On viewpoint complexity of 3D scenes

Dimitri PLEMENOS[1], Mateu SBERT[2], Miquel FEIXAS[2]

[1]University of Limoges, MSI laboratory, 83, rue d'Isle, 87000 Limoges (France)

[2]Institut d'Informàtica i Aplicacions, Universitat de Girona, Campus Montilivi, Edifici PIV, E-17071 Girona

plemenos@unilim.fr, mateu@ima.udg.es, feixas@ima.udg.es

Abstract

In this paper we try to give a comprehensive definition of the notion of viewpoint complexity for 3D scenes. We show that this notion is very useful in various areas of computer graphics and that its accurate measurement permits important improvements in these areas. Methods of more or less accurate computation of viewpoint complexity for a scene are given.

Keywords: *Intrinsic scene complexity, Viewpoint scene complexity, Viewpoint entropy, Good point of view, Monte Carlo radiosity, Virtual world exploration, Image-based modelling.*

1. INTRODUCTION

Several methods, using the notion of viewpoint complexity to improve computer graphics algorithms [BDP99, BDP00a, BDP00b, Sbert02, PPV03, PPV03d, Rigau00, Ple03], have been recently developed. However, the notion of viewpoint complexity of a scene is not really well known and many authors do not always realise that, behind some more or less complex techniques, they are implicitly using the notion of viewpoint complexity. More precisely, the notion of viewpoint complexity is used, or could be used, in various areas of computer graphics, such as scene understanding, and exploration of virtual worlds, radiosity and global illumination, image-based rendering and modelling, etc.

In scene understanding and exploration of virtual worlds, viewpoint complexity is used to automatically compute interesting positions and trajectories for a camera exploring a virtual world [BDP99, BDP00a, BDP00b, Ple03, And04, PPV03e].

In radiosity, viewpoint complexity is used to improve Monte Carlo techniques by allowing a more intelligent shooting of rays from each patch of the scene [JP98, JPS99, JPS00]. It could also be used in global illumination, with photon maps-based techniques.

In image-based modelling, viewpoint complexity is used to compute an optimised minimal set of positions of the camera [PPV03d].

However, the notion of viewpoint complexity is not yet clearly enough understood. In this paper, we will try to give a more precise idea of how we perceive viewpoint complexity and its measurement, before exploring its current and future applications.

In section 2 we will try to give a definition of viewpoint complexity. In section 3 we will give some more or less accurate methods to compute viewpoint complexity. Some applications of viewpoint complexity will be described in section 4. Future issues in using viewpoint complexity will be presented in section 5 and we will conclude in section 6.

2. WHAT IS VIEWPOINT COMPLEXITY OF A SCENE

It is generally admitted that there are scenes which are considered more complex than others. The notion of complexity of a scene is an intuitive one and, very often, given two different scenes, people are able to say which scene is more complex than the other. Another problem is that it is not always clear what kind of complexity people is speaking about. Is it computational complexity, taking into account the computational cost of rendering; geometric complexity, taking into account the complexity of each element of the scene; quantitative complexity, depending on the number of elements of the scene?

We can informally define the *intrinsic complexity* of a scene as a quantity which:

1. Does not depend on the point of view;
2. Depends on:
   - The number of details of the scene.
   - The nature of details (convex or concave surfaces).

Some steps towards a formal definition of scene complexity had been presented in [Feixas02, Feixas99]. Unlike intrinsic complexity, the *viewpoint complexity* of a scene depends on the point of view. An *intrinsically complex* scene, seen from a particular point of view, is not necessarily viewpoint complex. A first measure of the notion of viewpoint complexity of a scene from a point of view could be the number of visible details or, more precisely, the number of surfaces of the scene visible from this point of view. However, this definition of viewpoint complexity is not very satisfactory because the size of visible details is also important. Finally, we will define the viewpoint complexity of a scene from a given point of view as a quantity which depends on:

- The number of surfaces visible from the point of view.
- The area of visible part of each surface of the scene from the point of view.
- The orientation of each (partially) visible surface according to the point of view.
- The distance of each (partially) visible surface from the point of view.

An intuitive idea of viewpoint complexity of a scene is given in figure 1, where the viewpoint complexity of scene (a) is less than the viewpoint complexity of scene (b) and the viewpoint complexity of scene (b) is less than the viewpoint complexity of scene (c), even if each scene contains other non visible surfaces.

Given a point of view, a scene may be divided in several more or less viewpoint complex regions from this point. The viewpoint complexity of a (part or region of) scene from a point of view, as defined above, is mainly geometry-based as the elements taken into account are geometric elements. It would be possible to take into account other aspects of the scene, such as lighting, because lighting may modify the perception of a scene by a human user. However, only geometrical viewpoint complexity will be discussed in this paper, as the study on influence of lighting in scene perception is not yet enough advanced.

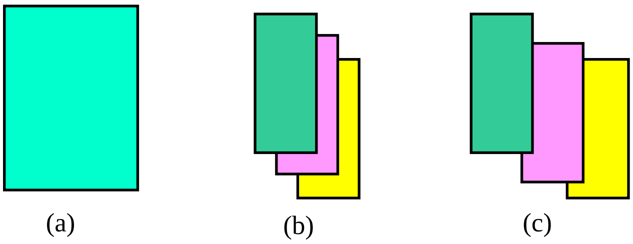

Figure 1: Viewpoint complexity of scene (c) is more than viewpoint complexity of scene (b); viewpoint complexity of scene (b) is more than viewpoint complexity of scene (a).

3. HOW TO COMPUTE VIEWPOINT COMPLEXITY

Following our definition of viewpoint complexity in section 2, its calculation depends on the number of visible surfaces, the area of the visible part of each surface and the distance and orientation of each (partially) visible surface, according to the point of view. A linear combination of these two quantities would give an accurate enough measure of viewpoint complexity. The most important problem is the way to compute the number of visible surfaces and the visible projected area of each surface. The used method may depend on some constraints of the application using this information. Some applications require real time calculation whereas for others the calculation time is not an important constraint. For some applications, it is very important to have accurate viewpoint complexity estimation and for others a fast approximate estimation is sufficient.

It is easy to see that the visible part, orientation and distance of (partially) visible surfaces from the point of view can be accurately approximated by the projection of the visible parts of the scene on the unitary sphere centred on the point of view. This approximation will be used in this section to estimate the viewpoint complexity of a scene from a point of view.

3.1 Accurate viewpoint complexity estimation

The most accurate estimation of the viewpoint complexity of a scene can be obtained by using a hidden surface removal algorithm, working in the user space and explicitly computing the visible part of each surface of the scene. Unfortunately, it is rarely possible in practice to use such an algorithm because of the computational complexity of this kind of algorithms. For this reason, less accurate but also less complex methods have to be used.

A method proposed in [BDP99, Ple03] permits to use hardware accelerated techniques in order to decrease the time complexity of estimation. This method uses image analysis to reduce the computation cost. Based on the use of the OpenGL graphical library and its integrated z-buffer, the technique used is the following.

If a distinct colour is given to each surface of the scene, displaying the scene using OpenGL allows to obtain a histogram (figure 2) which gives information on the number of displayed colours and the ratio of the image space occupied by each colour.

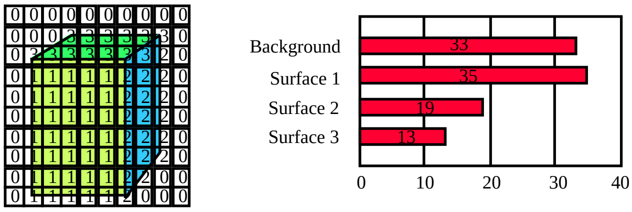

Figure 2: Fast computation of number of visible surfaces and area of projected viewpoint part of the scene by image analysis.

As each surface has a distinct colour, the number of displayed colours is the number of visible surfaces of the scene from the current position of the camera. The ratio of the image space occupied by a colour is the area of the projection of the viewpoint part of the corresponding surface. The sum of these ratios is the projected area of the visible part of the scene. With this technique, the two viewpoint complexity criteria are computed directly by means of an integrated fast display method.

The viewpoint complexity of a scene from a given viewpoint can now be computed by a formula like the following one:

$$C(V) = \frac{\sum_{i=1}^{n} \left[\frac{P_i(V)}{P_i(V)+1} \right]}{n} + \frac{\sum_{i=1}^{n} P_i(V)}{r}$$

where: C(V) is the viewpoint complexity of the scene from the view point V,

$P_i(V)$ is the number of pixels corresponding to the polygon number i in the image obtained from the view point V,

r is the total number of pixels of the image (resolution of the image),

n is the total number of polygons of the scene.

In this formula, [a] denotes the smallest integer, greater than or equal to a.

Another method to compute viewpoint complexity has been proposed in [PPV01, Sbert02, Rigau00, PPV03], based on information theory. In this method, the viewpoint complexity of a scene from a given point of view is approached by computing the viewpoint entropy. The viewpoint entropy is given by the formula:

$$H(S,P) = -\sum_{i=0}^{N_f} \frac{A_i}{A_t} \log \frac{A_i}{A_t}$$

where P is the point of view, Nf is the number of faces of the scene S, A_i is the projected area of the face i and A_t is the total area covered over a sphere centred on the point of view.

Both methods have to compute the number of visible surfaces and the visible projected areas by using the technique described above.

3.2 Fast approximated estimation of viewpoint complexity

In some cases, accurate viewpoint complexity estimation is not requested, either because of need of real time estimation of the viewpoint complexity or because a less accurate estimation is enough for the application using the viewpoint complexity.

In such a case, it is possible to roughly estimate the viewpoint complexity of a scene from a given point of view, as follows:

A more or less great number of rays are randomly shot from the point of view to the scene and intersections with the surfaces of the scene are computed. Only intersections with the closest to the point of view surfaces are retained (Figure 3).

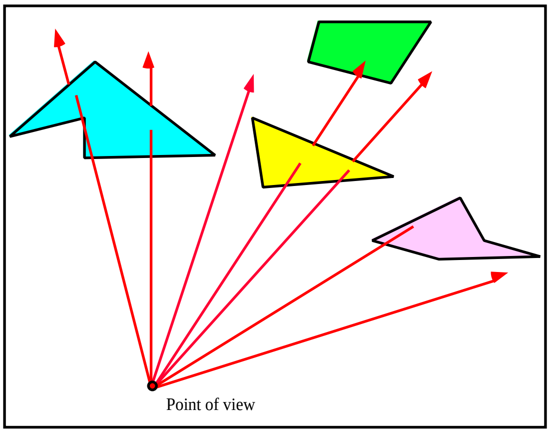

Figure 3: Approximated estimation of viewpoint complexity

Now, we can approximate the quantities used in viewpoint complexity calculation. We need first to define the notion of visible intersection. A *visible intersection* for a ray is the closest to the point of view intersection of the ray with the surfaces of the scene.

- *Number of visible surfaces* = number of surfaces containing at least one visible intersection with a ray shot from the point of view.
- *Visible projected area of a surface* = number of visible intersections on the surface.
- *Total visible projected area* = number of visible intersections on the surfaces of the scene.
- *Total projected area* = total number of rays shot.

The main interest of this method is that the user can choose the degree of accuracy, which depends on the number of rays shot.

4. APPLICATIONS

As it has been written above, many works, in various areas of computer graphics, are, directly or indirectly, based on the notion of viewpoint complexity. Generally, in these works, elements of viewpoint complexity are used together with other notions and it is difficult to estimate the pertinence of a method using together dissimilar elements. As the notion of viewpoint complexity is now precisely defined in the above sections, it is possible to present applications where the contribution of viewpoint complexity is clearly delimited.

4.1 Scene understanding

Very often, the rendering of a scene, obtained after a long computation time, is not possible to exploit because the choice of the angle of view was bad. In such a case, the only possibility is to choose another angle of view and to try again by running the time consuming rendering algorithm once again. We think that it is very difficult to find a good angle of view for a 3D scene when the working interface is a 2D screen. On the other hand, we think that the choice of an angle of view is very important for understanding a scene.

As the program (modeller, renderer) has a full knowledge of the geometry of the world to visualise, we thought that it could be more interesting to ask the program to find a good angle of view for this world. The problem is to know what is a good angle of view.

4.1.1 Static scene understanding

When a scene is not very complex, a single well chosen point of view may be sufficient to well understand the scene. For this reason, algorithms have been developed in order to automatically compute a good point of view, allowing to well understand a scene.

In [KK88] a method to compute a good point of view is proposed. In this method, viewpoint complexity is not really used. The method only guarantees that from the computed point of view the user sees a minimum of degenerated edges of the scene.

In [Col88] some elements of viewpoint complexity are indirectly used. A good point of view is computed by an approximated method from three selected points of view. This method has been tested with octree-based scenes.

In another method [PB96, BDP99, Ple03], two main notions are used: viewpoint complexity and heuristic solution search. The method can be resumed as follows:

1. The points of view are supposed to be on the surface of a virtual sphere with the scene at the center. The surface of the sphere of points of view is divided in 8 spherical triangles (Figure 4).

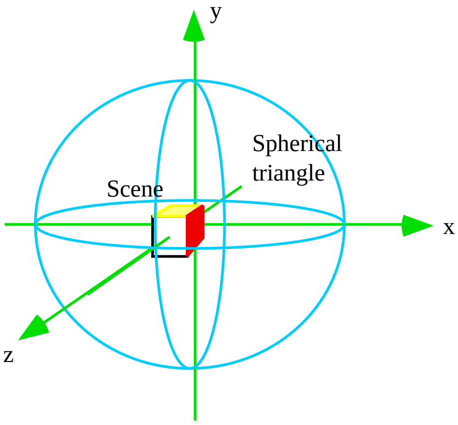

Figure 4: sphere of viewpoints divided in 8 spherical triangles

2. The best spherical triangle is determined by positioning the camera at each intersection point of the three main axes with the sphere and computing the viewpoint complexity of the scene from this point of view. The three intersection points with the best evaluation are selected. These three points on the sphere give the best spherical triangle, selected as the best one.
3. Now, the selected spherical triangle ABC of Figure 5 is processed. If the scene is viewpoint more complex from the vertex A than from vertices B or C, two new vertices E

and F are chosen at the middles of the edges AB and AC respectively and the new spherical triangle ADE becomes the current spherical triangle. This process is recursively repeated until the viewpoint complexity of the scene from the obtained points of view does not increase. The vertex of the final spherical triangle corresponding to the viewpoint more complex scene is chosen as the best point of view.

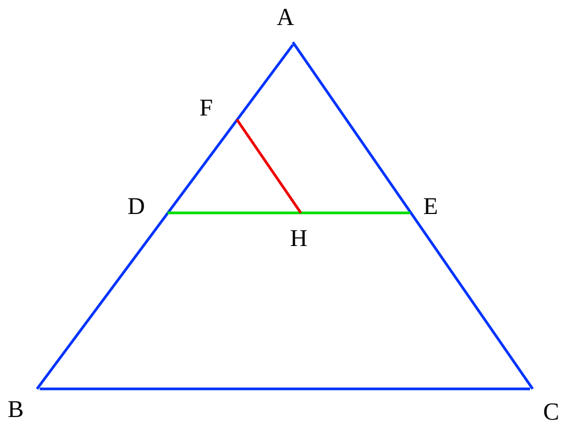

Figure 5: Heuristic search of the best point of view by subdivision of a spherical triangle

4.1.1 Dynamic scene understanding

For very complex scenes, a single point of view is generally not enough to well understand them. Moreover, even if several points of view are computed, it is not easy to understand a scene, seen from various points of view, without knowing how to reach the current point of view from the previous one. The best solution, in such a case, is to give the user the possibility to ask an automatic exploration of the scene by a virtual camera. This camera will be supposed to move on the surface of points of view and exploration will be based on incremental evaluation of the viewpoint complexity of the scene from the next possible point of view. However, the viewpoint complexity of the scene from the next candidate point of view is not enough to ensure intelligent computation of the camera path. The movement of the camera must obey to the following rules:

- It is important that the camera moves on positions which are good points of view.
- The camera must avoid fast returns to the starting point or to already visited points.
- The camera's path must be as smooth as possible in order to allow the user to well understand the explored world. A movement with brusque changes of direction is confusing for the user and must be avoided.

In order to apply these heuristic rules, the next position of the camera is computed in the following way:

- The best point of view is chosen as the starting position for exploration.
- Given the current position and the current direction of the camera (the vector from the previous to the current position), only directions insuring smooth movement are considered in computing the next position of the camera (figure 6).

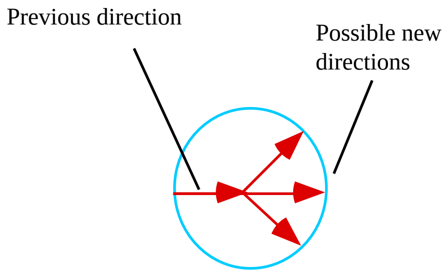

Figure 6: Only 3 directions are considered for a smooth movement of the camera

- In order to avoid fast returns of the camera to the starting position, the importance of the distance of the camera from the starting position must be inversely proportional to path of the camera from the starting to the current position (figure 7).

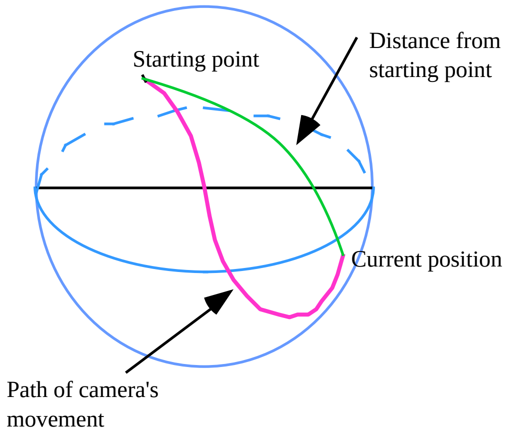

Figure 7: Distance of the current position of the camera from the starting point.

Thus, the following evaluation function is used to evaluate the next position of the camera on the surface of the sphere:

$$w_c = \frac{v_c}{2}\left(1+\frac{d_c}{p_c}\right)$$

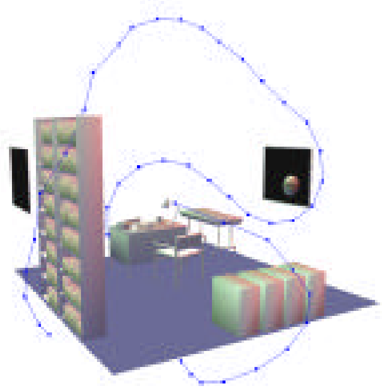

Figure 8: Exploration of a virtual office by incremental outside exploration.

In this formula:

- w_c is the weight of the current camera position,
- v_c is the viewpoint complexity of the scene from the camera's current position,

- p_c is the path traced by the camera from the starting point to the current position,
- d_c is the distance of the current position from the starting point.

Several variants of this technique have been proposed and applied. In figure8 one can see an example of exploration of a simple virtual world representing an office.

In [PPV03, PPV03d, And04], viewpoint entropy has also been used both for static and dynamic scene understanding.

4.2 Radiosity and global illumination

Viewpoint complexity can also be used in radiosity, in order to improve Monte Carlo-based computation. To do this, we are going to use viewpoint complexity of a region of a scene from a given point of view.

4.2.1 Sampling problems in Monte Carlo radiosity

The Monte Carlo based radiosity computation is an elegant manner to compute the radiosity of a scene without having to explicitly compute form factors. However the Monte Carlo sampling, using rays to distribute the energy of each patch, is not entirely satisfactory because, on average, the same number of rays is shot to all parts of the scene from a given patch. This sampling problem of the Monte Carlo radiosity may produce noisy images, especially in the case where the scene contains both simple and complex parts. In order to obtain a better distribution of the rays, it is necessary to have a mean to recognise the complexity of a region in a scene.

4.2.2 Improvement of Monte Carlo radiosity by using viewpoint complexity

In this section, a method of ray distribution, taking into account the viewpoint complexity of a region of a scene, will be described. This method is based on heuristic search in the different regions of the scene, from each patch.

The principle of the proposed method is the following:

For each patch of the scene,

- divide the remaining of the scene in regions,
- estimate the *viewpoint complexity* of each region from the patch,
- distribute a part of the rays leaving the patch, according to the estimated *viewpoint complexity* of each region, until the whole energy of each path is distributed.

The proposed method involves the following steps:

1. A hemisphere, divided in 4 spherical triangles, is associated with each patch of the scene (see figure 9).

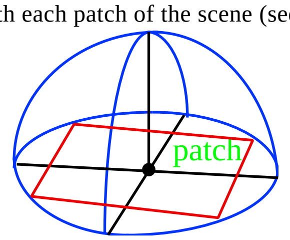

Figure 9: Initial subdivision of a hemisphere associated with a patch

2. At any phase of the process, all spherical triangles of the hemisphere associated with each patch are processed independently of each other. Starting from the current situation, the viewpoint complexity of the region of the scene contained in the pyramid defined by the centre of the patch and the current spherical triangle is measured and one of the following actions is done, according to the value of the viewpoint complexity of the region:
   2.1 If the viewpoint complexity of the region delimited by the centre of the patch and the current spherical triangle is greater than a threshold value, or if the maximum number of subdivisions fixed by the user is not reached, the spherical triangle is divided in 4 new spherical triangles (see figure 10) and the heuristic search starts again with each of the 4 new spherical triangles.

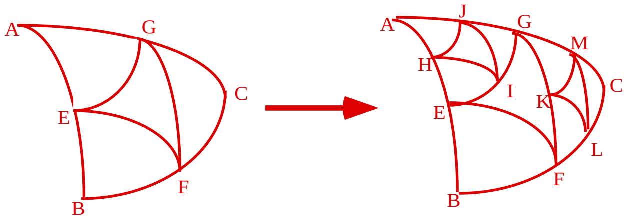

Figure 10: The spherical triangles AEG and CGF are divided in 4 new spherical triangles

   2.2 Otherwise, the heuristic search process is finished for the current spherical triangle of the hemisphere associated with the current patch.

   The subdivision process is finished when no more subdivision is possible for any patch. The scene is now divided in regions, each region being delimited by the pyramid defined by the centre of the patch and one of the obtained spherical triangles.

The second problem to resolve is the problem of distributing the rays shot from a patch according to the viewpoint complexity of each region. In the ideal case, all spherical triangles of the hemisphere associated with a patch define regions with equal complexities. Unfortunately, it is generally not true because the cost of the region subdivision process greatly increases with each new subdivision level and must be stopped at a small number of levels (typically 3 or 4). Thus, the ray distribution method must take into account the viewpoint complexity of each region. Unlike with the classical Monte Carlo sampling, where each ray contains the same amount of energy, the relationship between rays and transported energy is more complex in our case. The following rules are applied to determine the number of rays and the transported energy by a ray:

1. The number of rays shot in a region is proportional to the viewpoint complexity of the region.
2. The amount of energy distributed in a region from a path is proportional to the area of the corresponding spherical triangle, according to the cosine of its average direction with the normal of the patch.
3. The amount of energy distributed by a ray in a region is proportional to the cosine of its direction with the normal of the patch.

Figures 11 and 12 illustrate improvements obtained with a viewpoint complexity-based sampling, compared to classical Monte Carlo sampling. Convergence with the new sampling is much faster. The results shown in figure 12 have been obtained by an approximated estimation of the viewpoint

complexity of a region from a patch.
Roughly estimated region complexity is used in order to have a fast estimation and not to penalise the radiosity algorithm. However, we think that it is possible to have a fast, more accurate estimation of the viewpoint complexity of a region from a patch, based on the method used in subsection 3.1 for fast accurate viewpoint complexity estimation. The method includes the following steps:

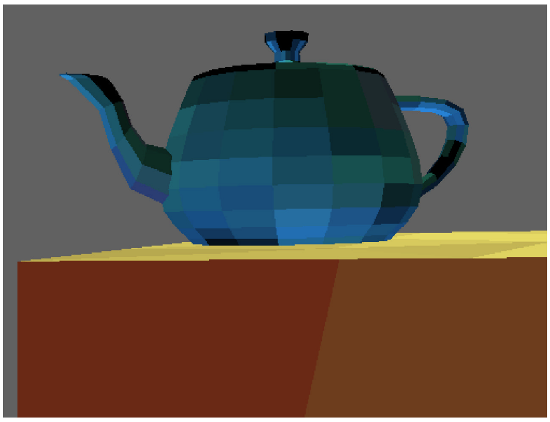

Figure 11: Detail of a reference scene obtained with classical Monte Carlo sampling at the end of the first step

1. Give a different colour to each polygon of the scene.
2. Display the scene using the integrated z-buffer of OpenGL.
3. Estimate the viewpoint complexity of the region, using the part of the image covered by the projection of the current spherical triangle (figure 13).
4. Subdivide, according on the estimated viewpoint complexity.

For a given patch and a given spherical triangle, the displaying of the scene is made only once, while several several viewpoint complexity estimations can be performed by clipping the image by the projection of the current spherical triangle, obtained by the last subdivision.

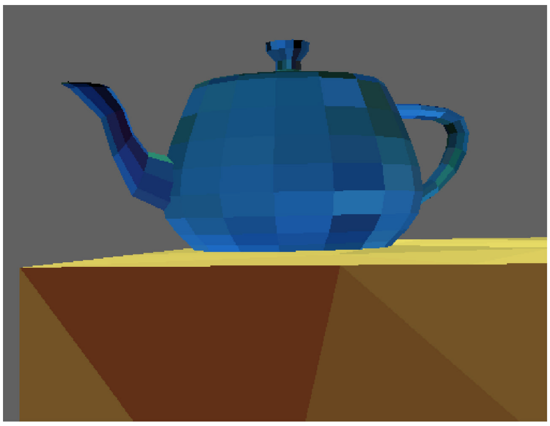

Figure 12: Detail of a reference scene obtained with heuristic search based sampling at the end of the first step

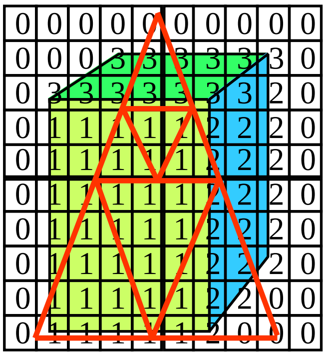

Figure 13: accurate estimation of the viewpoint complexity of a region

4.3 Image-based modelling and rendering

In image-based modelling, it is important to compute a minimum set of points of view in order to use them in obtaining the set of images which will replace the scene display. A method to do this, based on viewpoint complexity, is the following:

1. Compute a sufficient number of points of view, according to the viewpoint complexity of the scene from a region of the sphere of points of view. A region is represented by a spherical triangle. The viewpoint complexity of a scene from a region may be defined as the average value of viewpoint complexities from the three vertices of the spherical triangle.
   - If the scene is viewpoint complex from the region, subdivide the region in 4 subregions.
   - Otherwise, add the centre of the region to the current list of point of view.
2. Using an evaluation function for a set of points of view, try to eliminate elements of the list of points of view in several steps, by considering each point of view as a candidate to be eliminated. A point of view is eliminated if its contribution is empty (or quasi-empty) to the result of evaluation of the remaining point of view.
3. If, at the end of a step, no one point of view is eliminated, the process is finished and the current list of point of view contains a minimum set of points of view.

In Figure 14 we see the reconstructed images using a classic approach (top) and a best view point approach (bottom) with entropy as the measure of viewpoint complexity [PPV03d].

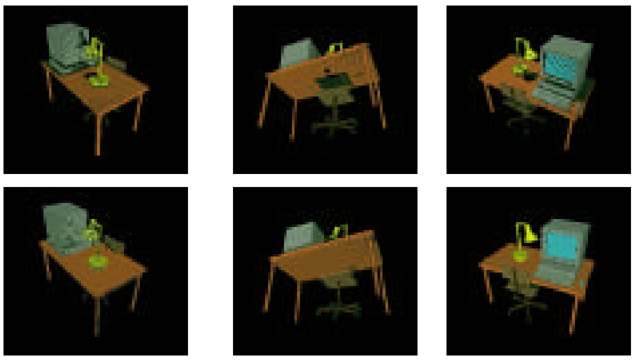

Figure 14: Classic vs. best view entropy based capture for IBR representation

4.4 Application to Ray-Tracing

Obtaining a good quality image with ray-tracing demands to cast a lot of rays through each pixel of the screen plane. However, not all pixels need this amount of supersampling. An homogeneous region will need less rays than a region with geometrical discontinuities and/or high illumination gradients. Viewpoint complexity, when restricted to a square area of pixels, to a pixel or subpixel area, can give a measure of the additional sampling necessity, and thus it can be the base for adaptive sampling methods. This has been explored in [Ple87] and in [Rigau02a,b] where entropy is used as the viewpoint complexity measure. In Figure 15 right we can see the improvement vs. uniform sampling, shown in Fig.15 left.

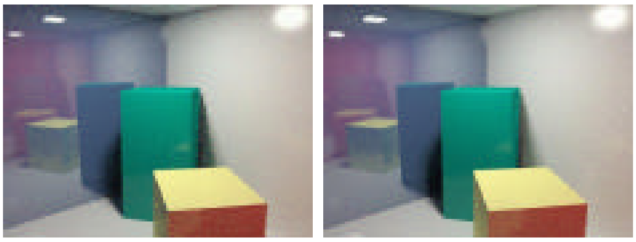

Figure 15: Uniform sampling (left) vs. Entropy based adaptive sampling (right).

4.5 Molecular visualisation

Visualization of molecules is relevant for molecular science, a discipline which falls in several areas such as Crystalography, Chemistry and Biology. Two kinds of views are important for scientists, for a set of molecules low entropy views (that is, low viewpoint complexity) and for a single molecule views with high entropy (high viewpoint complexity) [PPV02]. In the first case the views allow to see how the molecules arranges in space and thus infer physical properties. The second case shows how the atoms are arranged in a molecule and allows to infer its chemical properties. In Figure 16 we see two elements of Carbon: graphite and diamond. From these views molecular scientists can infer the resistance to physical pressure. While diamond is very strong in three directions, the layered structure of graphite makes it easily exfoliable.

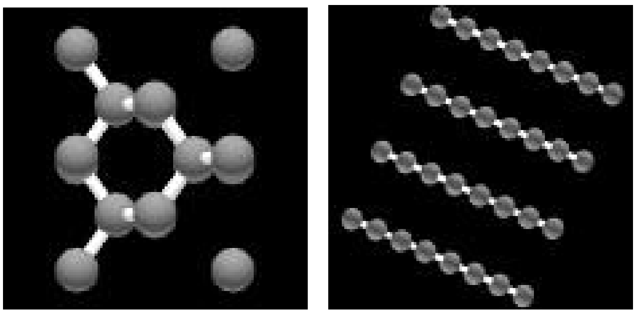

Figure 16: Minimum entropy views for graphite and diamond.

5. FUTURE ISSUES

Scene understanding and virtual worlds exploration can be improved by combining a separated computation of viewpoint complexity and semantic information on the various parts of the virtual world to explore. As the viewpoint complexity is well defined and can be computed independently, its combination with semantic knowledge should permit more intelligent exploration.

Viewpoint selection using viewpoint complexity can also play an important role in data visualization. When complex data need to be shown and/or interpreted, the automatic selection of views can make the process easier. In this sense, molecular visualization shown in section 4.3. can be seen as a first step in this direction.

Another application area which is worth investigating is protein docking [Stern98,Vak95]. A protein could move in order to see the other one from the most appropriate viewpoint for docking.

6. CONCLUSION

In this paper, we have tried to give an informal but complete definition of the viewpoint complexity concept. We have then presented methods to estimate, more or less accurately, the viewpoint complexity of a scene from a given point of view. Finally, various applications of this concept have been presented, permitting to improve classical computer graphics techniques.

This work reviews the first steps made towards a definition of viewpoint complexity. It also permits to see how viewpoint complexity is important in several areas of computer graphics and how an accurate definition of this concept is useful in order to well separate its computation from computation of other concepts and to use it as a parameter, with a pertinent weight, to improve a lot of computer graphics areas.

ACKNOWLEDGMENTS

We would like to thank Vincent Jolivet, Guillaume Dorme, Pere P. Vázquez and Jaume Rigau for providing discussion and images. This project has been funded in part with grant numbers TIC-2001-2416-C03-01 of Spanish Ministry of Science and Technology, and 2001-SGR-00296 and ACI2002-29 of the Catalan Government. It has also been partly supported and financed by the Limousin Region (France).

REFERENCES

[And04] Carlos Andújar, Pere Pau Vázquez, Marta Fairén, Way-Finder: guided tours through complex walthrough models, Computer Graphics Forum (Eurographics 2004), 2004.

[BDP00a] P. Barral, G. Dorme, D. Plemenos. Intelligent scene exploration with a camera. International Conference 3IA'2002, Limoges (France), May 3-4, 2000.

[BDP00b] P. Barral, G. Dorme, D. Plemenos. Scene understanding techniques using a virtual camera. Eurographics 2000, Interlagen (Switzerland), August 20-25, 2000, Short papers proceedings.

[BDP99] P. Barral, G. Dorme, D. Plemenos. Visual understanding of a scene by automatic movement of a camera. International Conference GraphiCon'99, Moscow (Russia), August 26 – September 3, 1999.

[Col88] C. Colin. A System for Exploring the Universe of Polyhedral Shapes., Eurographics'88, Nice (France), September 1988.

[Feixas02] Miquel Feixas, An Information Theory Framework for the Study of the Complexity of Visibility and Radiosity in a Scene. PhD thesis, Technical University of Catalonia, 2002

[Feixas99] M.Feixas, E.Acebo, Philippe Bekaert and M.Sbert, An information theory framework for the analysis of scene complexity, Eurographics'99

[FP93] M. Feda, W. Purgathofer, Progressive Ray Refinement for Monte Carlo Radiosity. Fourth Eurographics Workshop on Rendering, Conf. Proc. (June 1993), pp. 15-25.

[JP98] V. Jolivet, D. Plemenos, A new hemisphere subdivision technique for computing radiosity. International Conference GraphiCon'98, Moscow (Russia), 7-12 of September 1998.

[JPS00] V.Jolivet, D.Plemenos, M.Sbert. Pyramidal Hemisphere Subdivision Radiosity. Definition and improvements. . International conference WSCG'2000, Plzen (Czech Republic), February 7-11, 2000.

[JPS99] V.Jolivet, D.Plemenos, M.Sbert, A pyramidal hemisphere subdivision method for Monte Carlo radiosity, Eurographics 99 Short Papers proceedings.

[KK88] T. Kamada, S. Kawai. A Simple Method for Computing General Position in Displaying Three-Dimensional Objects. Computer Vision, Graphics and Image Processing, vol. 41, 1988.

[PB96] D. Plemenos, M. Benayada. Intelligent Display Techniques in Scene Modelling. New Techniques to Automatically Compute Good Views. International Conference GraphiCon'96, St Petersburg (Russia), 1-5 of July 1996.

[Ple03], D. Plemenos. Exploring Virtual Worlds: Current Techniques and Future Issues. International Conference GraphiCon'2003, Moscow (Russia), September 5-10, 2003.

[Ple87] D. Plemenos. Selective refinement techniques for realistic rendering of 3D scenes. International Journal of CAD and Computer Graphics, vol. 1, no 4, 1987, in French.

[PPV01] P.P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich. Viewpoint Selection Using Viewpoint Entropy. Vision, Modeling, and Visualization 2001 (Stuttgart, Germany), pp. 273-280, 2001.

[PPV02] P.P. Vázquez, M. Feixas, M. Sbert, and A. Llobet. Viewpoint Entropy: A New Tool for Obtaining Good Views for Molecules. VisSym '02 (Eurographics - IEEE TCVG Symposium on Visualization) (Barcelona, Spain), 2002.

[PPV03] Pere Pau Vázquez, PhD thesis, On the Selection of Good Views and its Application to Computer Graphics. Technical University of Catalonia, 2003.

[PPV03b] Pere-Pau Vázquez and Mateu Sbert. Fast adaptive selection of best views. Lecture Notes in Computer Science, 2003 (Proc. of ICCSA'2003).

[PPV03c] Pere-Pau Vázquez and Mateu Sbert. Perception-based illumination information measurement and light source placement. Lecture Notes in Computer Science, 2003 (Proc. of ICCSA'2003).

[PPV03d] P.P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich. Automatic View Selection Using Viewpoint Entropy and its Application to Image-Based Modeling. Computer Graphics Forum, desember-2003.

[PPV03e] Pere-Pau Vázquez and Mateu Sbert. Automatic indoor scene exploration. In International Conference on Artificial Intelligence and Computer Graphics, 3IA, Limoges, may 2003.

[Rigau00] J. Rigau, M. Feixas, and M. Sbert. Information Theory Point Measures in a Scene. IIiA-00-08-RR, Institut d'Informàtica i Aplicacions, Universitat de Girona (Girona, Spain), 2000.

[Rigau02a] J. Rigau, M. Feixas, and M. Sbert. New Contrast Measures for Pixel Supersampling. Advances in Modeling, Animation and Rendering. Proceedings of CGI'02 (Bradford, UK), pp. 439-451, 2002. Springer-Verlag London Limited, London, UK.

[Rigau02b] J. Rigau, M. Feixas, and M. Sbert. Entropy-Based Adaptive Sampling. Graphics Interface 2003 (Halifax, Canada), june-2003.

[Sbert02] M. Sbert, M. Feixas, J. Rigau, F. Castro, and P.P. Vázquez. Applications of Information Theory to Computer Graphics. Proceedings of 5th International Conference on Computer Graphics and Artificial Intelligence, 3IA'02 (Limoges, France), pp. 21-36, 2002.

[Stern98] Sternberg, M.J.E., Gabb, H.A., and Jackson, R.M. 1998. Predictive docking of protein–protein and protein–DNA complexes. Curr. Opin. Struct. Biol. 8:250–256.

[Vak95]Vakser, I.A. 1995. Protein docking for low-resolution structures. Protein Eng. 8: 371–377.

About the authors

Dimitri PLEMENOS is a full professor at the University of Limoges (France) and director of the MSI research laboratory. His research area is intelligent techniques in computer graphics, including Declarative Modelling, Intelligent Rendering and Intelligent Virtual World Exloration. He is author or co-author of several papers and member of the IPC of many international conferences and journals. Dimitri PLEMENOS is the organiser and general chair of the 3IA international conference on Computer Graphics and Artificial Intelligence.

Mateu Sbert is an associate professor in Computer Science at the University of Girona. He received a M.Sc. in Theoretical Physics (1977) at the university of Valencia, a M.Sc. in Mathematics (Statistics and Operations Research, 1983) at U.N.E.D. university (Madrid) and a Ph.D. in Computer Science at the U.P.C. (Universitat Politècnica de Catalunya, 1997, Best PhD Award). Mateu Sbert's research interests include the application of Monte Carlo, Integral Geometry and Information Theory techniques to Radiosity, Global Illumination and Image Based Rendering. He has authored or co-authored about 80 papers in his areas of research and served as a member of program comittee in several spanish and international conferences.

Miquel Feixas is an associate professor in Computer Science at the Universitat de Girona. He received a M.Sc. in Theoretical Physics (1979) at the UAB (Universitat Autònoma de Barcelona) and a Ph.D. in Computer Science at the UPC (Universitat Politècnica de Catalunya, 2002).
His research interests include the application of Information Theory techniques to Radiosity, Global Illumination and Image Processing. He has authored or co-authored 25 papers in his area of research. He has participated in several Spanish research projects and joint actions with several European universities.