прекрасной платформой для предоставления интернет-сервисов и организации эффективного бизнеса.

Дополнительная информация

Дополнительную информацию о компании и продуктах Microsoft можно найти на веб-серверах Microsoft:

www.microsoft.com/rus (русскоязычная страница);
www.microsoft.com (англоязычная страница).

Для партнеров существуют специальные выделенные англо- и русскоязычные веб-ресурсы.

Контактная информация

тел.: +7 495 967-85-85
факс: +7 495 967-85-00
электронная почта: russia@microsoft.com
www.microsoft.com/rus