

The 23<sup>rd</sup> International Conference  
on Computer Graphics and Vision

**GraphiCon'2013**

---

September 16–20, 2013  
Vladivostok, Russia

**Conference Proceedings**

23-я Международная Конференция  
по Компьютерной Графике и Зрению

**ГрафиКон'2013**

16–20 сентября, 2013  
Владивосток, Россия

**Труды Конференции**

Институт автоматизации и процессов управления ДВО РАН  
Дальневосточный федеральный университет

## Committee

**Conference Chair:** Sergienko Valentin (FEB RAS)

**Conference Vice-chair:** Bobkov Valery (Institute of Automation and Control Processes FEB RAS)

### Organizing Committee Members:

Bayakovskiy Yuri (Lomonosov Moscow State University)

Galaktionov Vladimir (Keldysh Institute of Applied Mathematics RAS)

Debelov Victor (Institute of Computational Mathematics and Mathematical Geophysics SB RAS)

Klimenko Stanislav (Moscow Institute of Physics and Technology(State University))

Krylov Andrey (Lomonosov Moscow State University)

Paltashev Timur (Advanced Micro Devices, U.S.A.)

Sherbatuk Alexandr (Institute of Marine Technology Problems FEB RAS)

### International Program Committee Chair

Lei Xu (Chinese University Hong Kong , China)

Tom Sederberg (Brigham Young University , USA)

Bobkov Valery (Institute of Automation and Control Processes FEB RAS, Russia)

### International Program Committee

Valery Adzhiev (UK)

Boris Alpatov (Russia)

Mark Bannatyne (USA)

Gladimir Baranoski (Canada)

Boris Barladyan (Russia)

Natalia Baygarova (Russia)

Alexander Belyaev (UK)

Valery Bobkov (Russia)

Andrey Boguslavskiy (Russia)

Alexander Bondarev (Russia)

Alex Bovyrin (Russia)

Aleksey Chernavskiy (Russia)

Dmitry Chetverikov (Hungary)

Sabine Coquillart (France)

Marc Daniel (France)

Victor Debelov (Russia)

Viktor Eruhimov (Russia)

Victor Gaganov (Russia)

Alexander Goncharov (Russia)

Vladimir Grishin (Russia)

Andres Iglesias (Spain)

Alexey Ignatenko (Russia)

Vladimir Kaleyda (Russia)

Nikolay Kim (Russia)

Stanislav Klimenko (Russia)

Anton Konushin (Russia)

Andrey Kopylov (Russia)

Vassili Kovalev (Belarus)

Andrey Krylov (Russia)

Victor Lempitsky (UK)

Alexey Lukin (Russia)

Leonid Mestetskiy (Russia)

Maxim Mizotin (Russia)

Vadim Mottl (Russia)

Karol Myszkowski (Germany)

Vahagn Nahapetyan (Russia)

Gregory Nielson (USA)

Alexander Pasko (Russia)

Victor Pilyugin (Russia)

Alexander Reshetov (Russia)

Michael Rychagov (Russia)

Filip Sadlo (Germany)

Ilya Safonov (Russia)

Vladimir Savchenko (Japan)

Yann Savoye (Portugal)

Tom Sederberg (USA)

Olga Senyukova (Russia)

Nadegda Shchegoleva (Russia)

Sergey Sokolov (Russia)

Igor Solovyev (Russia)

Daniel Thalmann (Switzerland)

Vadim Turlapov (Russia)

Sergey Umnyashin (Russia)

Natalia Vassilieva (Russia)

Dmitriy Vatolin (Russia)

Alexander Velizhev (Switzerland)

Dmitriy Vetrov (Russia)

Yuri Vizilter (Russia)

Aleksey Voloboy (Russia)

Konstantin Vostryakov (Russia)

Dmitry Yurin (Russia)

Evgeniy Zhilyakov (Russia)

Xinbo Gao (China)

Changyin Sun (China)

Yanning Zhang (China)

Jian-huang Lai (China)

Jieyu Zhao (China)

Tang Xianglong (China)

Zhi-Yong Liu (China)

Xuelei Hu (China)

### Address for Communication/Conference Secretariat

GraphiCon'2013, Computer Graphics Lab, 4<sup>th</sup> floor, 408 room, Radio 5 street, IACP FEB RAS, Vladivostok, Russia, 690041

Tel: +7 (423) 231-37-76, E-mail: [2013@graphicon.ru](mailto:2013@graphicon.ru) (or [contact from the website](#)), Conference Web Site: [2013.graphicon.ru](http://2013.graphicon.ru)



**Organized by Graphics Lab IACP FEB RAS**



**Far Eastern Federal University**



**Graphic and Media Lab (Lomonosov Moscow State University)**



**In cooperation with M.V.Keldysh Institute for Applied Mathematics of Russian Academy of Science**



**Supported by Russian Academy of Science**



**Sponsored by Russian Foundation for Basic Research**



**Sponsored by C3D Labs (ASCON Group)**

# Contents

Preface .....	10
---------------	----

## Invited Talks

---

A5 Problem Solving Formulations for Object Detection, Discriminant Analysis, and Integrative Hypothesis Testing .....	12
<i>Lei Xu</i>	

Актуальные информационные технологии: визуализация информации, виртуальное окружение, осязаемые изображения, неогеография .....	13
<i>Станислав Клименко</i>	

## Technical section (English)

---

### EN1: Vision

A self-calibration method for a multi stereo-rig system based on robust 3D motion estimation .....	20
<i>Oleg Stepanenko</i>	
A Minimal Six-Point Auto-Calibration Algorithm .....	24
<i>Eugeniy Martyushev</i>	
A Novel Approach for Attention Points Extraction from Saliency Maps based on T-Junctions .....	28
<i>Ekaterina Potapova, Michael Zillich, Markus Vincze</i>	
Dependency of detectors' and descriptors' efficiency on image resolution for SIFT, SURF and ORB ....	32
<i>Victor Eruhimov, Anna Kogan</i>	
"Particle" Approach for Flame Front Propagation Fast Modeling in the Given Premixed Gas Flow Field	36
<i>Sergey Minaev, Boris Dolgovesov, Boris Mazurok</i>	
Improved analysis of membrane potential oscillations in the network of cells from islet of Langerhans .....	39
<i>Denis Špelič, Jurij Dolensšek, Andraž Stožer, Marjan Slak Rupnik, Borut Žalik, Domen Mongus</i>	

### EN2: Graphics

<STAR> Human Body Shape Modeling Using 3D Range Data for Healthcare Applications and Beauty Services .....	43
<i>Igor Goncharenko</i>	
<STAR> Heterogeneous System Architecture (HSA): Software Ecosystem for CPU/GPU/DSP and other accelerators .....	48
<i>Timour Paltashev</i>	
<STAR> Rendering of transparent optically anisotropic objects .....	50
<i>Victor A. Debelov, Dmitry S. Kozlov</i>	
Robust Silhouette Shadow Volumes on Contemporary Hardware .....	56
<i>Jan Peciva, Tomas Starka, Tomas Milet, Jozef Kobrtek, Pavel Zemcik</i>	

Mathematical Models of Fingerprints on the Basis of Lines Description and Delaunay triangulation . . . .	60
<i>Vladimir Gudkov</i>	
Edge Based Image Comparison and Noise Removal . . . . .	65
<i>Vladislav Shubnikov, Sergey Belyaev</i>	
Adaptive Feature Extraction for Gender Classification of Human Faces . . . . .	71
<i>Vladimir Khryashchev, Lev Shmaglit, Andrey Priorov, Andrey Shemyakov</i>	
Multidimensional filtering in application to progressive video rendering . . . . .	75
<i>Gruzdev A., Frolov V., Vostryakov K., Ignatenko A.</i>	
Image Filtration Based on Principal Component Analysis and Nonlocal Image Processing: Algorithms and Applications . . . . .	79
<i>Andrey Priorov, Kirill Tumanov, Vladimir Volokhov</i>	
Wavelet Lifting on Application Specific Vector Processor . . . . .	83
<i>David Barina, Pavel Zemcik</i>	
Regularization parameter choice for total variation based image deringing algorithm . . . . .	87
<i>Andrey V. Nasonov, Andrey S. Krylov</i>	
Curve-skeleton extraction using silhouettes' medial axes . . . . .	91
<i>Andrey Zimovnov, Leonid Mestetskiy</i>	
Quantum Hough Transform . . . . .	95
<i>Karthik Mahesh Varadarajan</i>	
Temporal multiple instance clustering for dynamic region selection in video . . . . .	99
<i>Yongqiang Zhang, Xudong Zhao, Daming Shi, Xianglong Tang</i>	
Dynamic region selection in video based on spatio-temporal multiple instance learning . . . . .	103
<i>XiaoZheng Wang, XuDong Zhao, Peng Liu, XiangLong Tang</i>	
An Integrated Image Clarity Method for Coal Mine Safety Monitoring . . . . .	106
<i>Yanqin Zhao</i>	
Data Processing for Managing Remote Sensing Images . . . . .	109
<i>Jianting Shi, Jiancai Wang, Fujun Zhao</i>	

## Technical section (Russian)

---

### RU1: Vision

3D SLAM по стереоизображениям . . . . .	114
<i>Бобков В.А., Роньшин Ю.И., Кудряшов А.П., Машенцев В.Ю.</i>	
Распознавание динамических жестов руки посредством обработки дальностных изображений человека . . . . .	118
<i>Ваагн Нагапетян</i>	
Система локализации робота на основе полусферической камеры . . . . .	122
<i>Александр Овчинников, Хоа Фан</i>	

## RU2: Visualization and modeling

<STAR> Современные направления развития визуализации данных .....	127
<i>А.Е. Бондарев, В.А. Галактионов</i>	
<STAR> Опыт создания систем визуализации реального времени и их применение в тренажерных и обучающих системах .....	131
<i>Б.С. Долговесов, Б.Б. Морозов</i>	
Восстановление 3D поверхности лица .....	135
<i>Щеголева Надежда</i>	
Манипулирование Свободными Формами с Функциями Возмущения .....	140
<i>С.И. Вяткин, Б.С. Долговесов, М.А. Городилов</i>	
Объектно-ориентированная модель построения фотореалистичных изображений .....	144
<i>Жданов Дмитрий Дмитриевич, Ершов Сергей Валентинович, Поздняков Сергей Георгиевич</i>	
Построение реалистичных изображений в системах автоматизированного проектирования .....	148
<i>Б.Х. Барладян, А.Г. Волобой, Л.З. Шапиро</i>	
Программно-аппаратный комплекс интерактивных мультимедийных презентаций .....	152
<i>Б.С. Мазурок, Б.С. Долговесов, Е.И. Коростелев, Т.Н. Артюков, А.Н. Артюков</i>	
Интерпретационный метод визуализации графовых алгоритмов .....	157
<i>Гордеев Дмитрий Станиславович</i>	
Графический моделирующий комплекс для автономного необитаемого подводного аппарата .....	161
<i>Михаил Морозов, Игорь Туфанов</i>	
Построение распределённой мультимедийной виртуальной среды с многоканальной визуализацией медиа-данных на графических акселераторах. ....	166
<i>Б.Б. Морозов, Б.С. Долговесов, Б.С. Мазурок, М.А. Городилов</i>	
Интерактивный выбор объектов трёхмерной сцены .....	171
<i>Е.И. Коростелев, М.А. Городилов, Б.С. Долговесов</i>	
О подсистеме истории в среде научной визуализации SharpEye .....	174
<i>П.А. Васёв, С.С. Кумков, Е.Ю. Шмаков</i>	
Алгоритм реализации скользящего стерео-курсора .....	178
<i>Александра Афанасьева, Алексей Игнатенко</i>	
Получение оверлеев векторных данных большого объёма .....	182
<i>И.А. Матвеев, И.А. Юдин</i>	

## RU3: Graphics

<STAR> Компьютерная графика как эффективный инструмент развития современных технологий .....	186
<i>А.Г. Волобой, В.А. Галактионов</i>	
Физически корректный алгоритм рендеринга анизотропных поглощающих кристаллов .....	191
<i>Д.С. Козлов, В.А. Дебелов</i>	
Количественная оценка уровня артефактов в объёмной визуализации .....	195
<i>Николай И. Гаврилов, Вадим Е. Турлапов</i>	
Тестирование систем реалистичной компьютерной графики .....	199
<i>А.Г. Волобой, Е.Ю. Денисов</i>	

Интерактивный рендеринг при помощи сферических дизайнов для низкочастотного окружающего освещения .....	203
<i>Свистунов Сергей</i>	
Эффективная и практичная реализация фотонных карт на GPU .....	207
<i>Владимир Фролов, Александр Харламов, Владимир Галактионов, Константин Востряков</i>	
Реконструкция непрозрачных включений в объектах с высоким показателем преломления .....	211
<i>Андрей Лебедев, Виктор Гаганов, Алексей Игнатенко</i>	
Калибровка источника света в сцене с прозрачным объектом по фотографиям .....	215
<i>В.В. Афанасьев, А.В. Игнатенко, И.О. Тисевич</i>	

## RU4: Image processing and recognition

Алгоритм автоматического обнаружения изображений номерных знаков железнодорожного подвижного состава на основе метода адаптивного бустинга .....	219
<i>И.В. Бекетова, С.Л. Каратеев, Ю.В. Визильтер</i>	
Концептуальный компонент в гибридной системе анализа изображений .....	222
<i>Геннадий Хахалин, Сергей Курбатов, Алексей Лобзин</i>	
Стеганографическое преобразование цветного изображения в полутоновое для распределения цифровых фотографий .....	226
<i>В.Н. Горбачев, И.К. Метелев, Е.М. Кайнарова, Е.С. Яковлева</i>	
О субполосном анализе изображений .....	230
<i>Е.Г. Жилияков, А.А. Черноморец</i>	
Создание универсальных классификаторов текстовых образов на основе сверточных нейросетевых технологий .....	234
<i>Кузьмицкий Николай Николаевич</i>	
Оценка применимости алгоритмов сегментации изображения для решения задачи оконтуривания левого желудочка сердца на УЗИ-изображениях .....	238
<i>Анастасия Бобкова, Сергей Поршнев, Василий Зюзин, Владимир Бобков</i>	
Двухфазная схема решения задачи классификации .....	241
<i>Евгений Прохоров, Михаил Кумсков</i>	
Исследование методов удаления спекл-шумов на ультразвуковых изображениях .....	244
<i>Анастасия Бобкова, Сергей Поршнев, Василий Зюзин, Владимир Бобков</i>	
Анализ временных рядов на основе алгебраических методов распознавания образов (концепции активного сенсора и многомасштабного анализа) .....	247
<i>Алексей Кузьмин, Михаил Кумсков</i>	
Система распознавания символов на изображениях со сложным фоном .....	250
<i>Анна Балахонцева, Александр Годоба, Нгуен Тьен</i>	
Алгоритм нелокального среднего на основе разложения по функциям Эрмита в задачах компьютерной томографии .....	254
<i>Н.В. Мамаев, А.С. Луккин, Д.В. Юрин, М.А. Глазкова, В.Е. Синицин</i>	
Детектирование жирных линий и древовидных структур на изображениях .....	259
<i>А.Е. Левашов, Д.В. Юрин</i>	

# Young Scientists School

---

Метод вычисления ошибок распознавания частных признаков дактилоскопических изображений <i>Гудков Владимир Юльевич, Дорофеев Константин Андреевич</i>	264
Автоматическое преобразование линейных объектов при построении цифровых моделей рельефа (ЦМР) <i>Шулькин Евгений</i>	266
Особенности использования регистровой памяти GPU в OpenCL на примере текстурного компрессора <i>Илья Перминов</i>	267
Навигация с применением RANSAC фильтрации точечных особенностей <i>Машенцев Владимир Юрьевич</i>	271
Система материалов и эффектов для реалистичной визуализации трехмерных сцен <i>А.Н. Артиков, Т.Н. Артиков</i>	275
Моделирование позвоночника <i>Дмитриев Сергей Валерьевич</i>	278
Обнаружение и слежение за объектом на основе совместного использования алгоритмов Виолы-Джонса и анализа траекторий <i>Мария Николаевна Гурьева</i>	280
Использование контрольных кластеров для корректирования деформации антропоморфных полигональных моделей <i>Елена Гридчина, Дмитрий Заставной, Александр Букатов, Борис Заставной</i>	282
Сжатие геометрической информации и оценка близости сложных объектов на основе порождающих моделей <i>Грудинин Сергей Николаевич, Фроловский Владимир Дмитриевич</i>	286
Выделение контуров зданий и распознавание служебных символов для трехмерной реконструкции объектов городской обстановки по топографическому плану <i>Игорь Соловьёв</i>	290
Отслеживание направления взгляда в реальном времени без использования специальной видеоаппаратуры <i>Иван Малин</i>	294
Создание панорамных карт подводного дна на основе больших массивов изображений <i>Камаев Александр</i>	298
Плагин метода световых сеток для 3ds Max <i>Наталья Бологова</i>	302
Алгоритм фильтрации папиллярных линий, наложенных на регулярную структуру <i>Суслов Александр Алексеевич, Гудков Владимир Юльевич</i>	306
Гомеоморфное отображение циркулярных фигур <i>Никита Ломов</i>	310
Влияние ложных признаков на качество сравнения дактилоскопических изображений <i>Гудков Владимир Юльевич, Лепихова Дарья Николаевна</i>	314
Автоматическая настройка параметров алгоритма сегментации изображений Quick Shift <i>Максим Колосовский, Елена Крючкова</i>	316
Визуализация поляризации света в сценах с кристаллами <i>Д.С. Козлов</i>	321



Двухуровневый параллелизм в задачах обработки и визуализации синоптических данных .....	325
<i>Черкашин Александр</i>	
Метод построения навигационной системы мобильного робота на основе использования бортовой веб-камеры .....	329
<i>Дмитрий Юхимец, Александр Новицкий</i>	
Сопровождение людей в системах многокамерного видеонаблюдения для спортивных игр .....	333
<i>Павел Батанов, Владимир Кононов, Антон Конушин</i>	
Визуализация средствами графических процессоров (GPGPU) смоделированных атомных структур аморфных сплавов .....	337
<i>Александр Дубинец, Евгений Пустовалов, Александр Федорен, Станислав Полищук, Владимир Ткачѐв</i>	

# Preface

Dear Participant,

We would like to welcome you at GraphiCon'2013, a leading international conference on computer graphics, computer vision, image and video processing in Russia. The 23rd event is hosted by Institute of Automatics and Control Processes Far Eastern Branch RAS and Far Eastern Federal University on September 16-20, 2013. Preparation for a conference was carried out in co-operation with Graphics and Media Lab, Faculty of computational mathematics and cybernetics, Lomonosov Moscow State University and Keldysh Institute of Applied Mathematics RAS.

This year we have an intense two-day program consisting of keynote presentations and scientific papers, carefully selected by International Program Committee for oral and poster presentations, and a special poster section of young scientist school.

The International Program Committee was formed of 70 members representing 13 countries from all over the world. Being top experts in the respective areas, all of them have done a tremendous job reviewing on average 3 papers out of 81 submitted. We express thanks to the committee members, who served at a 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 33 works selected for oral presentation and 13 papers for poster session. Also 22 papers will be presented as posters at young scientist school. Keynote presentations are made by Prof. Lei Xu (Chinese University of Hong Kong, China) and Prof. Stanislav Klimenko (Institute of Computing for Physics and Technology, Russia).

Keeping traditions, this year GraphiCon has not only scientific, 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. We would like to thank lecturers for their willingness to share state-of-the-art knowledge in the respective fields.

We would like to thank our sponsors - Russian Foundation for Basic Research and Russian Academy of Sciences, volunteers organized by School of Natural Sciences, Far Eastern Federal University, and everyone who made this event happen.

Valery Bobkov, on behalf of GraphiCon'2013 Organizing Committee



## Technical section (English)

---

GraphiCon'2013

September 16-20, 2013

Vladivostok, Russia

# A5 Problem Solving Formulations for Object Detection, Discriminant Analysis, and Integrative Hypothesis Testing

Lei Xu

Dept. of Computer Science and Engineering, Chinese Univ. of Hong Kong

xulei@phy.cuhk.edu.hk

## Abstract

We show that the A5 problem solving formulation provides a unified perspective for implementing not only object detection on images but also discriminant analysis and hypothesis testing on sample populations as widely encountered in bioinformatics and medical informatics. Under this formulation, two effective methods for object detection, namely randomized Hough transform (RHT) and multisets mixture learning (MML), are systematically elaborated and further extended by combining strengths of each.

Moreover, this formulation is also proposed as a general formulation of integrative hypothesis test (IHT) that not only improves hypothesis test with RHT mechanisms to enhance testing reliability but also integrates two types of hypothesis test (namely A-test versus I-test). Furthermore, this A5 formulation is generalized for three levels of integrative discriminant analysis that performs classification of samples into populations, learning parametric structures for modeling the populations, and hypothesis testing on whether the populations are significantly different, as well as identifying significant features that contribute to this distinction.

# Актуальные информационные технологии: визуализация информации, виртуальное окружение, неогеография, осязаемые изображения

С.В. Клименко

Московский физико-технический институт (государственный университет);  
Институт физико-технической информатики (Протвино)

## Аннотация

Постоянно растущая сложность физических явлений, изучаемых в научных и инженерных дисциплинах, требует разработки новых подходов и мощной техники для обработки и анализа сложных данных. Научная визуализация - Scientific Visualization - развивает методы и средства понимания решаемых проблем за счёт привлечения способности человека видеть и понимать изображения. В настоящее время научная визуализация считается вполне завершённой научной дисциплиной и наиболее перспективными являются направления визуализации информации и визуальная аналитика. Предназначение систем виртуального окружения заключается в том, чтобы обеспечить отдельных пользователей, или группы учёных, инженеров, дизайнеров виртуальным рабочим пространством, в котором они могут наблюдать, исследовать и создавать в реальном времени необходимые им виртуальные данные, модели и сцены.

**Ключевые слова:** научная визуализация, системы визуализации и виртуального окружения, визуализация информации, неогеография, осязаемые изображения

## 1. ВВЕДЕНИЕ

*«Визуализация - это способ, позволяющий одновременно совершенствовать концентрацию сознания и развивать в человеке соучастие»*

(А.Н.Райков. Лепесток опоры)

По мнению одного из выдающихся ученых в области информационных технологий - Фредерика Брукса - в настоящее время системы визуализации и виртуального окружения (ВО) переместились из научных лабораторий в сферу практических приложений, где её используют в своей повседневной работе инженеры, ученые, медики, военные - не обязательно специалисты по машинной графике.

Текущий момент характеризуется катастрофическим ростом информации, которое необходимо обрабатывать для поддержания прогресса в развитии современной цивилизации. Объем данных растет быстрее, чем производительность компьютеров, которая, следуя закону Мура, удваивается каждые полтора года. Но ведь данные, помимо порождения компьютерами, извлекаются непосредственно из самой природы с помощью всевозможных детекторов и сенсоров, быстрдействие которых также следует экспоненциальному росту.

Важнейшей вехой на пути преодоления «кризиса данных» стал отчет Национального научного фонда США «Визуализация в научных вычислениях» (1987-г.), подчеркнувший важность интерактивной визуализации больших массивов данных и обративший внимание научной общественности на знаменитый афоризм Хемминга: «Целью вычислений являются не числа, а

*понимание (постижение, проникновение в суть, интуиция, insight)». В результате было сформировано новое научное направление «Научная визуализация», развивающее методы и средства понимания решаемых проблем за счёт привлечения к анализу данных способности человека видеть и понимать изображения.*

Современные вычислительные системы позволяют моделировать сложные явления природы и решать задачи, недоступные прямой экспериментальной проверке. Суперкомпьютеры (наиболее мощные на текущий момент машины) и кластерные вычислительные системы позволяют достаточно полно и точно моделировать различные экзотические эксперименты и экстремальные ситуации, рожденные богатым воображением исследователей. Однако, огромные объёмы данных, получаемые при моделировании сложных явлений, невозможно проанализировать, не прибегая к предварительному исследованию с помощью активного взаимодействия человека с компьютером, в основе которого лежит синтез компьютером графического изображения и его анализ человеком, который опирается на свою мощную способность видеть и понимать визуальные изображения.

## 2. НАУЧНАЯ ВИЗУАЛИЗАЦИЯ

Научная визуализация как дисциплина сформировалась только в конце 90-х прошлого столетия. Ещё в 1993 г. Фред Брукс в своём ключевом докладе на конференции Визуализация'93 отмечал, что научная визуализация ещё не является дисциплиной, хотя уже зарождается как таковая.

Научная визуализация как перспективная и развивающаяся дисциплина чётко обозначила этапы, демонстрирующие её рост. Появились новые, определённые требованиями реальных задач, алгоритмы, которые эффективно работают с данными различного разрешения. Разработаны методы объёмной визуализации, предназначенные для реконструкции и отрисовки объектов по данным объёмного послayersового сканирования с помощью медицинских детекторов. Быстро прогрессирующая визуализация потоков в вычислительной аэро- гидродинамике обеспечила решение актуальных задач в исследовании космоса и аэро-космической промышленности. Особенно впечатляющие результаты были достигнуты по обработке изображений и реконструкции объектов в различных областях дистанционного зондирования планет и других космических объектов.

В настоящее время научная визуализация считается вполне завершённой научной дисциплиной и наиболее перспективными являются направления визуализации информации и визуальная аналитика. Вот почему на предстоящей конференции Vis2013 (13-18 октября, Атланта, США) приоритеты расставлены в порядке:

Visual Analytics Science and Technology, IEEE Information Visualization, and IEEE Scientific Visualization.



Визуализация информации, по идее, зародилась достаточно давно, но технологическую поддержку она получила на многочисленных воркшопах, которые регулярно проводились на конференциях IEEE\_CS Visualization.

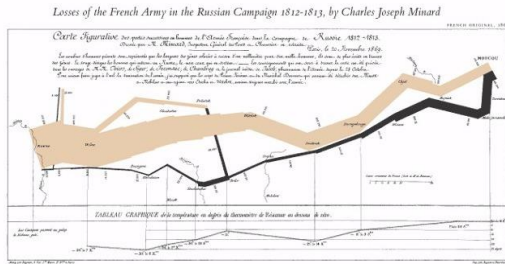


Рис. 1. Визуализация потерь армии Наполеона в войне 1812-1813 гг. (image courtesy by Charles Minard, 1869).

Следует отметить, что Слово ВИЗУАЛИЗАЦИЯ - производное существительное от глагола - visualize - to form a mental image; picture in the mind (Webster's Dictionary).

Известный профессор психобиологии Калифорнийского технологического института (в Пасадине) Роджер Сперри доказал, что два человеческих полушария мозга ответственны за обработку различной информации:

ЛЕВОЕ - вербальная, логическая,

ПРАВОЕ - визуальная, интуитивная.

В 1981 году Роджеру Сперри была присуждена Нобелевская премия по физиологии «за открытия, касающиеся функциональной специализации полушарий головного мозга».

Примерно 100 лет назад американские исследователи обнаружили: стереоскопическое представление при обучении усиливает пространственное воображение и понимание трёхмерных объектов и сцен и может стимулировать проявление у школьников и студентов *интуиции (insight)*, которая служит основой «догадок» при решении задач и которая редко возникает при традиционных методах обучения.



Рис. 2. Топологический зоопарк в виртуальном окружении.

Особенно эффективно работает визуализация в математике, точнее, в геометрии и топологии. Свои первые работы мы посвятили наглядному представлению двух моделей проективной плоскости в трехмерном пространстве, где показана непрерывная деформацию между ними, используя технику морфинга, и проводится анимация решения известной топологической проблемы о склеивании краёв диска и ленты Мёбиуса, которая, говоря словами Дж.Франсиса, была воротами в топологию для ряда поколений студентов [1].



Рис.3. Системы визуализации и ВО в образовании: виртуальный планетарий

### 3. ОТ ВИЗУАЛИЗАЦИИ К ВО

В процессе развития визуализации как научной дисциплины и технологии анализа данных было осознано, что человек лучше всего понимает и проникает в суть исследуемого явления, когда он может «погрузиться в мир исследуемого явления», то есть в пространство модели, и когда его «погружение» усиливается возможностью непосредственно манипулировать данными в пространстве модели. Так сформировалась технология ВО, определяемая как «интерактивная графика в реальном времени с трехмерными моделями, когда комбинируется специализированная технология отображения, погружающая пользователя в мир модели, с прямым манипулированием объектами в пространстве модели».

Следующей важной вехой в осознании важности развития систем анализа больших массивов данных, визуализации и ВО явилось стратегическое решение администрации Президента Клинтона о прекращении испытаний ядерного оружия. По этому решению от 25 сентября 1995 г. Министерству энергетики США, ответственному за разработку и производство ядерного оружия, предписывалось разработать систему компьютерного моделирования полномасштабных испытаний и по мере готовности такой системы полностью отказаться от ядерных испытаний. За этим решением последовал запуск Министерством энергетики инициативы, получившей название ASCI - Accelerated Strategic Computing Initiative. Важное место в этой программе отведено научной визуализации и ВО.

Создание действующих систем ВО рассматривается как комплексная проблема, поэтапное решение которой будет

оказывать самое серьезное воздействие на науку, промышленность, образование и социальную сферу. По своей сложности и долгосрочным последствиям она сопоставима с ядерной и космической проблемами.

По оценке профессора Брукса в настоящее время в мире действует более тысячи полномасштабных установок ВО, которые реально приносят ощутимую пользу в своих областях применения. Наиболее серьезные результаты получены на авиационных и автомобильных тренажерах, в системах подготовки экипажей торговых судов и военных кораблей, в задачах проектирования автомобилей, при тренировках космонавтов работе в условиях невесомости, при исследованиях и разработке нанотехнологий, при лечении артефактов у ветеранов войны во Вьетнаме и др.

Первая идея виртуальной реальности принадлежит основателю компьютерной графики Ивану Сазерленду [2], который в 1965 году на одной из своих лекций говорил: - «Не думайте об этом, как об экране монитора, думайте об этом как об окне, окне, через которое каждый может заглянуть в виртуальный мир. Основной задачей компьютерной графики является создание виртуального мира, реально выглядящего, реально звучащего, мира, в котором перемещения и реакции на воздействия происходят в реальном времени, мира, который ощущается реальным.» Эта идея Сазерленда тогда несбыточной мечтой, но прошло немногим более 20 лет и задуманный виртуальный мир был реализован, благодаря фантастическим достижениям компьютерной графики.

Визуальное погружение достигается за счёт создания стерео-эффекта наблюдаемой искусственной сцены и визуальным экранированием реальной среды, в которой находится пользователь. Для добавления звукового впечатления обычно используется синтезированный звук, синхронизированный с видео-информацией. Ещё больше усиливает впечатление от погружения в якобы реальный мир тактильная информация, создаваемая датчиками и устройствами силовой обратной связи, имитирующими сопротивление среды или предметов, с которыми взаимодействует пользователь. Следует отметить, что получение «реалистической» ВО требует кооперации многих исследовательских и промышленных лабораторий. Например, получение реалистичной текстуры многих природных материалов (медь, мрамор, гранит и т.п.) не поддаётся моделированию - для генерации этих текстур приходится измерять отражательные характеристики таких материалов экспериментальными методами физической оптики [3]. Нынешний быстрый рост технологии ВО был обеспечен научными, коммерческими и развлекательными применениями.

В настоящее время в мире существует более тысячи крупномасштабных установок ВО, которые используются в самых различных областях науки и техники для решения задач как фундаментальных научных дисциплин (астрономия, математика, физика, химия), так и в узко специализированных прикладных направлениях: аэрогидродинамика (визуализация динамики течения потоков), океанология и геофизика (инженерия землетрясений), металлообработка (авто- и авиа-индустрия), сопротивление материалов (моделирование упругих объектов), анализ столкновений и разрушений (моделирование аварий и катастроф), биомедицинская инженерия (протезирование и диагностика), и др.

Основным требованием, предъявляемым к программному обеспечению таких систем является высокая скорость графической обработки, интерактивная визуализация сложных сцен, эффективная синхронизация параллельно исполняемых процессов. Системы для разработки приложений в ВО [4-7] обеспечивают разработчика высокоуровневым интерфейсом для представления сложных геометрических моделей в виде графа сцены и обработки этого графа. Разработчик освобождён от рутинных операций взаимодействия с низкоуровневой графикой и системными программными интерфейсами и может сконцентрироваться на разработке собственно приложения.

#### Современные установки Barco [16]

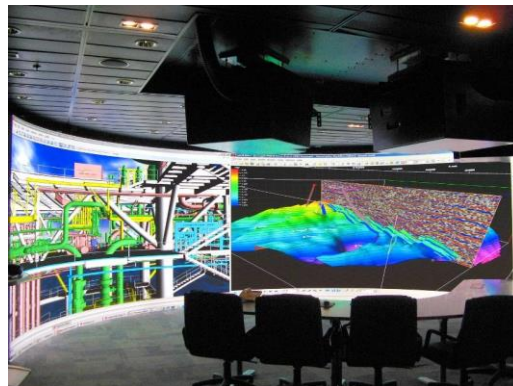


Рис. 4. Панорамная система виртуального проектирования Schell на базе установки Barco.

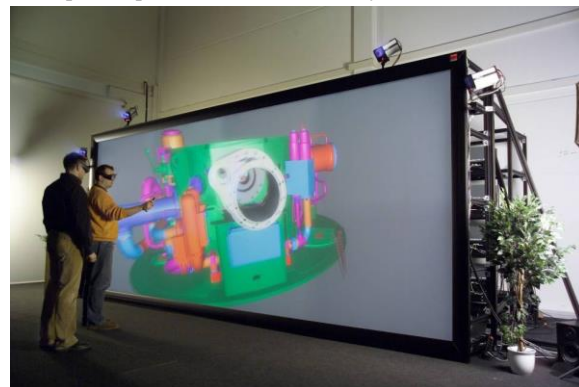


Рис. 5. Работа с виртуальным прототипом изделия на мозаичной системе высокого разрешения Barco MegaWall.

## 4. ОПЫТ СЕРЬЕЗНЫХ ИССЛЕДОВАНИЙ

### Визуализация индуцированного ВО.

В последние годы все более востребованными становятся возможности систем ВО, позволяющие воспроизводить в виртуальной среде объекты (их структуру, вид, поведение и взаимодействие), которые в режиме реального времени могут копировать поведение своих реальных прототипов, находящихся и функционирующих в реальной среде. Эти возможности можно кратко охарактеризовать метафорой «видеть невидимое», - точнее, получить изображение объектов наблюдения без использования средств прямого оптического наблюдения.

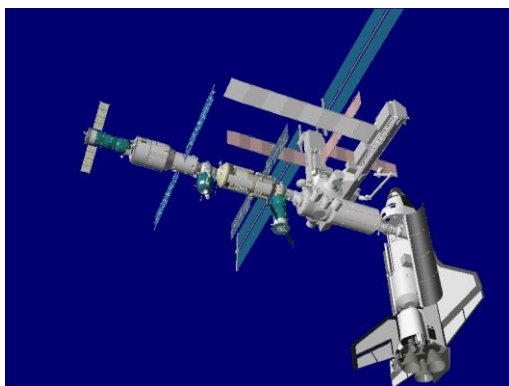


Рис.6. Виртуальная сцена: МКС с пристыкованным к ней Space Shuttle («Атлантис») и космическими кораблями («Союз» и «Прогресс»).

В первой половине прошлого века такая возможность появилась, благодаря открытию телевидения, причём сам термин «телевидение» как раз и отражает возможность «видения на расстоянии».

Имея сходство с системой телевидения, система ВО, в которой воспроизводятся реальные события, происходящие с реальными объектами, предоставляет гораздо более мощные возможности по сравнению с телевизионными системами, прежде всего заключающиеся в принципиальном отсутствии ограничений на свободу передвижения наблюдателя внутри виртуального пространства. Такие системы ВО имеют ярко выраженные функциональные особенности - реальные объекты фактически управляют своими «виртуальными двойниками». В работах В.О.Афанасьева с коллегами была реализована технология, в которой поведение виртуальных объектов индуцируется поведением реальных объектов. Поэтому созданную визуальную среду назвали Индуцированным Виртуальным Окружением (ИВО) [8].

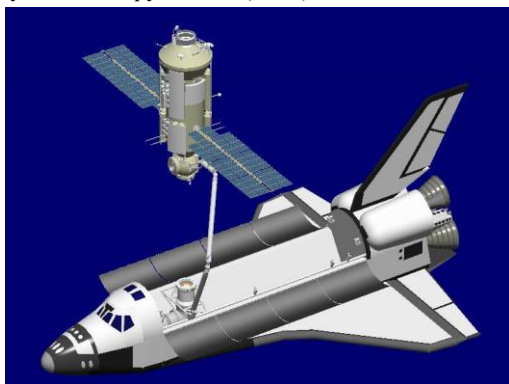


Рис.7. Виртуальная сцена: моменты стыковки модуля ФГБ при помощи манипулятора "Canada Arm" к стыковочному узлу корабля Space Shuttle

Виртуальные сцены в разных ракурсах показывают наиболее ответственные моменты некоторых орбитальных операций. Следует заметить, что в реальности увидеть развитие событий в том виде, как показано на рисунках, было бы невозможно, так как в космосе невозможно было бы нужным образом (и в таком количестве) разместить телекамеры.

#### База исследований Кафедры.

Программно-аппаратной базой проектов, связанных с восприятием 3D пространства, служит созданный на

Кафедре действующий прототип комплекса «тренажер-симулятор - система визуального погружения тренирующегося в виртуальную сцену трассы» на основе активной платформы тренажера-симулятора и системы ВО [9].



Рис.8. Интерьер Лаборатории информационных технологий в спорте.

Возможности визуализации с точки зрения использования в научных исследованиях вышли за обычные рамки создания визуальных образов различных физических явлений. В работах В.П.Алешина с коллегами разработаны методы и алгоритмы визуализации для решения задач мониторинга космических аппаратов (КА). Актуальность задачи мониторинга возрастает в связи с активным использованием космического пространства в различных прикладных целях (системы глобального позиционирования, космические телекоммуникации, исследование ресурсов, экологический контроль и т.д.). Непосредственным способом мониторинга является отображение соответствующих аварийных ситуаций, нерегистрируемых с помощью прямых видеонаблюдений, а генерируемых на основе идеологии ИВО по косвенным измерениям [10].

Для оценки сложности решаемых задач необходимо сравнить рисунки 9 и 10.



Рис.9. Реальные изображения КА UARS.

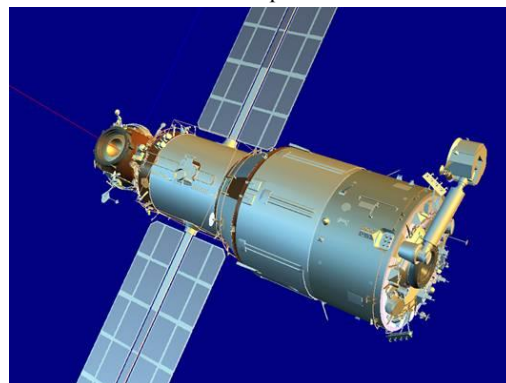


Рис.10. Реальные изображения КА UARS.



## 5. НОВЫЕ ПЕРСПЕКТИВНЫЕ НАПРАВЛЕНИЯ

### Поддержка принятия решений.

Предлагается реализовать ПТК ситуационного центра и системы класса 4i - Интегрированная Интерактивная Интеллектуальная Информационно-аналитическая система (ИАС4i), в основу которой положена интеграция методов и подходов ситуационной осведомленности, Неогеографии, ВО, серьезных игр, Грид, семантической Паутины, интеллектуальных информационных технологий и хранилищ данных [11].

ИАС4i обеспечивает развитые возможности анализа динамики развития ситуации на основе геопространственной и релевантной семантической информации, поставляемой распределенными сетями сенсоров и вычисляемых карт рисков. ПТК-ВО обеспечивает «погружение» аналитиков и ЛПР в обстановку ситуации и интуитивный интерфейс для доступа к информации. Интерактивные возможности ИАС4i обеспечивают вызов любой информации для любого объекта и «точки на карте».

### Обеспечение ситуационной осведомленности.

ИАС4i имеет 2 «окна» (1) ситуационный центр на основе системы ВО для ЛПР и экспертов; (2) Интернет-портал для обеспечения публичной осведомленности граждан.

Предлагаемый интеграционный подход обеспечивает решение актуальных задач, связанных с совершенствованием методов принятия управленческих решений в текущей жизни и в случаях ЧС на основе риск-информированного подхода, облегчением доступа к большому объему смежной информации, возможностью исследования множественных альтернативных сценариев, вовлечением большего числа участников в процесс принятия решений, обеспечением эффективной коммуникации между СКЦ, населением, отрядами оперативного реагирования и/или аварийно-спасательными формированиями.

Ярким примером ситуационного центра является Decision Theater (театр принятия решений), Университет штата Аризона (<http://dt.asu.edu/>).



Рис.11. Демонстрация возможностей Decision Theater.

### Неогеография.

Работы Института по визуальной аналитике были инициированы Е.Н.Еремченко, руководителем группы «Неогеография». *Неогеография* - новое поколение средств и методов работы с геопространственной информацией - основана на очень простых принципах, взяв за основу первичную информацию: - использованием географических систем координат, а не

картографических; - применением растрового, а не векторного представления географической информации в качестве исходного; -- использованием открытых гипертекстовых форматов представления геоданных [12].

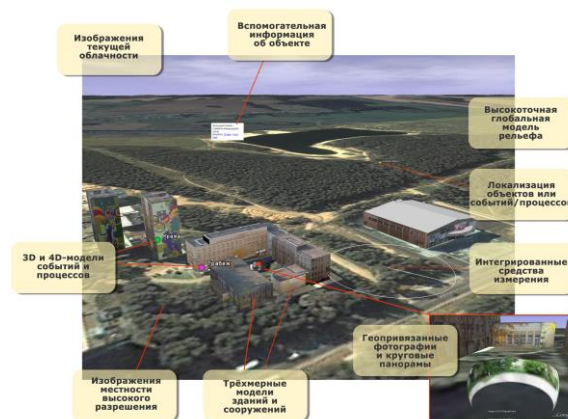


Рис.12. Визуальная аналитика с геоинтергейсом (Группа «Неогеография», Протвино).

*Неогеография* на деле показала действенность одного из принципов автоматизации (компьютеризации) человеческой деятельности, сформулированный В.М.Глушковым, о необходимости преобразования технологии этой деятельности к условиям и возможностям автоматизации. *Неогеография* своей необычайной привлекательностью обязана возможностям виртуальных путешествий.

### Осязаемые изображения.

Минимально инвазивная хирургия стала «золотым стандартом» для ортопедических операций на коленных суставах, которые делаются через небольшие разрезы типа «замочной скважины». Такие операции требуют от хирургов развития пространственного восприятия, чтобы понять трёхмерность хирургических сцен, а также научиться управлять артроскопическим инструментом. Обучение артроскопической хирургии с помощью виртуального тренажера является весьма актуальным, так как в реальной операции хирурги видят только двумерное изображение на плоском мониторе и вынуждены интуитивно управлять сложным инструментом в реально трёхмерной сцене. Конечно, в реальной операции хирург «ощущает» прикосновение инструмента к тканям пациента, поэтому для виртуального тренажера очень важно создать систему, в которой тренирующийся «видит» прикосновение инструмента к виртуальной ткани и «ощущает» через тактильную обратную связь это прикосновение. Такие изображения будем называть «осязаемыми».

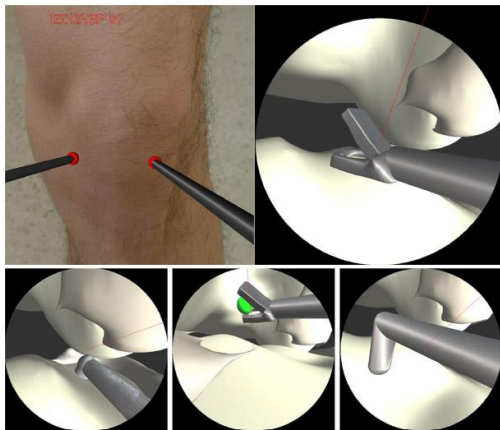


Рис.13. Виртуальный артроскоп и триангуляция в виртуальном пространстве.

В работах А.И.Сурина с коллегами [13-15] предложен принципиально новый подход к виртуальному моделированию минимально инвазивных операций с использованием тактильных устройств и осязаемых изображений - отказаться от трёхмерного моделирования операционного поля и использовать вместо этого реальные изображения и видео получаемые с помощью эндоскопических камер и микроскопов. На рисунках 13 и 14 фрагменты реальной и виртуальной «артроскопической операции».



Рис.14. Реальная операция и использование изображения операционного поля в виртуальном тренажере.

## 6. ЗАКЛЮЧЕНИЕ

**Новейшие технологии** человеко-машинного взаимодействия - визуализация и виртуальное окружение - обеспечивают повышение эффективности при проведении фундаментальных исследований, при решении сложных задач науки, образования и промышленности.

**Перспективными направлениями** исследований в данной предметной области являются визуализация информации, визуальная аналитика, новые человеко-машинные интерфейсы, основанные на тактильных устройствах и осязаемых изображениях.

Работы, послужившие основой доклада, были выполнены в рамках проектов, поддержанных грантами РФФИ 99-01-00451, 01-07-90327, 02-01-01139, 04-07-08026, 05-07-90344, 08-07-00468, 08-07-00469, 08-07-12037, 09-07-00401, 10-07-00513, 11-07-00514, 12-07-

00157, 12-07-00678, 12-07-00780, 12-07-12082, 13-07-00180, 13-07-00367, 13-07-00369, 13-07-00398.

## 7. ЛИТЕРАТУРА

- [1] George K. Francis, Topological Picturebook, Springer-Verlag, 1987 (М.:Мир, 1991).
- [2] Sutherland, I.E.: The Ultimate Display, Proc. IFIP 65, 2, pp.~506--508, 582--583, 1965
- [3] Ф. Баяковский Ю.М., Галактионов В.А. Современные проблемы компьютерной (машинной) графики <http://spkurdyumov.narod.ru/GalaktionovVladimir.htm>
- [4] H.Tramberend, A Distributed Virtual Reality Framework, Proc. of the IEEE Virtual Reality, 1999, pp.14-21
- [5] Клименко С.В., Никитин И.Н., Никитина Л.Д. «Аванго: система разработки виртуальных окружений» - Москва-Протвино, 2006, Институт физико-технической информатики, ISBN 5-88835-017-6, 252С.
- [6] D.S.Staneker, A first step towards occlusion culling in OpenSG PLUS// in Proc. of the 1st OpenSG Symposium, 2002
- [7] D.Bartz et al, Jupiter: A Toolkit for Interactive Large Model Visualization// Proc. of Symposium on Parallel and Large Data Visualization and Graphics, pp.129-134, 2001
- [8] Афанасьев В.О., Байгозин Д.А., Батурин Ю.М., и др. Системы визуализации и виртуального окружения в задачах исследования космоса: настоящее и будущее // в книге «Космонавтика XXI века. Попытка прогноза развития до 2101» под редакцией академика РАН Б.Е.Чертока, М. Изд. "ПТСофт", 2010, - С.185-256
- [9] Vladimir Aleshin, Valery Afanasiev, Alexander Bobkov, Stanislav Klimenko, Vitaly Kuliev and Dmitry Novgorodtsev. Visual 3D Perception of Motion Environment and Visibility Factors in Virtual Space // Transactions on Computational Science XVI Lecture Notes in Computer Science, 2012, Volume 7380/2012, 17-33, DOI: 10.1007/978-3-642-32663-9\2, ISBN: 978-3-642-32662-2, pp.17-33
- [10] Vladimir Aleshin, Stanislav Klimenko, Dmitry Novgorodtsev. Space objects localization and recognition using an adaptive optical observation system // in The Proc. of the 19 International Conference GraphiCon-2009, Moscow, Russia, ISBN 978-5-317-02975-3, pp.311-313
- [11] В.П.Алешин, А.П.Афанасьев, В.О.Афанасьев, и др. Программно-технический комплекс ситуационного центра для мониторинга и управления терри-ториями и проектами модернизации // MEDIAS2010 Труды Международной научной конференции, 10-14 мая 2010 г., Лимассол, Республика Кипр, Изд.ИФТИ, С.149-230, ISBN 978-5-88835-030-0
- [12] Володченко А., Ерёмченко Е.Н., Клименко С.В. О новых междисциплинарных ориентирах «Неогеографии» // MEDIAS2012 Труды Международной научной конференции, 07-14 мая 2012 г., Лимассол, Республика Кипр, Изд.ИФТИ, С.254-257, ISBN 978-5-88835-023-2
- [13] Шахзад Расул, Суриин Алексей, Клименко Станислав. Тактильное взаимодействие на основе осязаемых изображений // SC-IAS4i-VRTerro2011 Труды

Международных научных конференций:  
"Ситуационные центры и информационно-аналитические системы класса 4i", Изд.ИФТИ, Москва-Протвино, С.220-232, ISBN 978-5-88835-021-8

- [14] Pestrikov V., Sourin A., Towards Making Panoramic Images in Virtual Arthroscopy, In Proc 2013 Int Conf on Cyberworlds, IEEE Computer Society, Japan, 21-23 October, 2013 (in print).
- [15] Rasool S., Sourin A. Image-driven virtual simulation of arthroscopy, The Visual Computer, Springer, 29(5): 333-355, 2013.
- [16] Брусенцев П.А., Клименко С.В. Визуализация проектирования в виртуальном окружении // Клуб 3D. Инновационное проектирование. Изд. Литера, Нижний Новгород. С.37-48

## Abstract

The ever increasing complexity of the physical phenomena studied in scientific and engineering disciplines, requires the development of new approaches and powerful technique for processing and analysis of complex data. Scientific visualization develops methods and tools for understanding the problems to be solved by bringing in a person's ability to see and understand the image. Currently, scientific visualization is quite complete scientific discipline and the most promising areas of information visualization and visual analytics. Purpose of virtual environment is to ensure that individual users or groups of scientists, engineers and designers of the virtual work space, where they can observe and examine real-time create the necessary virtual their data models, and scenes.

**Keywords:** *scientific visualization, visualization systems and virtual environment, information visualization, neo-geography, tangible images*

## Author:

Станислав Владимирович Клименко, д.ф.-м.н., проф., научный руководитель кафедры МФТИ,  
E-mail: Stanislav.Klimenko@gmail.com

# A self-calibration method for a multi stereo-rig system based on robust 3D motion estimation

Oleg Stepanenko  
Vocord Company  
Moscow, Russia  
oleg.stepanenko@vocord.ru

## Abstract

This paper discusses self-calibration method for a multi stereo-rig passive face recognition system with overlapping views of individual cameras.

Two cameras fixed inside a rig will be easily calibrated in a factory environment so each rig will be considered individually calibrated. It is incredibly important to estimate relative positions and orientations of the stereo rigs with respect to each other without using any calibration objects. We propose calibration method using point correspondences between all cameras.

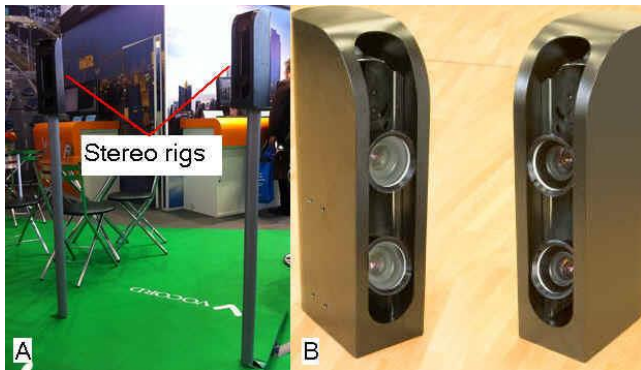
In this paper we present a novel more robust method that calculates metric reconstruction. Our method reconstructs 3D point set for every stereo rig from multiple dynamic scene images. Point correspondences are established by tracking points over all images captured simultaneously. So pairing between points is known but data keeping outliers. Then RANSAC technique is applied to reject outliers and finally 3D motion is estimated.

In this paper it is shown that the presented method can be accepted as sufficient in accuracy. Our method can be easily adopted for various numbers of stereo rigs.

**Keywords:** *Self-calibration, 3D motion estimation, stereo-rig.*

## 1. INTRODUCTION

Passive 3D face recognition system is the area of intense research over the past decade. A wide range of 3D acquisition technologies, with different cost and operation characteristics exists [1,10]. The most cost-effective solution is to use several calibrated 2D cameras fixed inside a rig to capture images simultaneously, and to reconstruct a 3D surface [11] (Fig.1A).



**Figure 1:** Passive 3D faces recognition system: A – common view; B – stereo-rigs.

The term *stereo rig* is used in this paper to refer to any two-camera system, which comprises a set of two cameras (with overlapping views) that are physically connected together and capture images simultaneously (Fig.1B).

Passive 3D face recognition system using stereo-rigs demands an accurate calibration of the devices which includes, first, intrinsic parameter measurement and estimation of the relative poses of the cameras with respect to each other inside a rig, second, estimation of relative positions and orientations of the stereo rigs with respect to each other. Our paper focuses on the second part of the calibration procedure (estimation of relative positions and orientations of the stereo rigs with respect to each other).

Roughly speaking, there are two groups of calibration methods. The first group is based on using of some object with known geometry (calibration pattern) or moving single feature like an LED [13, 15, 19].

One of such methods can be used for a calibration of two cameras fixed inside each rig. Due to the fact that calibration can be performed in a factory environment, stereo rigs are considered individually calibrated in this paper. Since the goal of this paper is to calibrate a multi-camera system without using any calibration objects, methods from the first group are out of our consideration.

The second and most suitable group of methods is self-calibration. Both the scene shape (3D structure) and the camera parameters (motion) consistent with a set of correspondences between scene and image features are estimated using this group of methods [3, 8, 17, 20].

The main part of calibration literature from the mentioned second group of calibration methods concerns extrinsic calibration. The goal of extrinsic calibration is to determine the 3D rotation and translation (3D motion) parameters relative to a fixed coordinate system. The estimations are based on 2D point features as they appear in an image sequence. Such methods are called structure from motion (SFM) methods. Usually 3D motion estimation methods involve two steps: first, 2D motion estimation that might be represented by 2D displacement vector field and, second, calculation of 3D motion from 2D displacements.

We propose to use 3D displacements for 3D motion estimation in this paper instead of using 2D displacement. We have 2D features as an input of our algorithm. We reconstruct 3D point set for every stereo rig using given calibration. So we receive sets of 3D displacements (keeping outliers) that can be used for estimating of 3D motion between 3D point sets (and consequently between stereo rigs). Such kind of 3D motion estimation methods using 3D points are sometimes called 3D alignment [16].

There are two main approaches to the problem of 3D motion estimation. If pairing between 3d points is known, closed form (analytic) solution is suitable [2, 6]. But analytic solution is breaking down in presence outliers (even if we have only one outlier). If pairing between points is unknown, iterative algorithms that start by matching nearby points and then update the most likely correspondence can be used [4]. (See [13] for an overview of applications.). Iterative algorithms require good initialization and they are sensitive to overlap and outliers [4]. Some approaches are proposed to deal with 2D feature correspondences

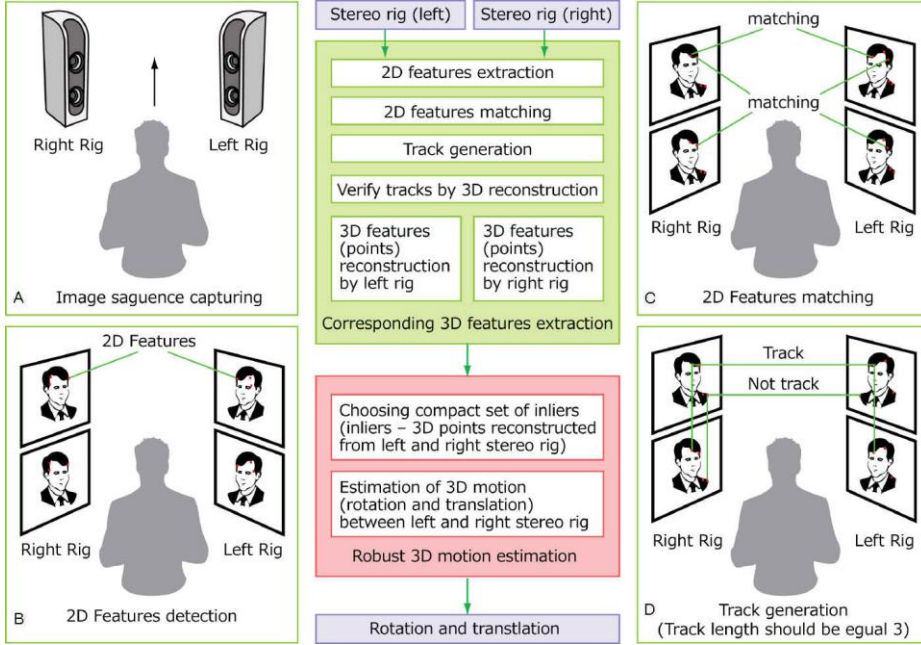
selection for robust camera calibration [12, 14, 18, 20]. However, reliable and automatic SFM is a difficult problem so far [18].

The paper is organized as follows. The proposed method is described in Section 2. Experiments on several multi-camera sequences are presented in Section 3. Conclusions are given in Section 4.

## 2. SELF-CALIBRATED METHOD

### 2.1 The overview of multi stereo-rig self-calibration method

The overview of multi stereo-rig self-calibration method we propose is shown in Fig.2.



**Figure 2:** The overview of multi stereo-rig self-calibration method: A – captured image sequence, B – 2D features detection, C - 2D features matching, D – track generation

The method mainly includes two stages, corresponding 3D feature extraction and robust 3D motion estimation. To obtain corresponding points, we let a person walks between stereo rigs (Fig.1A).

By analyzing the motion of human body from synchronous video sequences, we can find 2D points suitable for corresponding point (Fig.1B). Matching is performed to find corresponding points on the images captured simultaneously (Fig.1C).

Once we have pairwise matches, next step is to link up matches to form tracks (Fig.1D). Each track can potentially grow up to become eventually a 3D point. Some tracks might be inconsistent. Track should have length equal three in order to be consistent. We remove inconsistent tracks on the feature extraction stage. Then generated tracks are verified by 3D triangulation. On this step reconstructed 3D points should have distance from the appropriate stereo rig within certain reasonable range (for example, from 0.5 m to 1.5 m).

The next stage of our method is robust 3D motion estimation.

### 2.2 3D motion estimation algorithm based on RANSAC

The basic 3D alignment algorithms presented in literature are sensitive to outliers in the data [16]. As 3D tracks (and appropriate points correspondences) automatically extracted from images will almost always contain false matches, robustness with respect to outliers is very important. In this section, we will describe algorithm for this.

#### 2.2.1 Problem definition

Given a set of point correspondences  $= \{(P_{1,j}, P_{2,j}) \in \mathbb{P}^3 \times \mathbb{P}^3 | 1 \leq j \leq m\}$  measured in two Cartesian coordinate systems (left stereo rig, right stereo rig) find the rigid transformation (rotation and translation)  $R, T$ , between the two systems so that for corresponding points  $P_1$  (from left coordinate system) and  $P_2$  (from right coordinate system) we have:  $P_1 = RP_2 + T$ .

To achieve robustness with respect to false correspondences, the well-known (adaptive) Random Sample Consensus (RANSAC) approach [7, 9] can be applied. RANSAC is a very generic method for rejecting outliers. Here, we will describe a robust motion estimation algorithm based on RANSAC and one of the basic 3D alignment algorithms [6].

2.2.2 RANSAC based algorithm

The motion of point  $P_1$  from the left rig coordinate system can be expressed as  $P_1 = [X_1, Y_1, Z_1]^T = RP_2 + T$ , where  $P_2$  is the appropriate point in the right rig coordinate system, the orthogonal matrix  $R$  describes rotation and vector  $T$  describes translation of  $P_2$ . We assume that camera geometry is described by perspective projection with intrinsic camera parameters

$K \stackrel{\text{def}}{=} \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$ , where  $f_x$  and  $f_y$  are the effective focal lengths,  $s$  is the skew parameter, and  $(c_x, c_y)^T$  is the principal point.

RANSAC repeatedly samples a small subset  $D_s \subseteq D$  containing  $M$  point correspondences, and generates a hypothesis for the solution using only the sample  $D_s$ .

$M$  has to contain at least four non-planar 3D points. Each of the multiple solution candidates generated by one of the appropriate algorithm [6] can be treated as a single hypothesis  $E$ . Each hypothesis  $E$  describes appropriate motion  $(R_E, T_E)$  which is in turn a candidate of a solution of the whole task.

Each hypothesis is evaluated by counting how many 3D point correspondences are consistent with it. A correspondence  $(P_1, P_2) \in D$  is considered consistent with a hypothesis  $E$  if a suitable error measure  $dE(P_1, P_2)$  (error measure is presented further), is below a certain threshold  $\beta$ .

We propose to use the following error measure  $dE(P_1, P_2) = \left| \frac{1}{M} \sum_{(P_{1,j}, P_{2,j}) \in D_s} |P_{1,j} - RP_{2,j} - T| - |P_1 - RP_2 - T| \right|$ .

The set of all consistent correspondences  $S = \{(P_1, P_2) \in D | dE(P_1, P_2) < \beta\}$  is called the support set of

hypothesis  $E$ . The hypothesis with the largest support set  $S_L$  found during the iterations could be returned as the result of RANSAC.

Usually, in the final step, the result is estimated from  $S_L$ . However this approach is based on the assumption that  $S_L$  is outlier-free, which in general cannot be guaranteed. So in our algorithm the result is estimated from  $S$ . As the influence of noise is typically lower when estimating from a large set of data (as opposed to the very small samples  $D_s$ ) we sampled a subset  $D_s \subseteq D$  containing more than 4 point correspondences (as usual – 10 point correspondences).

Let us assume the data  $D$  contains a proportion  $\varepsilon$  of outliers. The probability of getting at least one outlier-free sample  $D_s$  is  $p_1 = 1 - (1 - (1 - \varepsilon)^M)^N$ , where  $N$  denotes the number of RANSAC iterations. In order to get at least one outlier-free sample  $D_s \subseteq D$  with a probability of (at least)  $p_1$ , we hence need to perform at least  $N > \log_{1-(1-\varepsilon)^M}(1-p_1)$  iterations. Typically, the proportion of outliers  $\varepsilon$  is unknown. The number of iterations of the RANSAC algorithm can optionally be adapted on-line basing on the following approach [2]. Let us assume the largest support set  $S_L$  is founded during previous iterations. It can be used to derive an upper bound for  $\varepsilon$  [2]:  $\varepsilon \leq \frac{|S_L|}{|D|}$ .

Hence, the required number of iterations is [2]:

$$N_{S_L} \stackrel{\text{def}}{=} \left\lceil \frac{\log(1-p_1)}{\log\left(1 - \left(1 - \frac{|S_L|}{|D|}\right)^M\right)} \right\rceil.$$

If the proportion of outliers is very high, however,  $N_{S_L}$  might always stay very large leading to a very big running time. In order to enforce a certain limit on the running time, we specify a maximum number of iterations in the beginning and make sure that  $N$  not increased by using the following adaptation rule [2]:  $N := \min(N, N_{S_L})$ .

When we have 3D motion ( $R$  and  $T$ ) calculated we can easily calculate position and orientation of all cameras relatively to each other.

### 3. EXPERIMENTAL RESULTS

The algorithm described above was tested with synthetic and experimental data. Synthetic data allows us to study the algorithm with respect to 3D image noise and to assess the conditions under which reliable results are expected.

We used two types of experimental data: calibration points and real points. Calibration points are obtained from the images of 3D calibration object (chessboard pattern). Since the sizes of this pattern are known, we can use standard camera calibration algorithm and compare the results obtained with our self-calibration algorithm with standard camera calibration algorithm. Calibration points are so accurate that the motion parameters obtained with this data and with the standard calibration algorithm may be considered as the ground-truth.

The synthetic data consists of a hundred 3D points. The points in the set were chosen randomly from a uniform distribution within a cube of size 750x750x750 mm centered about the origin. The synthetic data was formed by adding noise to the individual points and transforming them to a new location. Then a percentage of outliers was injected in the point set. The noise added to each component was uncorrelated, isotropic and Gaussian, with zero mean and variable standard deviation. For each noise level one hundred trials were performed. The average response of algorithm over this hundred trials was used to compute several different error statistics for the calculated transformations. For absolute

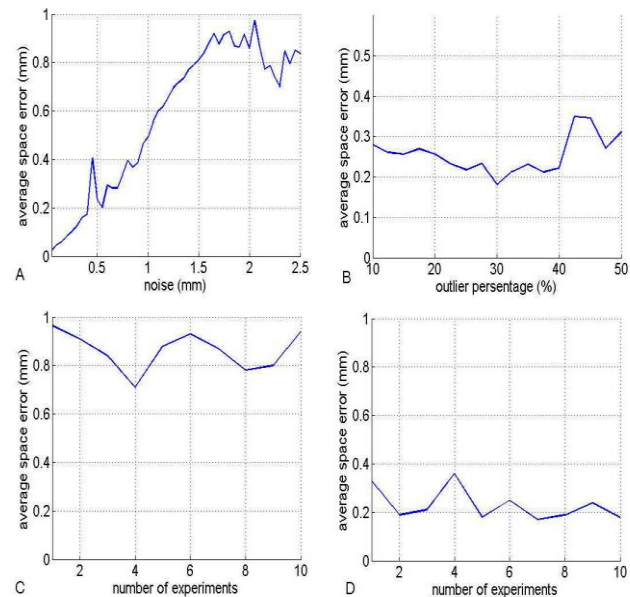
accuracy error estimation in this article we propose mean 3D position error. It is given by  $e_{mean} = \frac{1}{m} \sum_j^m |P_{1,j} - \hat{R}P_{2,j} - \hat{T}|$ , where  $\hat{R}$  is rotation estimation,  $\hat{T}$  is translation estimation.

#### 3.1.1 Noise sensitivity analysis

The noise added to the 3D points is Gaussian with standard deviation varying from 0.05 to 2.5 mm. A percentage of outliers that was injected in the dataset was established as 25%.

We studied the behavior of the algorithm as a function of 3D point noise while a percentage of outliers is fixed.

Fig.3A shows mean 3D position error with various standard deviation.



**Figure 3:** Mean 3D position error: A – synthetic data for different standard deviation of noise; B – synthetic data for different outlier percentage; C – real data: images of moving human body; D – real data: images of static chessboard pattern.

From fig.3A we can deduce the stability of self-calibration method against noise on the poses of the input 3D features.

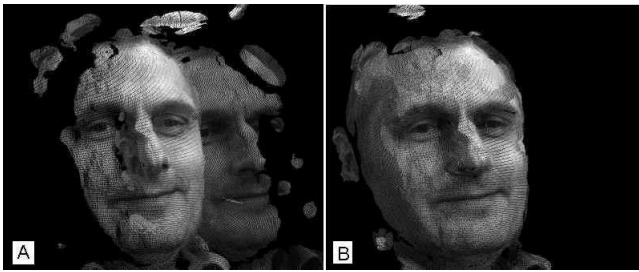
#### 3.1.2 Outliers sensitivity analysis

Percentage of outliers that was injected in the dataset was varying from 0 to 50 %. The noise added to the 3D points is Gaussian with standard deviation that was established as 1.0 mm.

Fig.3B shows mean 3D position error with various percentages of outliers. From fig.3B we can deduce the quality of the estimation is independent of the outlier percentage.

#### 3.1.3 Experiments with real data

Real experiments were conducted using image sequences of a moving person. Two stereo-rigs were calibrated by using chessboard pattern. Feature points of the body were found, filtered and reconstructed. Motion of these points was estimated using the proposed method.



**Figure 4:** 3D image of a head: A - two stereo rigs are not calibrated so two 3D images of the head are not aligned. B – after self-calibration. Two 3D images of the head are aligned after self-calibration has performed using the proposed method.

Fig.4A shows motion that exists between two 3D images of a head reconstructed using images from different rigs since extrinsic calibration between stereo rigs was not known. Then 3D motion between point sets was estimated using the proposed method. Fig.4B shows that two 3D images of the head are aligned after 3D motion estimation.

Fig.3C and Fig.3D show mean 3D position error in various experiments with real data. Fig.3C shows mean 3D position error for 3D motion calculated using proposed method and points detected and matched on images of moving human body. Fig.3D shows mean 3D position error for 3D motion calculated using classical calibration method [19].

From fig.3C and fig.3D we can deduce that accuracy of the presented method is worse than accuracy of common calibration techniques but it can be accepted as sufficient for some practical purposes. For example, proposed method can be used when using of calibration objects is undesirable or impossible.

#### 4. CONCLUSION

A new self-calibration algorithm has been proposed for obtaining 3D motion parameters of a multi stereo-rig system over time without using any particular calibration apparatus. The idea is to use previously valid stereo-rig calibration parameters and image point matches to perform an alignment of two 3D paired point sets that contains outliers. The method is tested in both artificial data and real video sequences. The results show that our method is robust in datasets with up to 50% of outliers. The advantage of our approach is the fact that no calibration objects are needed to perform metric calibration of the multi stereo rig system as most reference approaches demand.

The proposed method was evaluated against standard method for multi-rig calibration and proved to have acceptable in accuracy.

#### 5. REFERENCES

- [1] Mostafa Abdelrahman, Asem M. Ali, Shireen Y. Elhajian, Ham Rara, Aly A. Farag "A passive stereo system for 3D human face reconstruction and recognition at a distance", CVPR Workshops, 2012.
- [2] K.S. Arun, T.S. Huang, S.D. Blostein Least-squares fitting of two 3-D point sets. *IEEE Trans Pattern Anal Machine Intell* 9:698–700, 1987.
- [3] F.Bajramovic "Self-Calibration of Multi-Camera Systems . PhD thesis", Friedrich Schiller University of Jena ,2010.
- [4] D. Chetverikov, D. Stepanov, P. Krsek "Robust Euclidean alignment of 3D point sets: the Trimmed Iterative Closest Point

algorithm" , *Image and Video Computing* (2005), vol.23 (3), p.p.299–309.

[5] D. Colbry "Human Face Verification by Robust 3D Surface Alignment. PhD thesis", Michigan State University ,2006.

[6] D.W. Eggert, A. Lorusso, R.B. Fisher "Estimating 3-D rigid body transformations: a comparison of four major algorithms", *Machine Vision and Applications* (1997) vol. 9, p.p. 272–290.

[7] M. Fischler and R. Bolles. *Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography*. *CACM*, 24(6)(1981),p.p.381–395.

[8] A. Fusiello "Uncalibrated Euclidean reconstruction: a review", *International Journal of Image and Vision Computing*, vol. 18, No.6-7 (2000), p.p.555-563

[9] Hartley, R. and Zisserman, "A. *Multiple View Geometry in Computer Vision* (2nd ed.)",Cambridge: Cambridge University Press, 2003.

[10] Akihiro Hayasaka, Takuma Shibahara, Koichi Ito, Takafumi Aoki, Hiroshi Nakajima, Koji Kobayashi "A Passive 3D Face Recognition System and Its Performance Evaluation", *IEICE Trans.Fund.* v. E91-A, p. 1974-1981, 2008.

[11] Svetlana V. Korobkova, Archil Tsiskaridze "Face recognition system using 2D and 3D information fusion" , *Proceedings of Graphicon 2011*, pp. 153–156, 2011.

[12] Y. Ma, S. Soatto, J. Kosecka and Shankar Sastry. "An Invitation to 3D Vision: From Images to Models". Springer Verlag, December 2003.

[13] F. Pedersini, A. Sarti, S. Tubaro "Accurate and simple geometric calibration of multi-camera systems", *Signal Processing* vol. 77, No.3 (1999), p.p. 309-334.

[14] N. Snavely, S. M. Seitz, and R. Szeliski. *Photo tourism: exploring photo collections in 3d*. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 835–846, 2006.

[15] T. Svoboda "Quick guide to multi-camera self-calibration. Technical report", Computer Vision Lab, Swiss Federal Institute of Technology, Zurich, 2003.

[16] R. Szeliski "Computer Vision: Algorithms and Applications". Springer, 2010.

[17] B. Triggs "Autocalibration from Planar Scenes". *Proceedings of the European Conference on Computer Vision (ECCV)*, vol. 1(1998),p.p. 89–105

[18] X. Zhang, Y. Zhang, X. Zhang, T. Yang, X. Tong, H. Zhang A convenient multi-camera self-calibration method based on human body motion analysis. *Proceedings of the Fifth International Conference on Image and Graphics, ICIG 2009, China*, p.3-8, 2009.

[19] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.22, No.11(2000), p.p. 1330-1334

[20] Z. Zhang. "Motion and Structure From Two Perspective Views: From Essential Parameters to Euclidean Motion Via Fundamental Matrix". *Journal of the Optical Society of America* , Vol.14, no.11, pages 2938-2950, 1997.

#### About the author

Oleg Stepanenko (Ph.D, Associate Professor) is a scientist at Vocord Company, Department of Advanced Developing. His contact email is [oleg.stepanenko@vocord.ru](mailto:oleg.stepanenko@vocord.ru)

# A Minimal Six-Point Auto-Calibration Algorithm

Evgeniy Martyshev

Department of Mechanics and Mathematics  
 South Ural State University, Chelyabinsk, Russia  
 mev@susu.ac.ru

## Abstract

A non-iterative auto-calibration algorithm is presented. It deals with a minimal set of six scene points in three views taken by a camera with fixed but unknown intrinsic parameters. Calibration is based on the image correspondences only. The algorithm is implemented and validated on synthetic image data.

**Keywords:** *Projective reconstruction, Metric reconstruction, Auto-calibration.*

## 1. INTRODUCTION

The problem of camera calibration is a necessary part of computer vision applications such as path-planning and navigation for robots, self-parking systems, camera based industrial detection and recognition, etc. At present, a great deal of calibration algorithms and techniques have been developed. Some of them require to observe a planar pattern viewed at several different orientations [6, 15]. Other methods use the 3-dimensional calibration objects consisting of two or three pairwise orthogonal planes, whose geometry is known with good accuracy [14]. In contrast with the just mentioned methods, the *auto-calibration* does not require any special calibration objects [3, 4, 7, 8, 10, 13], so only point correspondences in several uncalibrated views are required. This provides the auto-calibration approach with a great flexibility and makes it indispensable in some real-time applications.

In this paper we give a new non-iterative solution to the auto-calibration problem in a minimal case of six scene points in three views, provided that the intrinsic parameters of a moving camera are fixed. Our method consists of two major steps. First, we use the efficient six-point three-view algorithm from [11] to solve for projective reconstruction. Then, using the well-known constraints on the absolute dual quadric [5, 13], we produce a system of non-linear polynomial equations, and resolve it in a numerically stable way by a series of Gauss-Jordan eliminations with partial pivoting.

The rest of the paper is organized as follows. In Section 2, we briefly recall how to construct a projective reconstruction from six matched scene points in three uncalibrated views. In Section 3, an algorithm of metric upgrading of the projective reconstruction is described. In Section 4, we test the algorithm on a set of synthetic data. Section 5 concludes.

### 1.1 Notation

We use  $\mathbf{a}, \mathbf{b}, \dots$  for column vectors, and  $\mathbf{A}, \mathbf{B}, \dots$  for matrices. For a matrix  $\mathbf{A}$ , the entries are  $A_{ij}$  or  $(\mathbf{A})_{i,j}$ , the transpose is  $\mathbf{A}^T$ , and the determinant is  $\det(\mathbf{A})$ . For two vectors  $\mathbf{a}$  and  $\mathbf{b}$ , the vector product is  $\mathbf{a} \times \mathbf{b}$ , and the scalar product is  $\mathbf{a}^T \mathbf{b}$ . We use  $\mathbf{I}_n$  for identical matrix of size  $n \times n$  and  $\mathbf{0}_n$  for zero  $n$ -vector.

## 2. PROJECTIVE RECONSTRUCTION

First of all, to avoid any degeneracies, we restrict ourselves to the “general position case” both for scene points and camera motions,

i.e., the sequence of camera motions is assumed to be non-critical and all the observed points do not lie on critical surfaces in a sense of [12]. In particular, this means that the scene is non-planar and the motion is not a pure translation or rotation around the same axis.

Given three uncalibrated images of six points of a rigid scene, we first produce a projective reconstruction of the cameras applying the minimal 3-view algorithm from [11]. Recall that the output of this algorithm is either one or three real solutions for the homogeneous coordinates of the sixth scene point  $\mathbf{X}_6$ , whereas the first five points are chosen to be the vectors of standard basis of the projective 3-space. The twelve entries of the camera matrix  $\mathbf{P}_i$  are then recovered by solving the twelve linearly independent equations (for each  $i = 1, 2, 3$ ):

$$\mathbf{x}_{ij} \times \mathbf{P}_i \mathbf{X}_j = \mathbf{0}_3, \quad j = 1, \dots, 6,$$

where  $\mathbf{x}_{ij}$  is the image of  $\mathbf{X}_j$  under the projection  $\mathbf{P}_i$ . Thus we found

$$\mathbf{P}_i = [\mathbf{A}_i \quad \mathbf{a}_i], \quad i = 1, 2, 3.$$

Using the projective ambiguity [5], we transform the obtained camera matrices to

$$\begin{aligned} \mathbf{P}'_1 &= \mathbf{P}_1 \mathbf{H}_0 = [\mathbf{I}_3 \quad \mathbf{0}_3], \\ \mathbf{P}'_2 &= \mathbf{P}_2 \mathbf{H}_0 = [\mathbf{B}_2 \quad \mathbf{b}_2], \\ \mathbf{P}'_3 &= \mathbf{P}_3 \mathbf{H}_0 = [\mathbf{B}_3 \quad \mathbf{b}_3], \end{aligned} \quad (1)$$

where

$$\mathbf{H}_0 = \begin{bmatrix} \mathbf{A}_1^{-1} & -\mathbf{A}_1^{-1} \mathbf{a}_1 \\ \mathbf{0}_3^T & 1 \end{bmatrix}.$$

## 3. METRIC RECONSTRUCTION

The projective reconstruction (1) is the starting point for our auto-calibration algorithm. Let the metric camera matrices be represented as

$$\begin{aligned} \mathbf{P}_1^M &= \mathbf{K} [\mathbf{I}_3 \quad \mathbf{0}_3], \\ \mathbf{P}_2^M &= \mathbf{K} [\mathbf{R}_2 \quad \mathbf{t}_2], \\ \mathbf{P}_3^M &= \mathbf{K} [\mathbf{R}_3 \quad \mathbf{t}_3], \end{aligned} \quad (2)$$

where  $\mathbf{R}_i$  and  $\mathbf{t}_i$  are the rotation matrix and translation vector respectively, and  $\mathbf{K}$  is an upper triangular matrix called the *calibration matrix* of the camera. It is assumed to be identical for all three views. Our goal is to estimate  $\mathbf{K}$  and then upgrade the projective cameras to the metric ones.

Auto-calibration determines a  $4 \times 4$  projective matrix  $\mathbf{H}$ , that transforms the projective camera  $\mathbf{P}'_i$  from (1) into a metric camera  $\mathbf{P}_i^M$  from (2), i.e.,

$$\mathbf{P}_i^M = \mathbf{P}'_i \mathbf{H}, \quad i = 1, 2, 3. \quad (3)$$

The matrix  $\mathbf{H}$  must have the form [5]:

$$\mathbf{H} = \begin{bmatrix} \mathbf{K} & \mathbf{0}_3 \\ -\mathbf{p}^T \mathbf{K} & 1 \end{bmatrix}$$



for some 3-vector  $\mathbf{p}$ . Then the entries of  $\mathbf{H}$  are constrained by [1, 5]

$$\begin{aligned}\lambda\omega^* &= \mathbf{P}'_2 \mathbf{Q}_\infty^* \mathbf{P}'_2{}^T, \\ \mu\omega^* &= \mathbf{P}'_3 \mathbf{Q}_\infty^* \mathbf{P}'_3{}^T,\end{aligned}\quad (4)$$

where  $\omega^* = \mathbf{K}\mathbf{K}^T$  is the *dual image of the absolute conic*,  $\lambda, \mu$  are scalars and  $4 \times 4$  matrix

$$\mathbf{Q}_\infty^* = \begin{bmatrix} \omega^* & \mathbf{q} \\ \mathbf{q}^T & r \end{bmatrix},$$

with  $\mathbf{q} = -\omega^* \mathbf{p}$ ,  $r = \mathbf{p}^T \omega^* \mathbf{p}$ , is called the *absolute dual quadric* [13].

Thus, constraints (4) give 12 equations in 11 variables:  $r, q_1, q_2, q_3$ , five components of  $\omega^*$  (recall that  $\omega_{33}^* = 1$ ),  $\lambda$  and  $\mu$ . Let us rewrite these equations in form

$$\mathbf{C}\mathbf{x} = \mathbf{0}_{12}, \quad (5)$$

where

$$\mathbf{C} = \mathbf{C}(\lambda, \mu) = \begin{bmatrix} \mathbf{0}_{6 \times 4} & \lambda \mathbf{I}_6 \\ \mathbf{0}_{6 \times 4} & \mu \mathbf{I}_6 \end{bmatrix} - \mathbf{D}, \quad (6)$$

$\mathbf{D}$  is a  $12 \times 10$  scalar matrix, and

$$\mathbf{x} = [r \quad q_1 \quad q_2 \quad q_3 \quad \omega_{11}^* \quad \omega_{12}^* \quad \omega_{13}^* \quad \omega_{22}^* \quad \omega_{23}^* \quad 1]^T$$

is a monomial vector.

It follows that the determinant of any  $10 \times 10$  submatrix of  $\mathbf{C}$  must vanish. Denote by  $S_i(\lambda, \mu)$  the determinant of a submatrix of  $\mathbf{C}$  obtained by eliminating the rows with numbers  $i$  and  $i + 6$  for  $i = 1, \dots, 6$ . Hence we get the system  $S_i = 0$  of polynomial equations in only two variables  $\lambda$  and  $\mu$ .

*Remark 1.* Due to the form (6) of matrix  $\mathbf{C}$ , we do not need to compute a  $10 \times 10$  functional determinant here. Each polynomial  $S_i$  can be found as

$$\det(\mathbf{C}_1 + \lambda \mathbf{C}_2 + \mu \mathbf{C}_3),$$

where the  $5 \times 5$  scalar matrices  $\mathbf{C}_j$  are obtained by a partial Gauss-Jordan elimination on matrix  $\mathbf{C}$ .

Let us rewrite the system  $S_i = 0, i = 1, \dots, 6$ , in form:

$$\mathbf{F}_0 \mathbf{y} = \mathbf{0}_6, \quad (7)$$

where  $\mathbf{F}_0$  is a  $6 \times 18$  coefficient matrix, and

$$\mathbf{y} = [\lambda^4 \mu \quad \lambda^3 \mu^2 \quad \lambda^2 \mu^3 \quad \lambda \mu^4 \quad \mu^4 \quad \lambda^3 \mu \quad \lambda^2 \mu^2 \quad \lambda \mu^3 \quad \lambda^3 \quad \lambda^2 \mu \quad \lambda \mu^2 \quad \mu^3 \quad \lambda^2 \quad \lambda \mu \quad \mu^2 \quad \lambda \quad \mu]^T \quad (8)$$

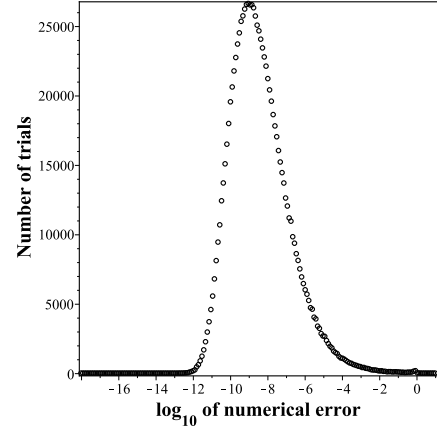
is a monomial vector. To solve the system (7) in a numerically stable way, we perform the following sequence of matrix transformations:

$$\mathbf{F}_0 \rightarrow \tilde{\mathbf{F}}_0 \rightarrow \mathbf{F}_1 \rightarrow \tilde{\mathbf{F}}_1 \rightarrow \mathbf{F}_2 \rightarrow \tilde{\mathbf{F}}_2 \rightarrow \mathbf{F}_3 \rightarrow \tilde{\mathbf{F}}_3, \quad (9)$$

where each  $\tilde{\mathbf{F}}_i$  is obtained from  $\mathbf{F}_i$  by the Gauss-Jordan elimination with partial pivoting.

The matrix  $\mathbf{F}_1$  of size  $8 \times 18$  is obtained from  $\tilde{\mathbf{F}}_0$  by adding two new rows: first one corresponds to the last row of  $\tilde{\mathbf{F}}_0$  multiplied by  $\lambda$ , second one — to the next to last row of  $\tilde{\mathbf{F}}_0$  multiplied by  $\mu$ .

The matrix  $\mathbf{F}_2$  of size  $12 \times 18$  is obtained from  $\tilde{\mathbf{F}}_1$  by adding four new rows corresponding to the last two rows of  $\tilde{\mathbf{F}}_1$  multiplied by  $\lambda$  and  $\mu$ .



**Figure 1:** Numerical error distribution. Median error is  $2.8 \times 10^{-9}$ .

The matrix  $\mathbf{F}_3$  of size  $17 \times 18$  is obtained from  $\tilde{\mathbf{F}}_2$  by adding five new rows. We multiply the last two rows of  $\tilde{\mathbf{F}}_2$  by  $\lambda$  and  $\mu$ , and thus get four additional rows. One more row is obtained by multiplying the 10th row of  $\tilde{\mathbf{F}}_2$  by  $\mu$ .

Finally we get

$$\mu = -(\tilde{\mathbf{F}}_3)_{16,18}, \quad \lambda = -\mu(\tilde{\mathbf{F}}_3)_{17,18}.$$

*Remark 2.* From algebraic point of view, the above sequence (9) *interreduces* the ideal  $J = \langle S_i \mid i = 1, \dots, 6 \rangle$ . The result is the Gröbner basis of  $J$  with respect to the graded lexicographic order. It consists of two polynomials represented by the last two rows of matrix  $\tilde{\mathbf{F}}_3$ .

Having found  $\lambda$  and  $\mu$ , we compute the entries of  $\omega^*$  performing the Gauss-Jordan elimination with partial pivoting on matrix  $\mathbf{C}$  in (6). Finally, we compute the calibration matrix by the Cholesky decomposition of  $\omega^* = \mathbf{K}\mathbf{K}^T$ , and then find (up to scale) the metric camera matrices  $\mathbf{P}_i^M$  by (3).

*Remark 3.* Note that the matrices  $\mathbf{R}_i$  estimated from (2) are not in general rotations and thus need to be corrected [15]. We used the singular value decomposition  $\mathbf{R}_i = \mathbf{U}_i \mathbf{D}_i \mathbf{V}_i^T$  and then replaced  $\mathbf{R}_i$  by  $\tilde{\mathbf{R}}_i = \mathbf{U}_i \mathbf{V}_i^T$ . It is well-known that the rotation matrix  $\tilde{\mathbf{R}}_i$  is the closest to  $\mathbf{R}_i$  with respect to Frobenius norm.

#### 4. EXPERIMENTS ON SYNTHETIC DATA

The algorithm has been implemented in C/C++. All computations were performed in double precision. Synthetic data setup is given in Table 1, where the baseline length is the distance between the first and third camera centers. The second camera center varies randomly around the baseline middle point with amplitude 0.025.

Distance to the scene	1		
Scene depth	0.5		
Baseline length	0.1		
Image dimensions	352 × 288		
Calibration matrix	425	0	176
	0	425	144
	0	0	1

**Table 1:** Synthetic data setup.

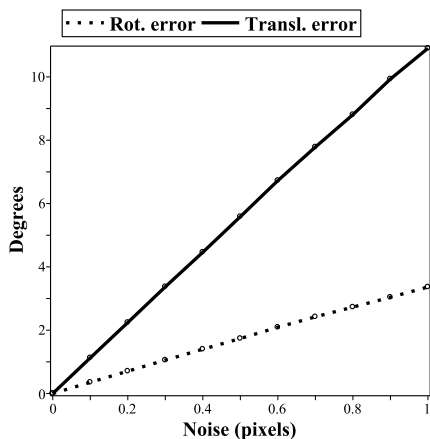


Figure 2: Rotational and translational errors relative to noise level.

We have measured the numerical error by the value

$$\frac{\|\mathbf{K} - \hat{\mathbf{K}}\|}{\|\hat{\mathbf{K}}\|},$$

where  $\hat{\mathbf{K}}$  is the ground truth calibration matrix,  $\|\cdot\|$  is the Frobenius norm. The distribution of the numerical error is reported in Figure 1, where the total number of trials is  $10^6$ .

The running time information for our implementation of the algorithm is given in Table 2.

Step	Projective reconstr.	Metric reconstr.
$\mu s$	7.9	28.4/root

Table 2: Average running times for the algorithm steps on a system with Intel Core i5 2.3 GHz processor.

In Figure 2, we demonstrate the stability of the algorithm under increasing image noise. We have added a Gaussian noise with a standard deviation varying from 0 to 1 pixel in a  $352 \times 288$  image. Each point is a median of  $10^6$  trials.

#### 4.1 Outliers

To test the algorithm in presence of outliers (incorrect matches), we have modeled a sequence of 70 cameras with centers on a circle, and 400 scene points viewed by all the cameras. For each image, we have added a Gaussian noise with one pixel standard deviation and 20% of outliers (uniformly distributed points in the image plane).

The auto-calibration algorithm was used as a hypothesis generator within a random sample consensus (RANSAC) framework [2]. For better computational efficiency we used the *preemptive* RANSAC from [9]. The motion hypotheses were scored by the Sampson approximation to geometric error [5]. The number of hypotheses was set to 400 for each camera position, and the preemption block size was set to 100.

The results are presented in Figure 3 and Figure 4. No iterative refinements were performed in the estimation. The calibration matrix averaged from the image sequence is as follows:

$$\mathbf{K} = \begin{bmatrix} 399.52 & 2.16 & 161.54 \\ 0 & 405.37 & 142.14 \\ 0 & 0 & 1 \end{bmatrix}.$$

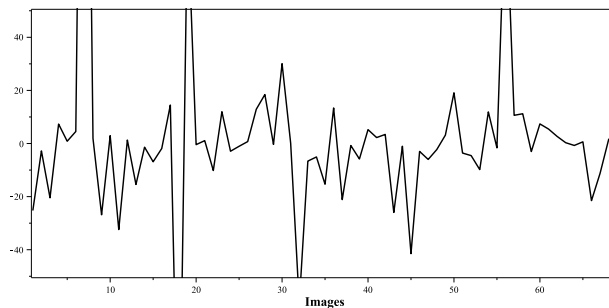


Figure 3: Skew parameter  $K_{12}$  estimated from the sequence of 70 synthetic images. Average value of  $K_{12}$  is 2.16.

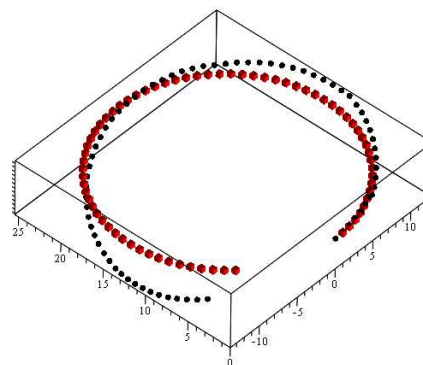


Figure 4: The camera track estimated from the sequence of 70 synthetic images. The red solid boxes are the ground truth camera positions.

## 5. CONCLUSION

A new non-iterative auto-calibration algorithm is presented. It derives the camera calibration from the smallest possible number of views and scene points. A computation on synthetic data confirms its accuracy and high speed performance. The algorithm is quite flexible. It is reliable, for example, even in case of pure rotations (baseline = 0), if the calibration matrix is only needed.

## 6. REFERENCES

- [1] Faugeras, O. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1993.
- [2] Fischler, M., Bolles, R. *Random Sample Consensus: a Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography*. Commun. Assoc. Comp. Mach., Vol. 24, 381–395, 1981.
- [3] Hartley R. *Estimation of Relative Camera Positions for Uncalibrated Cameras*. Proceedings of the 2nd European Conference on Computer Vision, Vol. 588 of Lecture Notes in Computer Science, 579–587, 1992.
- [4] Hartley, R.I. *Self-calibration from Multiple Views with a Rotating Camera*. Proceedings of the 3rd European Conference

on Computer Vision, Vol. 800–801 of Lecture Notes in Computer Science, 471–478, 1994.

- [5] Hartley, R., Zisserman, A. *Multiple View Geometry in Computer Vision. Second Edition*. Cambridge University Press, 2004.
- [6] Heikkilä, J. *Geometric Camera Calibration Using Circular Control Points*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 10, 1066–1077, 2000.
- [7] Maybank, S.J., Faugeras, O.D. *A Theory of Self Calibration of a Moving Camera*. International Journal of Computer Vision, Vol. 8, No. 2, 123–151, 1992.
- [8] Mendonca, P.R.S., Cipolla, R. *A Simple Technique for Self-Calibration*. Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 500–505, 1999.
- [9] Nistér, D. *Preemptive RANSAC for Live Structure and Motion Estimation*. Proceedings of the Ninth IEEE International Conference on Computer Vision, 199–206, 2003.
- [10] Quan, L., Triggs, B. *A Unification of Autocalibration Methods*. Proceedings of the Fourth Asian Conference on Computer Vision, 917–922, 2000.
- [11] Schaffalitzky, F., Zisserman, A., Hartley, R.I., Torr, P.H.S. *A Six Point Solution for Structure and Motion*. Proceedings of the European Conference on Computer Vision, Vol. 1, 632–648, 2000.
- [12] Sturm, P. *Critical Motion Sequences for Monocular Self-Calibration and Uncalibrated Euclidean Reconstruction*. Proceedings of the International Conference on Computer Vision and Pattern Recognition, 1100–1105, 1997.
- [13] Triggs, B. *Autocalibration and the Absolute Quadric*. Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 609–614, 1997.
- [14] Tsai, R.Y. *A Versatile Camera Calibration Technique for High-accuracy 3D Machine Vision Metrology Using Off-the-shelf TV Cameras and Lenses*. J. Robotics and Automation, Vol. 3, No. 4, 323–344, 1987.
- [15] Zhang, Z. *A Flexible New Technique for Camera Calibration*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 11, 1330–1334, 2000.

## ABOUT THE AUTHOR

Evgeniy Martyushev is an associated professor at South Ural State University, Department of Mechanics and Mathematics. His contact email is [mev@susu.ac.ru](mailto:mev@susu.ac.ru).

# A Novel Approach for Attention Points Extraction from Saliency Maps based on T-Junctions

Ekaterina Potapova, Michael Zillich, Markus Vincze  
{potapova,zillich,vincze}@acin.tuwien.ac.at  
Automation and Control Institute  
Vienna University of Technology

## Abstract

Object detection is of a vital importance in many computer vision and robotic tasks. Saliency maps can be used to extract attention points, that potentially indicate objects in the scene and therefore, can be further explored. For instance, attention points can serve as seed-points for object segmentation. However, it is important to guarantee the uniqueness and the quality of attention points. Therefore, the extraction of attention points is a challenging problem. In this paper, we propose a novel approach to extract attention points from saliency maps. Compared to several existing attention points extraction strategies, we show that the proposed strategy performs better in terms of the uniqueness of attention points and their proximity to the center of detected objects<sup>1</sup>.

**Keywords:** Visual Attention, Fixations, Saliency Maps.

## 1. INTRODUCTION

Object detection is a crucial task in many applications of computer vision such as in robotics. One way to detect objects is to generate saliency maps and extract fixation points, so-called attention points, from them. These fixations can potentially identify or detect objects and can be used as seeds for segmentation [7] and further recognition.

One of the main challenges of attention points extraction mechanisms is inability to guarantee the uniqueness and the quality of them. Therefore, attention points extraction remains a challenging process. An obvious approach to evaluate the quality of attention points is to compare them to fixation maps built on human data. However, when the task at hand is to extract attention points suitable for a specific task (*e.g.* attention-driven segmentation), comparison to fixation maps is not very useful.

In this paper, we compare several existing attention points extraction strategies and evaluate them on different types of saliency maps in terms of the uniqueness and proximity of fixations to the center of the detected object. The later is important, since Vishwanath *et al.* [11] showed that line of sight lands near the center of gravity of the object. We also propose a novel approach to extract attention points from saliency maps based on T-Junctions. This attention strategy can be applied for saliency maps in which several disjoint connected components are available. We show that the proposed strategy performs better than existing strategies in terms of the above criteria.

The paper is structured as follows: In the next section, we describe algorithms for saliency map calculation, subsequently, attention points extraction strategies are discussed in detail. The following

<sup>1</sup>The research leading to these results has received funding from the Austrian Science Fund (FWF) under grant agreement No. TRP 139-N23 InSitu and from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement No. 600623, STRANDS.

evaluation shows comparison of different strategies and in the end, we conclude our work with a discussion about future directions of research.

## 2. SALIENCY MAPS

We evaluated attention points extraction strategies on four different types of saliency maps: Attention based on Information Maximization (*AIM*) [1] (Fig.1,b), Graph-based visual saliency (*GBVS*) [3] (Fig.1,c), 2D Symmetry-based saliency (*SYM2D*) [5] (Fig.1,d), and 2.5D Symmetry-based saliency (*SYM2.5D*) [9] (Fig.1,e). In this section, we describe the basic principles of each saliency map.

### 2.1 Attention based on Information Maximization

Bruce *et al.* [1] proposed a model of bottom-up overt attention based on the principle of maximizing information sampled from the scene.

The model defines saliency by quantifying the self-information of each local image patch. Independent Component Analysis is performed on a large number of sampled patches to determine a suitable basis. The probability of observing a specific patch can be evaluated by independently considering the likelihood of each corresponding basis coefficient. Shannon's self-information measure, applied to the joint likelihood of statistics of the patch, provides an appropriate transformation between probability and the degree of information inherent in the local statistics. Therefore, saliency is determined as the self-information of each local image patch.

### 2.2 Graph-Based Visual Saliency

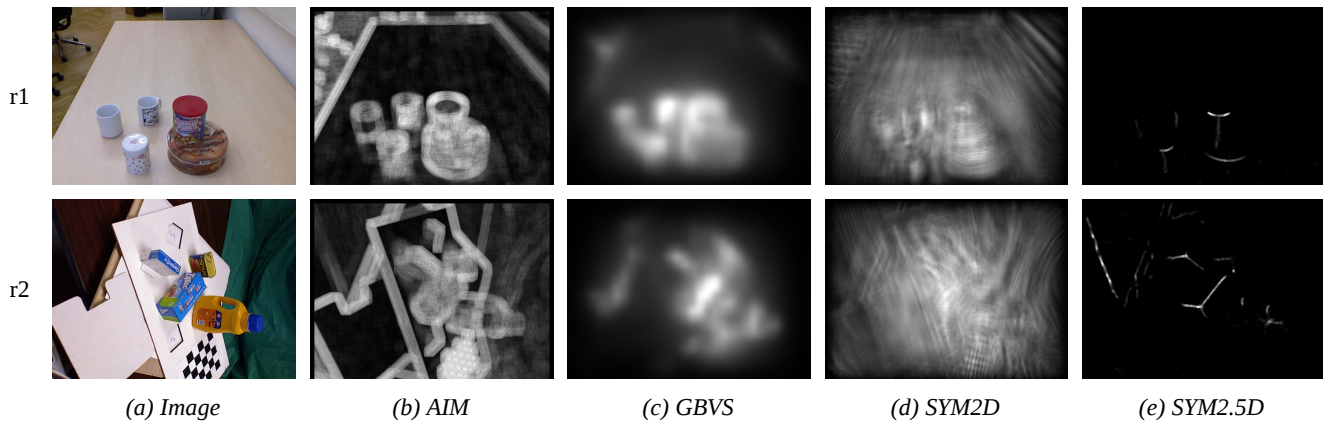
Harel *et al.* [3] described a simple and biologically plausible model for bottom-up saliency.

In the introduced model, activation maps are formed on certain feature channels (color, orientation), and then combined into the master saliency map. Activation maps are calculated using fully-connected directed graphs, where weights of the edges depend on the similarity between pixels they connect. Harel *et al.* [3] showed how to treat this graph as Markov chain. Therefore, activation map is an equilibrium state of the given Markov chain. Activation maps are normalized in a similar fashion using Markov chain, where the goal is to concentrate intensity of activation maps. Finally, normalized activation maps are combined into the master saliency map.

### 2.3 Saliency Map Based on 2D Symmetry

Kootstra *et al.* [5] proposed to use symmetry, one of the Gestalt principles for figure-ground segregation, to calculate saliency maps.

Saliency maps are built upon the local symmetry operator of Reisfeld *et al.* [10] and is extended to a multi-scale model similar to the contrast-saliency model [4]. A context free attentional



**Figure 1:** Examples of images and saliency maps (Attention based on Information Maximization (AIM), Graph-based visual saliency (GBVS), 2D Symmetry-based saliency (SYM2D) and 2.5D Symmetry-based saliency (SYM2.5D)) for TOSD (r1) and WILLOW (r2) database.

operator, described by Reisfeld *et al.* [10], based on the intuitive notion of symmetry. The amount of local symmetry at a point is calculated as the sum of similarity measures between pixel pairs in the symmetric kernel centered at the point. Two points in the kernel form a pair if the middle point of the line connecting them is the center of the kernel. The similarity measure takes into consideration gradient directions and magnitudes, as well as distances between points.

## 2.4 Saliency Map Based on 2.5D Symmetry

Potapova *et al.* [9] extended principles introduced by Kootstra *et al.* [5] to be used in 2.5D space.

In this approach, saliency is calculated as the amount of reflective symmetry in a local patch. Reflective planes are determined as planes perpendicular to the directions of the principal components. A local patch is divided into two sub-patches by a reflective plane. The reflective symmetry measure includes such characteristics as the mean distance between two sub-patches, difference in mean depth values, and collinearity between mean normal vectors.

## 3. ATTENTION POINTS EXTRACTION STRATEGIES

In this section, we will describe different strategies employed to extract attention points from saliency maps. We are going to discuss the following strategies: Winner-Take-All (WTA [6]), Maximum Salient Region (MSR [2]), and T-Junctions.

### 3.1 Winner-Take-All

Lee *et al.* [6] showed that the Winner-Take-All (WTA) neural network can be used to simulate humans behavior of scene components prioritization while observing a scene.

Excitatory input neurons in the network are independent and received from a saliency map and each neuron excites its corresponding WTA neuron. All WTA neurons evolve independently of each other. The “winner” is the one that fires first (*i.e.* reaches threshold). This triggers complete reset of all WTA neurons. Attention points can be defined as points of location of the “winner” neuron. Firing of the “winner” neuron is followed by the shift of the Focus of Attention (FOA) and the local inhibition. FOA is shifted to be at the location of the “winner” neuron. Input neurons are inhibited at the new location of the FOA. Local inhibition prevents returning the FOA to the just attended location and allows the next most salient location to become the winner. This process is called “Inhibition of Return” (IOR) and is described in [8]. All time constants, conductances, and firing thresholds used

in the WTA model implemented in this paper are the same as in [4]. Attention points are extracted using down-sampled saliency maps.

### 3.2 Most Salient Region

Frintrop [2] proposed the detection of the Most Salient Region (MSR) for object detection. The point with the maximum saliency value (attention point) determines the most salience region.

Starting from the attention point (seed), the surrounding salient region is extracted by means of seeded region growing. MSR consists of all neighbors of the seed with saliency values that differ by at most 25% from the saliency value of the attention point. The focus of attention (FOA) is directed to the MSR. After that the MSR is inhibited and the next MSR is computed. Though this way of attention points detection is less biologically plausible, Frinrop [2] argues that equivalent results are achieved with fewer computational resources.

### 3.3 T-Junction Attention Points

We propose a new T-Junction (TJ) attention points extraction strategy. This strategy requires saliency maps in which each object is detected as an individual blob (*i.e.* SYM2.5D). Attention points are defined as junction points of multi-segment skeletons, or as mid-points in the case of single segments. Skeletons are calculated from connected components of saliency maps.

## 4. EXPERIMENTS

Comparison of different attention points extraction strategies applied to different types of saliency maps was done with respect to two metrics. The first metric is the *Hit Ratio (HR)*, and the second metric is the *Distance to the Center (DC)*. Experiments were performed on two databases: *Table Object Scene Database<sup>2</sup> (TOSD, Fig.1,r1,a)* and *Willow Garage Table Objects Database<sup>3</sup> (WILLOW, Fig.1,r2,a)*. Objects in both databases were hand-labeled with polygon masks.

### 4.1 Hit Ratio

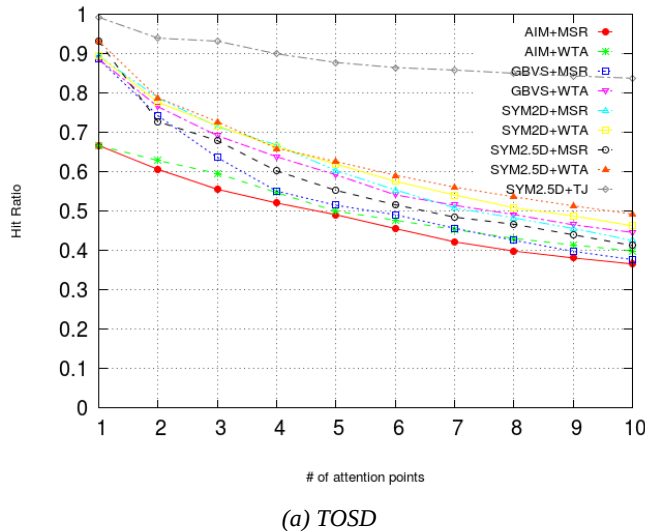
*Hit Ratio (HR)* shows how many different objects were covered by attention points with a given number of fixations, and is given by the percentage of unique attention points being situated inside different objects:

<sup>2</sup> <https://repo.acin.tuwien.ac.at/tmp/permanent/TOSD.zip>

<sup>3</sup> [http://vault.willowgarage.com/wgdata1/vol1/solutions\\_in\\_perception/Willow\\_Final\\_Test\\_Set/](http://vault.willowgarage.com/wgdata1/vol1/solutions_in_perception/Willow_Final_Test_Set/)

$$HR = n/N$$

where  $N$  is the number of fixations and  $n$  is the number of different attended objects. A perfect attention mechanism will hit every object exactly once, resulting in  $HR$  equaled to one.



**Figure 2:** Hit Ratio (HR) against the number of extracted attention points for: (a) TOSD, (b) Willow. HR results are shown for different combinations of saliency maps and extraction strategies. As can be seen from the plot, the combination of SYM2.5D [9] with T-Junction extraction strategy results in the best performance.

We evaluate  $HR$  against the number of fixations. Fig.2,a and Fig.2,b show  $HR$  results averaged over all images in *TOSD* and *WILLOW* respectively. As can be seen from the plot, for *TOSD* 2.5D symmetry-based saliency maps together with attention points extraction strategy based on T-Junctions result in the increase of performance up to 60% in terms of the number of detected objects. The second best performance is achieved by extracting attention points from 2.5D symmetry-based saliency maps using Winner-Take-All extraction strategy. These results show that 2.5D symmetry-based saliency maps capture the main structure of table scenes with man made objects and direct attention to those objects.

For *WILLOW* database combination of 2.5D symmetry-based saliency maps and T-Junctions based extraction strategy result in low scores with  $HR$  for the first few attention points with improvement in performance up to 50% with a larger number of attention points. In *WILLOW* database only objects standing on the table were labeled, and neither tables, nor cardboards are considered as objects. On the other hand, 2.5D symmetry-based saliency operator gives a strong response for regions that are symmetrical in 3D space, for *e.g.* box and table corners. Combination of those two factors result in a high false detection rate.

Furthermore, we can conclude from plots in Fig.2, that *WTA* extraction strategy shows better performance, than *MSR* extraction strategy for a given saliency map.

#### 4.2 Distance to the Center

The distance between the extracted attention point  $p$  and the center of the respective object  $c$  is defined as:

$$DC(p,c) = |p - c|_2$$

The centers of the respective objects represent physical centers of the visible parts of the objects. The  $DC$  shows the accuracy of attention points. The smaller the distance, the better is the detection quality of attention points.

Fig.3,r1-r2 and Fig.3,r3-r4 show evaluation of attention points with respect to the distance to object centers for *TOSD* and *WILLOW* respectively. Fig.3,r1 and Fig.3,r3 show evaluation results for attention points that were the first to detect objects. Fig.3,r2 and

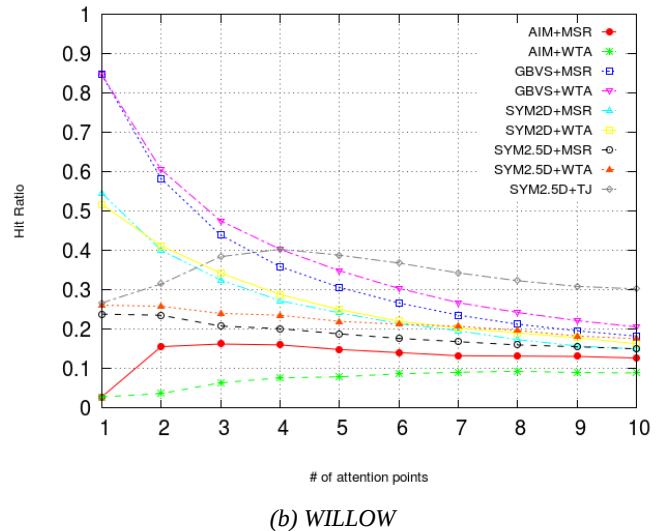
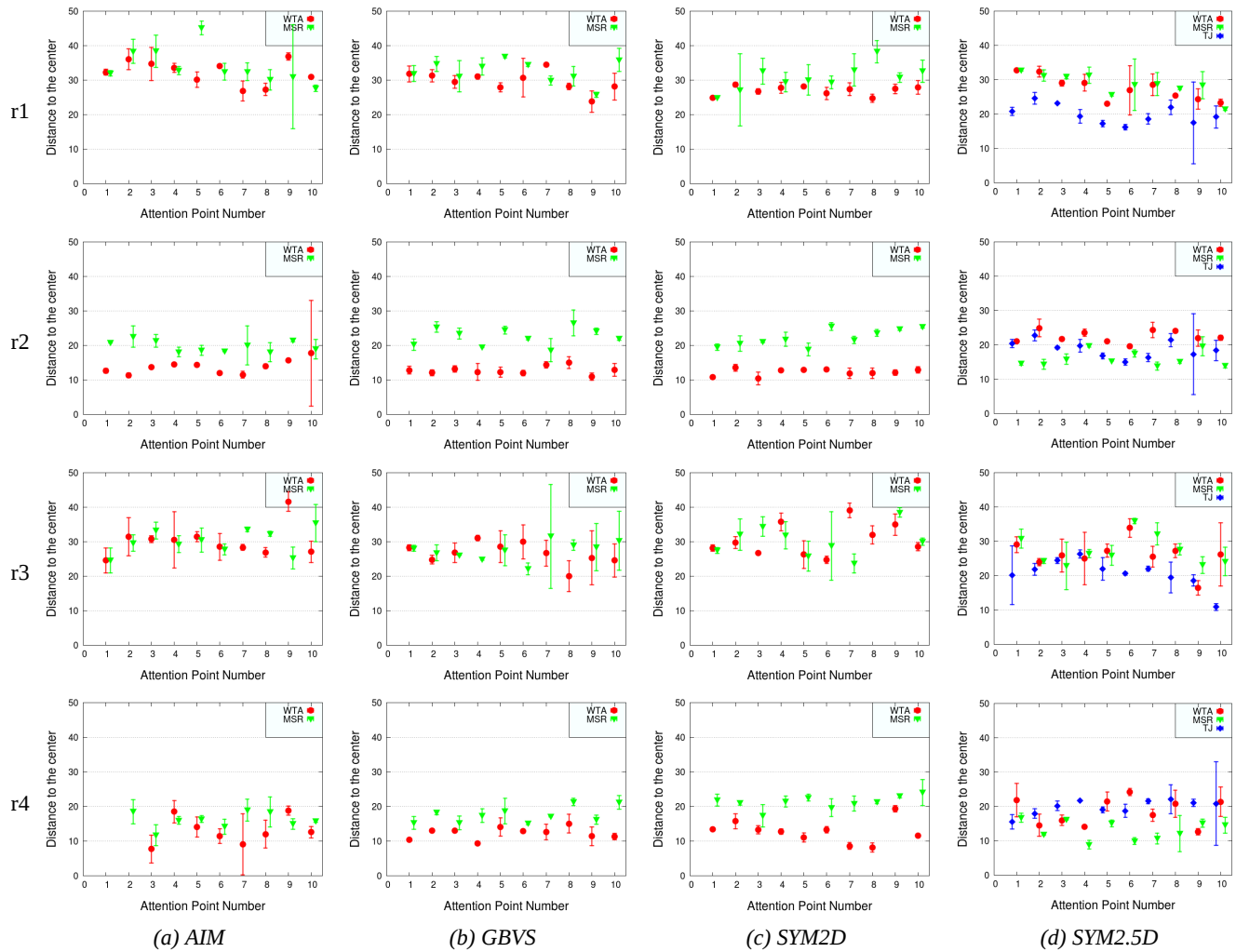


Fig.3,r4 show evaluation results for best attention points. The best detected attention point for an object is the one that is the closest to the center of the object.

A perfect attention mechanism would detect every object only once and directly at the center. As can be seen from plots in Fig.3 all detection strategies result in similar performance in terms of distance to the center for the first detected attention point. However, best attention points extracted by means of *WTA* strategy are situated closer to the center, than those detected by means of *MSR*. All extraction strategies show similar performance for 2.5D symmetry-based saliency map. This effect can be explained by the nature of 2.5D symmetry-based saliency in which salience blobs represent symmetry lines of objects. All described attention points extraction strategies detect attention points on these symmetry lines. Therefore, attention points are located close to each other.

## 5. CONCLUSION

In this paper we proposed a new strategy for extraction of attention points (*TJ*) and evaluated it against such strategies as Winner-Take-All (*WTA*) and Most Saliency Region (*MSR*) on several types of saliency maps (*AIM*, *GBVS*, *SYM2D*, *SYM2.5D*). Experiments show that the combination of 2.5D symmetry-based saliency map and T-Junctions extraction strategy performed up to 60% better than other combinations with respect to Hit Ratio criteria. Therefore, we can say that this combination can be specifically used to detect objects in cluttered table scenes containing man made objects. It was also shown that all extraction strategies give similar results with respect to the distance to the center of the object, when only first attention points are evaluated. However, the *WTA* strategy extracts attention points situated closer to the object center, when best attention points are evaluated. In future, we plan to modify the T-junction strategy to be applied not only for 2.5D symmetry-based saliency maps, but for other types of saliency maps as well.



**Figure 3:** Rows (r1) and (r3) show averaged distances from first attention points to the centers of detected objects for *TOSD* and *WILLOW* respectively. Rows (r2) and (r4) show averaged distances from best attention points to centers of the detected objects for *TOSD* and *WILLOW* respectively.

## 6. REFERENCES

- [1] Bruce, N.D.B., Tsotsos, J.K. 2009. Saliency, Attention, and Visual Search: An Information Theoretic Approach, *Journal of Vision* 9, 3.
- [2] Frintrap, S. 2006. *VOCUS: A Visual Attention System for Object Detection and Goal-directed Search*, vol. 3899 of *Lecture Notes in Computer Science*. Springer.
- [3] Harel, J., Koch, C., Perona, P. 2006. Graph-Based Visual Saliency. In *NIPS*, 545-552.
- [4] Itti, L., Koch, C., and Niebur, E. 1998. A Model of Saliency-based Visual Attention for Rapid Scene Analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 20, 11, 1254-1259.
- [5] Kootstra, G., Bergstroem, N., and Kragic, D. 2010. Using Symmetry to Select Fixation Points for Segmentation. In *20th International Conference on Pattern Recognition (ICPR)*, 3894-3897.
- [6] Lee, D. K., Itti, L., Koch, C., and Braun, J. 1999. Attention Activates Winner-Take-All Competition Among Visual Filters. *Nature Neuroscience* 2, 4, 375-381.
- [7] Mishra, A. K., and Aloimonos, Y. 2009. Active Segmentation with Fixation. In *IEEE 12th International Conference on Computer Vision (ICCV)*, 468-475.
- [8] Posner, M. I., and Cohen, Y. 1984. Components of Visual Orienting. *Attention and Performance X* 32, 531-556.
- [9] Potapova, E., Zillich, M., and Vincze, M. 2012. Local 3D Symmetry for Visual Saliency in 2.5D Point Clouds. In *11th Asian Conference on Computer Vision (ACCV)*, vol. 7724 of *Lecture Notes in Computer Science*. Springer, 434-445.
- [10] Reifeld, D., Wolfson, H., and Yeshurun, Y. 1995. Context Free Attentional Operators: the Generalized Symmetry Transform. *International Journal of Computer Vision* 14, 119-130.
- [11] Vishwanath, D., and Kowler, E. 2004. Saccadic Localization in the Presence of Cues to Three-Dimensional Shape. *Journal of Vision* 4, 445-458.

# Dependency of detectors' and descriptors' efficiency on image resolution for SIFT, SURF and ORB

Victor Eruhimov, Anna Kogan

Itseez Ltd, Nizhny Novgorod, Russia

Department of Computational Mathematics and Cybernetics  
Nizhny Novgorod State University, Nizhny Novgorod, Russia

victor.eruhimov@itseez.com, anna.kogan@itseez.com

## Abstract

In this paper we present a method for evaluation of dependency of object recognition efficiency by state-of-the-art methods on initial image size. By means of this method comparison of efficiency of various state-of-the-art image detectors and descriptors is made, and parameters providing best quality of ORB method are found.

**Keywords:** *detectors, descriptors, performance evaluation.*

## 1. INTRODUCTION

Detectors and descriptors of images are widely applied in the field of computer vision in such tasks as recognition of objects on images, stitching, visual odometry, splitting of objects into categories, creation of augmented reality systems and other tasks.

Often in practice it is necessary to analyze images of low resolution or quality. It happens, first, because of technical capabilities of the cameras installed, for example, on robots. Secondly, big dimension of images demands big computing expenses and memory sizes for information storage. We will investigate dependence of quality of recognition on resolution of the image and will try to find existing methods that allow achieving the best results at low resolutions.

In this work some detectors and the descriptors possessing high quality of recognition of the image are chosen. We will compare their efficiency using the benchmark offered by us. Criterion for comparison is the recognition accuracy, i.e. percent of the correct matches between keypoints among all established matches. For detectors we also will consider repeatability of detection. We will try to find parameters for the detector and a descriptor ORB providing high precision of recognition.

### 1.1 Related work

A huge number of affine regions detectors and local descriptors exist nowadays. To limit the comparison of quality made in this paper, we study only methods that seem to outperform others on recognition accuracy. First is SIFT [4], presented in 1999 by David Lowe, which is generally used as a benchmark when speaking about object recognition. Also we study SURF [1] which has been widely applied in computer vision and showed good performance. Third method to study is ORB [9], a new method presented in 2009 by group of researchers (Ethan Rublee, Vincent Rabaud, Kurt Konolige and Gary Gary Bradski). It is designed to perform good recognition quality in real-times.

Evaluation of detectors' and descriptors' efficiency on has been a popular topic of investigation in computer vision recently. [6] contains an exhaustive comparison of state-of-the-art affine region detectors in context of object recognition, introducing a method to compare efficiency on which we base in our studies. The paper studies dependencies of detectors' repeatibilities on

different scene transformations, such as changes of viewpoint, illumination, scale, blur, JPEG compression. No single method seems to outperform others in all the situations.

[5] presents an extensive research on quality of local descriptors and present a novel descriptor GLOH, based on SIFT method. The research shows GLOH and SIFT to give the best performance almost in any case.

In previous works only a little attention was given to dependency of recognition quality on initial image size. We evaluate the dependency in current work.

### 1.2 Overview

In chapters 2 and 3 we review existing state-of-the-art detectors and descriptors. Chapter 4 describes implementation details of a benchmark and dataset used for comparison. We present results of conducted experiments in chapter 5 and discuss them in chapter 6.

## 2. DETECTORS

Detector is an algorithm that finds image areas which contain some special points, constant characteristics of the image. These characteristics have to be robust to changes in the scale, noise and illumination of images to make detector repeatable, i. e. for detector to find keypoints with the same elements on different images.

The area found by the detector is represented by circle and is characterized by three parameters: coordinates of center, radius and angle of rotation.

### 2.1 SIFT (Scale-invariant feature transform)

SIFT is considered to be a reference algorithm since it possesses one of the highest repeatibilities among detectors.

SIFT consists of several stages, on each of which the number of considered areas decreases. Thus operations with big computing complexity are applied only to the areas which have passed preliminary tests.

- 1) Detection of the characteristic points robust to scaling of the image using scale-space extrema in Difference-of-Gaussians images.
- 2) Determination of radius of the area including the characteristic found earlier.
- 3) Determination of keypoint orientation.

### 2.2 SURF (Speeded Up Robust Feature)

SURF was conceived as more productive replacement to already existing effective detectors, for example, SIFT detector.

The algorithm is in many respects similar to SIFT. However, in SURF are realized rather exact and fast approximate calculations methods. It allows SURF to be 2–3 times faster than SIFT without significant quality loss.



## 2.3 ORB (Oriented FAST and Rotated BRIEF)

ORB also was designed as faster alternative of SIFT and SURF. ORB detector is based on FAST detector of corners presented in 2006 by Edward Rosten ([7], [8]). Center of a circle is considered to be a vertex of angle, if not less than  $\frac{1}{2}$  and no more than  $\frac{3}{4}$  the pixels lying on a circle are significantly more intensive than central pixel (or vice versa). In practice radius of circle equals 9 or 16 pixels.

ORB is modification of FAST which also calculates orientation (angle of rotation) of each corner. Besides, ORB looks through several scales of the initial image.



**Figure 1:** Definition of corners by means of the detector FAST with the radius of a considered circle of 3 pixels. Intensity of the pixels lying on a dashed line are less than central pixel  $p$  intensity by a set threshold  $t$ ,  $0 < t < 255$ .

## 3. DESCRIPTORS

Descriptor is a set of numbers characterizing a keypoint on the image. The descriptor is to have diverse values for keypoints containing different objects, and identical values for same objects, and to be robust to changes of lighting, affine and perspective transformations, noise and other image deformations. Thus, the descriptor is to use some invariant characteristics of the image.

### 3.1 SIFT

SIFT descriptor uses the algorithm based on model of human vision. Intensity gradients of pixels in 16 squares of  $4 \times 4$  pixels are summed and splitted into 8 bins. To store the sum of gradients in a  $4 \times 4$  square, we need 8 numbers describing magnitude of gradient in each of 8 directions, so the descriptor is stored by a  $16 \times 8 = 128$ -component vector. Big dimension of a vector causes difficulties in search of the descriptor closest to a selected one.

### 3.2 SURF

SURF descriptor of is based on the same algorithm as SIFT descriptor. However, for reduction of calculations volume SURF descriptor is stored by a 64-component vector. SURF contains some other improvements aimed on reduction of time of comparison of couple of descriptors.

### 3.3 ORB

ORB descriptor represents modified BRIEF descriptor ([2]).

BRIEF descriptor describes image area by means of a set of binary intensity tests in a  $31 \times 31$  image patch. 256 couples of pixels chosen by a fixed pattern on a patch are compared by intensity. If first pixel in a couple is brighter than the other, we store 1 to describe the couple, otherwise — 0. Thus we get 256 binary digits describing all the couples.

The descriptor of BRIEF is unstable to image rotations on more than  $10^\circ$ . ORB is its modification robust to rotation of a scene. ORB detector is also called rBrief - BRIEF robust to rotations.

## 4 BENCHMARK

In this chapter we discuss a benchmark for comparison of detectors and descriptors efficiency: how experiments are made, images used, criterion of correctness of recognition. We will use realizations of detectors and descriptors from OpenCV library.

Benchmark is designed to evaluate efficiency of various detectors and descriptors on image characteristics, such as size. While other benchmarks used to compare recognition efficiency an approximate homography matrix is used (computed on feature matches), we use exact homography matrix as with it we synthetically generate the image of object from the scene image.

### 4.1 Studied images

For establishment of dependences we will make series of experiments. Each experiment corresponds to a pair of resolutions of object and a scene.

In this work the object is represented by a part of a scene to which homography transformation is applied. Thus, we know the real homography converting object to a scene part.

Note that homography transformation can be applied to describe transform between two images, only if the translation between cameras used to get these images is much less, than distance between the cameras and a scene. In our case we use homography as some approximation of real distortions ([5], [6]).

Examples of objects and scenes are presented in fig. 2.



**Figure 2.** Examples of the images participating in experiments: images on the left — initial scenes, on the right — objects.

Minimum resolution of object is not less than 9000 pixels. 10–18 various images were used for experiments.

### 4.2 Repeatability

On the first stage we will detect areas on object and a scene. We will evaluate repeatability, the main quality characteristic of the detector — ability to detect same element on two and more various images. For calculation of repeatability we will use the following formula:

$$\text{repeatability} = \frac{\# \text{overlapping keypoints}}{\# \text{keypoints on object}}$$

Since we know how the real transformation for the scene and the object (ground truth homography matrix), we can project keypoints from scene to object and estimate overlap. We consider keypoints from different images to be equivalent if area of their overlap is not less than 50% of area of their union.

### 4.3 Accuracy

For detected keypoints we compute descriptors, and for each keypoint from object we find a match on a scene with the closest descriptor.

We can estimate a relative pose of corresponding keypoints again. Similarly, we will consider match to be correct if keypoints overlap is not less than 50% of their union. Thus we can determine recognition accuracy:

$$\text{repeatability} = \frac{\# \text{correct matches}}{\# \text{all matches}}$$

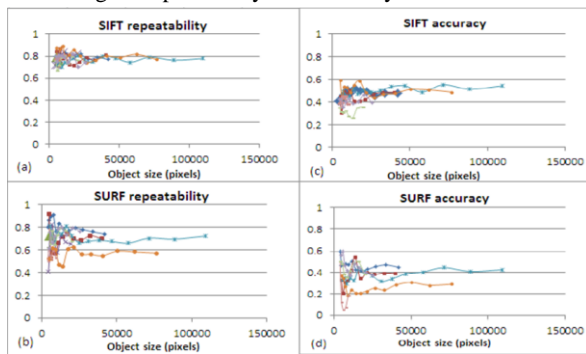
## 5 EXPERIMENTAL RESULTS

In this chapter we will discuss obtained results and we make conclusions on quality of various detectors and descriptors.

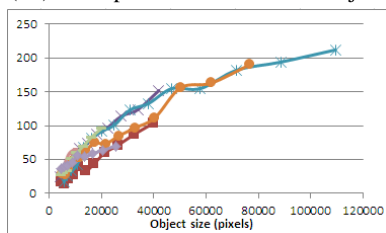
### 5.1 SIFT, SURF

Dependencies of repeatability and accuracy of SIFT and SURF on image resolutions are shown in fig. 3. Each of curves corresponds to a specific test image pair.

It is seen that both repeatability and accuracy of SIFT and SURF have almost no dependency on resolution of the image as corresponding curves are almost horizontal. At low resolutions on there are the some fluctuations caused by small absolute quantity of areas on images. Conducted measurements confirm that SIFT possesses higher repeatability and accuracy than SURF.



**Figure 3.** Dependence of repeatability of SIFT (3a) and SURF (3b) detectors, dependency of accuracy of SIFT (3c) and SURF (3d) descriptors on resolution of object.



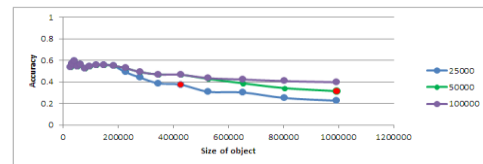
**Figure 4.** Dependence of absolute number of correct matches on resolution of the image (SIFT).

The conclusion that neither accuracy nor repeatability do not depend on image resolution was not an expected result: we expected to gain direct correlation between them. The more is resolution of the image, the more information it brings to us and to detector; despite that, keypoints detected by SIFT and SURF appear to be scalable, and percent of correct matched doesn't significantly change from scale to scale. Note that the absolute number of found keypoints linearly increases with increase of the image resolution, as shows fig. 4.

### 5.2 ORB

In ORB implementation in OpenCV we can find method parameters significantly affecting the algorithm quality. First is *nfeatures* – number of keypoints with maximal response than are returned by detector; all the other keypoints are rejected. By default it equals 500. At such *nfeatures* value it is possible to receive recognition accuracy not higher than 20%.

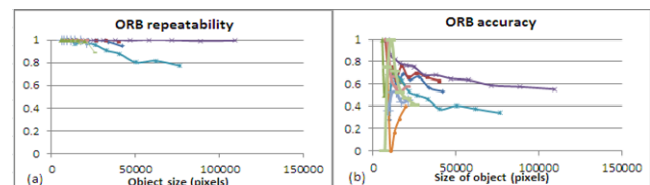
We found that to achieve maximal quality it is needed to leave all the keypoints in consideration, even with the small response. Dependency of recognition accuracy on resolution for different *nfeatures* values given in fig. 5. Therefore experiments were made with the *nfeatures* value of 150000 – such to exceed quantity of the recognizable areas on scenes of the highest resolutions. Note that this, of course, affects computation times, which we don't study in this work.



**Figure 5.** Dependence of accuracy of ORB on resolution of the image for different *nfeatures* (corresponding to different curves).

Note that at low resolutions of ORB detects very small quantity of areas. That is caused by an analog of FAST detector ORB uses: circle of 16 radius must be applied to detect corner in the center of a circle; on small images only a few such circles may fit the image. It causes poor quality of recognition at the lowest resolutions. However with increase of resolution of object ORB starts showing a good accuracy of recognition. On that reason we run experiments only on images with size of object not less than 9000 pixels.

Dependence of repeatability and accuracy of ORB on resolution of the image are given in fig. 6. All parameters are default, except *nfeatures* value.



**Figure 6.** Dependence of repeatability of the ORB detector repeatability (8a) and ORB descriptor accuracy (8b) on resolution of the image.

It is visible that ORB detector (unlike SIFT or SURF) possesses repeatability of almost 100% — highest of repeatabilities of considered detectors. ORB accuracy also seems to be the highest among the studied methods, which is surprising as SIFT was always thought to be the best recognition method.

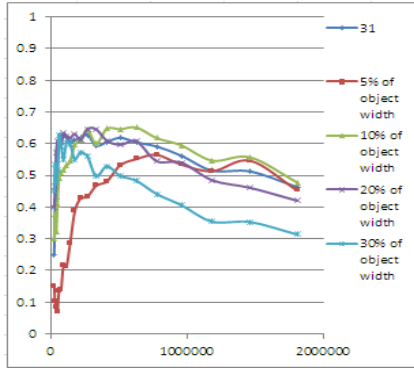
Almost absolute repeatability of ORB is achieved not only by high ability to detect same objects (corners, as ORB detects corners using oFast), but also by huge amount of found keypoints. Indeed, ORB detects up to 40 times more keypoints than SIFT or SURF. The keypoints are strictly located around the corners on image, but may have different radiuses and slightly different centers and orientations. Usually there are 5–15 keypoints around a single corner.

Besides, an inverse dependency on resolution of the image is observed. There may be a number of root causes for this effect, but the main one is that rBrief descriptor by default uses a fixed patch size of 31x31 pixels to calculate the descriptor. A fixed size of

patch on which a descriptor is computed is not a good solution to serve both for big and small images, and thus big and small key-points.

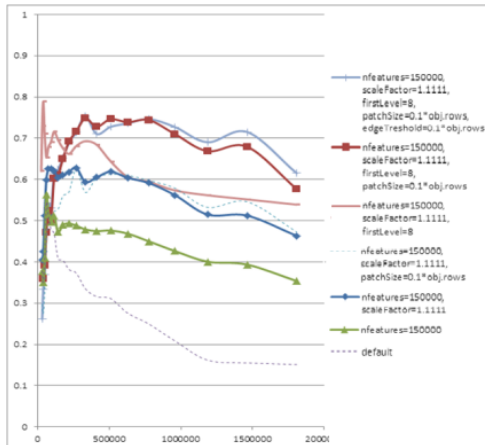
To avoid it, we can set *patchSize* parameter of ORB to depend on image size: for example, 5% or 10% of object width or height. In this case we get more slight and non-linear inverse dependency on image resolution, as seen on fig. 7.

The value of *patchSize* giving the best accuracy results is 10% of object width. Inverse dependency on image resolution is still present, but now shows off only on the biggest resolutions.



**Figure 7.** Dependence of accuracy of ORB on resolution of the image for different *patchSize* values (corresponding different curves).

Other parameters of ORB also can affect quality significantly, like *edgeThreshold* – size of image border in which no detection is made for noise reduction. To study it, we made experiments with various parameters of ORB detector and descriptor. Results of these experiments are presented in fig. 8.



**Figure 8.** Dependence of accuracy of ORB on resolution for different parameters.

The combination of parameters providing the maximum accuracy:  
 • *nfeatures* = 150000 (as big as possible not to reject any key-points);<sup>TM</sup>

• *scaleFactor* (coefficient of reduction of height and image width for receiving a pyramid) = 1.1111;

• *firstLevel* (the first level of a pyramid used for detecting) = 8;

• *patchSize* (the size of a patch of rBrief) = 10% from object height;

• *edgeThreshold* = 10% from object height aren't detected.

This combination of parameters allows to receive the accuracy of recognition of 65-70% higher than ORB accuracy when using parameters by default.

## 6 DISCUSSION

In this work we presented a method to evaluate the dependency of objects recognition accuracy by SIFT, SURF and ORB methods on resolution of the initial image. By means of this method we compared efficiency of various detectors and descriptors of images. We managed to choose the parameters of the ORB method providing the best quality of recognition of the image.

Accuracy and repeatability of SIFT and SURF don't depend on resolution of the studied image. SIFT possesses slightly higher quality of recognition than SURF. Neither SIFT nor SURF show dependency of quality on image size.

ORB seems to outperform both SIFT and SURF. Repeatability and accuracy of ORB have weak inverse dependency on resolution of the image: at high resolutions ORB has 10-15% smaller accuracy, than at low. To investigate the root cause of this effect we experimented with parameters of this method and managed to find optimal parameters that are able to significantly increase quality of object recognition. In that case, 640x480 pixels (300000 pixels) resolution is enough to achieve 65-70% recognition accuracy.

Obviously, the comparison carried out by us isn't full. We used 10-20 images of various categories, however their quantity can be increased for receiving more reliable results. Research of this sort can be applied to various detectors and descriptors, and also to various combinations of detectors and descriptors.

## 7 REFERENCES

- [1] Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool. "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346-359, 2008.
- [2] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features // European Conference on Computer Vision, 2010.
- [3] Martin A. Fischler and Robert C. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". June 1981, Comm. Of the ACM 24: 381—395.
- [4] David G. Lowe, "Distinctive image features from scale-invariant keypoints", International Journal of Computer Vision, 60, 2 (2004), pp. 91-110, 2004.
- [5] K. Mikolajczyk, C. Schmid. "A performance evaluation of local descriptors", IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(10): pp. 1615-1630, 2005.
- [6] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir and L. Van Gool. "A comparison of affine region detectors", International Journal of Computer Vision, 65(1/2): pp. 43-72, 2005.
- [7] Edward Rosten, Tom Drummond. Fusing points and lines for high performance tracking // IEEE International Conference on Computer Vision, 2005.
- [8] Edward Rosten, Tom Drummond. Machine learning for high-speed corner detection // European Conference on Computer Vision, 2006.
- [9] Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary Bradski. ORB: An Efficient Alternative to SIFT or SURF // International Conference on Computer Vision, Barcelona, Spain, November, 2011.

# “Particle” Approach for Flame Front Propagation Fast Modeling in the Given Premixed Gas Flow Field

Sergey Minaev<sup>1</sup>, Boris Dolgovesov<sup>2</sup>, Boris Mazurok<sup>2</sup>

<sup>1</sup> Engineering School, FEPU, Vladivostok, Russia

<sup>2</sup> Institute of Automation and Electrometry, SB RAS, Novosibirsk, Russia  
 bsd@iae.nsk.su

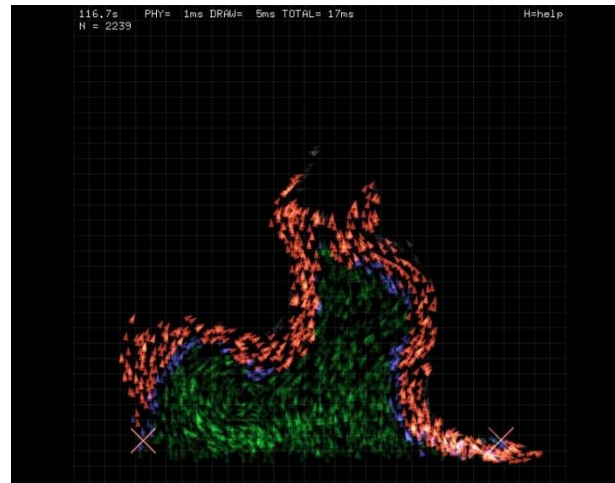
## Abstract

The objectives of the present study are the formulation of the models of flame front evolution capturing the effects of flame propagation in the prescribed non-stationary premixed gas flow field. The algorithm of real time simulations of flame evolution within frame of “flamelet” approach is offered to model shape of flame front propagating in the given non stationary gas flow fields. The algorithm is based on “particles” method and representation of the burned and unburned gases by a two sets of different particles, moving on the fixed trajectories, corresponding to the given flow field.

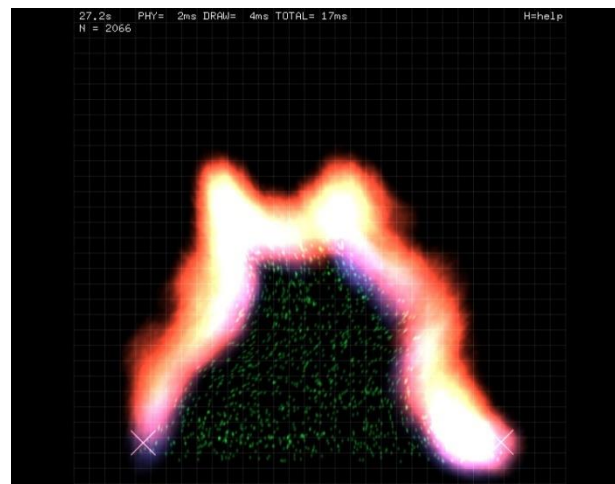
**Keywords:** flame modeling, numerical simulation.

## 1. INTRODUCTION

Nowadays the development of effective algorithms of simulations of the premixed flame evolution in the practical devices having complex geometry and nonstationary gas flows is issue of the day. As numerical modeling of gas combustion with detailed chemical kinetics and real gas flows demands huge computing expenses there is a necessity to apply more simple, but not losing the physical meaning, the simplified models of flame evolution. In a case when characteristic scale of internal structure of a combustion wave much less than characteristic size of gas flow perturbations it is reasonable to use the simplified representation of a flame as an interface between unburned and burned gasses. This model assumes that flame surface moves along its normal with given velocity called a burning velocity. The value of burning velocity may be estimated by preliminary simulation of 1D combustion wave taking into account detailed kinetics of chemical reactions. In this approximation the gas flow is determined by equations free of effects introducing by combustion. This approximation is known as “flamelet” model [1] and it is widely used in engineering simulations. In the simplest case one can assume that the burning velocity is constant and it is determined by mixture content only. In the present work the algorithm of real time simulations of flame evolution within frame of “flamelet” approach is offered to model shape of flame front propagating in the given non stationary gas flow fields appearing in the combustion chambers with complex geometry. The algorithm is based on representation of the continuous media as set of a large number of elementary particles, moving on the fixed trajectories, corresponding to given flow field. Each particle is characterized by spatial coordinates and two states corresponding to burned and unburned gases, respectively. The flame front is considered as a boundary dividing two sorts of the particles. The formulation of the model and the algorithm of the calculations is described in the next section. The figure 1 shows results of the method.



**Fig. 1:** Turbulent flame touch simulated by particles based real time algorithm. Green and red particles correspond to the unburned and burned mixture, blue particles marks the flame



**Fig. 2:** Real time post effect flame simulation.

In many practical devices with gas combustion the flame thermal thickness is small in comparison with the sizes of the combustion room. Therefore flame may be considered as a surface dividing unburned and burned mixtures. This surface is characterized by ability to move on unburned mixture with constant burning velocity  $U_b$  [5]. The burning velocity is considered to be known from theoretical evaluations or experimental data. Writing the flame front equation in the form  $F(x, t) = 0$  and assuming that flame front propagates in the unburned gas with given flow

velocity field  $V(x, t)$  one can obtain the flame front evolution equation in the form

$$\frac{\partial F}{\partial t} - \left( \vec{V}, \vec{\nabla} F \right) = U_b \left| \vec{\nabla} F \right| \quad (1)$$

In the particular case of flame front equation  $z = f(x, y, t)$  and velocity vector  $V = (0, 0, W(x, y))$  equation (1) has a form

$$\frac{\partial f}{\partial t} - W = U_b \left( \left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2 + 1 \right)^{1/2} \quad (2)$$

It is interesting to note that this equation is similar to the Hamilton-Jacobi equation describing propagation of the relativistic particle in the field with potential  $U(x, y)$  [4]:

$$\frac{1}{c_0^2} \left( \frac{\partial S}{\partial t} + U \right)^2 - \left( \frac{\partial S}{\partial x} \right)^2 - \left( \frac{\partial S}{\partial y} \right)^2 = m^2 c_0^2 \quad (3)$$

Flame front evolution in the case of resting gas may be described on the base of the Huygens principle applying in the ray optics. Let us suppose that flame front shape is known at some moment, for instance at  $t = 0$ . In order to find the flame front shape at subsequent time moment  $t = dt$  according to the Huygens principle the set of spheres of radius  $U_b dt$  with its centers belonging to the initial surface should be plotted. The envelope curve of this set of spheres represents the flame shape at  $t = dt$ . Applying this principle for subsequent time intervals the flame front evolution can be tracked. In the case of the moving fluid

with given velocity field  $\vec{V}(\vec{x}, t)$  the technique based on the Huygens principle can be applied too. The only difference is that all points of initial surface must be previously shifted on vector  $\vec{V}(\vec{x}, t) dt$ . Proposed algorithm assumes that continuum is filled by the finite number of randomly distributed particles moving with velocity equal to the local velocity in corresponding 2-D point. Each particle is characterized by spatial coordinates and value describing the state of the corresponding elementary gas volume, namely unburned or burned. Flame front defines as the interface separating the regions field by unburned and burned gases. More specifically, the flame front may be defines in the following manner. Let us divide the calculation domain on uniform cells so that each cell comprises at least two particles. Lets calculate difference between numbers of unburned and burned particles for each cell. Finally, lets single out the cells for which this difference is less than some fixed number. The flame front may be described by any surface passing through these cells. If the average distance between neighboring particles is  $\delta$  the size of the square in 2D case must be not less than  $2\delta$  and the accuracy of determination of the flame front position depends on the cells size. The following algorithm is proposed for numerical simulations of the flame front evolution. On each time step all unburned particles located inside the circles centered at the burned particles are replaced by the burning particles. The procedure of particles properties changing is applied at every fixed time step  $\tau$  and the circles radius is  $U_b \tau$ . From physical point of view, this algorithm models ignition of the fresh mixture volumes by hot burned gas. Dependency of the velocity of self-propagating wave of ignition on average distance between neighboring particles is discussed in the next section. Calculation of one algorithm iteration for system of  $N$  particles demands implementation of  $O(N^2)$  operations for checking the particles belonging to the igniting circles. However, usage of the particle spatial coordinates

sorting method allows to reduce calculation cost to  $N \cdot \log N$  operation per one time step.

This acceleration is achieved by sorting of the particles in cells and neglecting certainly far located particles during the search of the particles belonging to the ignition circle. The sorting results are also useful in determination of the flame front surface. Utilization of the memory pool for data structures describing particles also leads to the calculations acceleration.

## 2. RESULTS

Flame front burning velocity on the non-uniform grid of particles depends on average distance between particles. For simplicity, we consider the 1D grid case. The fluctuation of average distance between two neighbor particles  $d$  can cause the decrease of value of burning velocity in numerical simulations. Numerical experiments demonstrated that, for example, error value of 1% of burning velocity corresponds to 50 particles placed into the "ignition" radius  $U_b \tau$ . We also carried out a numerical experiment to measure the velocity of the outward propagating cylindrical flame to investigate the influence of fluctuation of partial density on flame characteristics. Numerical experiments shown that flame front speed did not depend on its radius and the proposed algorithm keeps the spatial isotropy.

This algorithm was applied to simulate the flame propagating in non-uniform non-stationary flows. The implemented algorithm allows interactively change number of the particles, the velocity field, and gives possibility for interactive ignition and quenching of flame in the real-time. The figure 1 shows results of simulations of flame front propagating in constant Poiseuille flow with randomly distributed non-stationary point vortices. This configuration roughly corresponds to a turbulent flame torch. The proposed algorithm provides 25 fps on the one core of Pentium Dual-Core 2.5GHz with particles number  $N = 20000$ . The figure 2 shows the result of real time post computation for more clear flame presentation.

## 3. CONCLUSION

In the paper the numerical algorithm describing flame front evolution within frame of "flamelet" model is suggested. This algorithm assumes the representation of the continuous medium by set of discrete particles. The developed algorithm can be generalized in future to the model that takes into account self-ignition and flame quenching mechanism. For this aim one can apply the flame front model that incorporates flame front inertial effects and the dependency of flame front velocity on the local flame front curvature [3]. The work was partially supported by cooperative grants of Siberian Branch of the Russian Academy of Sciences and Collaborative Research Project J11028 IFS Tohoku University.

## 4. REFERENCES

- [1] K.N.C. Bray, Turbulent Reacting Flows. Topics in applied physics, 44, (1980), pp.115-183.
- [2] R. Hockney, J. Eastwood, Computer simulation using particles, Mir, Moscow, (1987).
- [3] S. Minaev, K. Maruta, R. Fursenko, Combustion Theory Modelling, 11(2), (2007), 187
- [4] G. Sivashinsky, Lett. Nuovo Cimento, 77A(1), (1983), pp.21-38.
- [5] Ya.B. Zeldovich et al., Mathematical theory of combustion and explosions, Nauka, Moscow, (1980).

### **About the author**

Sergey Minaev (professor) is a head of Laboratory at Far Eastern Federal University. His contact email is [minaevss@yahoo.com](mailto:minaevss@yahoo.com)

Boris S. Dolgovesov (Ph.D.) is a head of Synthesizing Visualization Systems Laboratory at Institute of Automation and Electrometry SB RAS. His contact email is [bsd@iae.nsk.su](mailto:bsd@iae.nsk.su)

Boris S. Mazurok is a scientific researcher of Synthesizing Visualization Systems Laboratory at Institute of Automation and Electrometry SB RAS. His contact email is [boris@albatros.iae.nsk.su](mailto:boris@albatros.iae.nsk.su)

# Improved analysis of membrane potential oscillations in the network of cells from islet of Langerhans

Denis Špelič<sup>1</sup>, Jurij Dolensek<sup>2</sup>, Andraž Stožer<sup>2,3</sup>, Marjan Slak Rupnik<sup>2,3</sup>, Borut Žalik<sup>1,3</sup>, Domen Mongus<sup>1,3</sup>

<sup>1</sup> Faculty of Electrical Engineering and Computer Science, University of Maribor, Smetanova 17, SI-2000 Maribor, Slovenia

<sup>2</sup> Institute of Physiology, Faculty of Medicine, University of Maribor, Slomškov trg 15, SI-2000, Maribor, Slovenia

<sup>3</sup> Centre for Open Innovations and Research, University of Maribor, Slomškov trg 15, SI-2000, Maribor, Slovenia

{denis.spelic, jurij.dolensek, marjan.rupnik, borut.zalik, domen.mongus}@uni-mb.si; {stožera}@gmail.com

## Abstract

This paper proposes a new method for the visualisation of changes in the membrane potentials of biological cells, where the low image resolution and high presence of noise within an original series of images (i.e. a video) prevents accurate observations of this natural phenomenon. This method applies an alpha-trimmed mean filter within time-domain for decomposing a series into high and low frequency bands.  $\Theta$ -mapping is then applied in order to identify and denoise the signal. Finally, the output images are constructed by suppressing the low-frequency band, whilst high-frequencies are emphasised. As shown by the results, visualisation is significantly improved in this way, whilst the signal-characteristics remain well-preserved.

**Keywords:** *Scientific visualisation, signal processing,  $\Theta$ -mapping, islet of Langerhans, beta cell, membrane potential.*

## 1. INTRODUCTION

Islets of Langerhans are comprised of beta cells that secrete the anabolic hormone insulin [5]. Insulin is secreted into the circulation interprandially (i.e. between meals), keeping at bay the catabolism of bodily energy reserves, as well as postprandially (i.e. after meals) when the concentration of nutrients in the plasma increases, thus enabling the storage of energy-rich molecules in the forms of liver and muscle glycogen as well as fat-tissue triglycerides, and promoting protein anabolism [18]. The absence of insulin due to the destruction of islets but also a partial lack of its effects due to the resistance of tissues to insulin or a decreased secretion of insulin all result in a severe and chronically disabling clinical picture known as diabetes mellitus [2]. The global prevalence of diabetes mellitus in people aged 20-79 years was 8.3 % in 2011, consuming on average 11% of national healthcare budgets. However, the global prevalence in this age group is expected to increase by 50 % by 2030, thus presenting in depth motivation for researchers and their funders [19].

Islets present 1-2 % of the pancreas in mass and volume. They have a spherical structure (100-300  $\mu\text{m}$  in diameter) and are comprised of a few thousand cells per islet [6]. Glucose-responsive and insulin-secreting cells named beta cells comprise the highest fraction of the cells inside the islets of Langerhans [7]. The cells are functionally coupled via connexin36 molecules forming a syncytium of cells [15]. Due to the current consensus model, glucose and other nutrients enter the beta cell, are then metabolised yielding ATP that decreases the open probability of a specific channel protein conducting  $\text{K}^+$  in an ATP-dependent manner ( $\text{K}_{\text{ATP}}$  channels) and depolarises the cell membrane [1,4]. Following the initial depolarisation, the membrane potential (MP) in the cells oscillates in the form of bursts with superimposed high frequency spikes. Tightly-coupled with the bursts is the change in

the intracellular concentration of  $\text{Ca}^{2+}$  ( $[\text{Ca}^{2+}]_i$ ), oscillating phase-locked to the MP signal. The dynamics and network properties of the  $[\text{Ca}^{2+}]_i$  signal within a large number of simultaneously studied beta cells have been described recently [16,17]. Each oscillation originating in a single (or possibly a few) cell(s) spreads in a wave-like pattern throughout the islet [16]. The extent and exact nature of coupling the MP signal to the wave of the  $[\text{Ca}^{2+}]_i$  oscillation is presently unknown; the main reason being that when using the state-of-the-art electrophysiological approach, changes in MP can only be collected for a single or at most two cells from the networks of active beta cells [8]. In contrast, laser-scanning confocal  $[\text{Ca}^{2+}]_i$  imaging has been shown to provide data for many cells inside the beta cell network, with impeccable temporal and spatial resolutions [16]. To date, to the best of our knowledge, no reports have succeeded in using voltage-sensitive dyes to assess the electrical activities within networks of beta cells.

Recently, a novel family of voltage-sensitive dyes called VoltageFluor (VF) was introduced [9]. The dye incorporates within the outer leaflet of the cell membrane and its fluorescence changes by 2-3% for every 10 mV change in MP. The VF displays a several-fold increased voltage-sensitivity compared to the older families of voltage-sensitive dyes. We loaded islets of Langerhans in acute pancreatic tissue slices with the VF dye and stimulated them with 12 mM glucose, the most important nutrient secretagogue for beta cell activation. The VF signal during the glucose stimulation is comprised of rhythmical increases in fluorescence, most probably reflecting the characteristic oscillatory changes in MP recorded using a patch pipette. The MP signal spreads over the islets in a manner similar to the one observed for  $[\text{Ca}^{2+}]_i$  waves, with comparable durations of oscillations and velocities of wave-fronts. In order to resolve the degree of coupling between the MP and  $[\text{Ca}^{2+}]_i$  signals, double staining with voltage- and calcium-sensitive dyes is needed or, alternatively, a stimulus that would enable the detecting of differences between the two signals. Adding tetraethylammonium (TEA, 10 mM), a potent  $\text{K}^+$  channel inhibitor, to extracellular solution already containing a stimulatory concentration of glucose (12 mM), is expected to increase the size of the MP oscillations whilst shortening them at the same time [3]. Both effects can help unmask any potential temporal differences in the MP vs.  $[\text{Ca}^{2+}]_i$  kinetics masked by the longer oscillation durations during the glucose-only stimulation. Unfortunately, in addition to its benefits, TEA stimulation presents technical complications due to a faster spreading of the wave (1000  $\mu\text{m}/\text{s}$ ) and relatively short oscillation duration (100 ms). In order to achieve a sufficiently high temporal resolution, (i) a decrease in spatial resolution or (ii) a worsening of the signal-to-noise ratios of the single frames is needed (see example in Fig. 1a). Both results demand improved analysis of the MP wave spatiotemporal images.

Unfortunately, the obtained images are highly noisy and only experts can interpret them correctly. In order to cope with high noise and low image resolution, we employed an approach based on connected operators of mathematical morphology [14]. Namely, by acting on connected pixels with constant intensities (called flat-zones) rather than individual pixels, the connected operators uphold good contour preservation properties and are able to prevent significant image distortions when applied for filtering. Section 2 provides definitions of the connected operators used in this paper. The new approach for image filtering is proposed in Section 3. The results are shown in Section 4, whilst Section 5 concludes the paper.

## 2. THEORETICAL BACKGROUND

Let  $f: E \rightarrow \mathbb{N}$ , where  $E \subset \mathbb{N}^2$ , be an image,  $p \in E$  a pixel, and  $T_l = \{p \mid f[p] > l\}$  a threshold set of  $f$  obtained at level  $l$ . A peak connected-component  $C_l^p \in T_l$  is a connected set of pixels from  $T_l$  containing  $p$ , whilst the relationship  $l' \geq l \rightarrow T_{l'} \subseteq T_l$  explains the nesting of peak connected-components. An attribute opening  $\gamma_a^A(f)$  acting on  $f$  is an operator that removes all  $C_l^p$  with an attribute value  $A(C_l^p)$  smaller than a given threshold  $a$ . Note that increasing attributes are described by the relationship  $C_{l'}^p \subseteq C_l^p \rightarrow A(C_{l'}^p) \leq A(C_l^p)$  and further definitions are given only for them. Examples of such attributes are the area of  $C_l^p$ , its volume or its diagonal. An attribute closing, denoted as  $\phi_a^A$ , is a dual operator of  $\gamma_a^A$  and it is given by  $\gamma_a^A(f) = -\phi_a^A(-f)$  [10]. An image decomposition, named differential attribute profiles (DAP), that is based on granulometries of attributed openings was proposed by Ouzounis et al. [12]. A granulometry is an ordered set of morphological filters that progressively reduce the content of the image by filtering it on an increasing scale. A granulometry based on the attribute  $A$  with  $N+1$  members is defined by a set of attribute threshold values  $\mathbf{a} = \{a_i\}$ , where  $a_{i-1} < a_i$  and  $i \in [0, N]$ . The definition of DAP obtained from  $f$ , is given as

$$\Gamma_a^A(f) = \{\gamma_{a_{i-1}}^A(f) - \gamma_{a_i}^A(f)\}, \quad (1)$$

where  $i \in [1, N]$ . Thus, DAP is a top-hat scale-space obtained from  $f$ , where  $\Gamma_a^A(f)[p]$  is a vector containing  $N$  response values obtained at  $p$ . This vector is named the positive response vector whilst its dual (i.e. negative response vector) is obtained by an ordered set of attribute closings, and is denoted as  $\Phi_a^A(f)$ .

Recently, Mongus and Žalik [10] proposed extraction of the most-contrasted connected-components from  $\Gamma_a^A(f)$  by a mapping schema named  $\Theta$ -mapping. Consider a set of peak connected-components  $C^p = \{C_l^p\}$  containing  $p$ . The most-contrasted connected-component  $C_{max}^p \in C^p$  with respect to the given  $\Gamma_a^A(f)$  can be uniquely identified by registering the largest response contained in  $\Gamma_a^A(f)[p]$  and the associated attribute threshold value at which it was obtained.  $\Theta$ -mapping is, therefore, given by  $\Theta_a^A: f \rightarrow (R(f), Q(f), P(f))$ , where

- $R(f)$  is a transformation function registering maximal responses obtained at pixel  $p$  and is formally given by

$$R(f)[p] = \max \Gamma_a^A(f)[p]. \quad (2)$$

- $Q(f)$ , called associated function, maps the attribute value that lead to the maximal response. Formally it can be written formally as

$$Q(f)[p] = \max i \mid \Gamma_a^A(f)[p][i] = R(f)[p]. \quad (3)$$

Note that due to the increasing property of  $A$ , there exist at most one  $C_l^p$  where  $A(C_l^p) = Q(f)[p]$  and thus it defines  $C_{max}^p$ . However, it is possible that no response is

obtained at a given  $p$ , meaning that the corresponding peak connected-components are not in contrast against their surroundings in regard to the given  $\Gamma_a^A(f)$ . In this case,  $p$  is recognised as a background [10].

- $P(f)$  is an arbitrary parameter that can be measured for  $C_{max}^p$  and can be used for estimating multiple characteristics that lead to a particular response. Formally, it can be defined as

$$P(f)[p] = \text{parameter}(C_{max}^p). \quad (4)$$

## 3. REMOVAL OF NOISE FROM LOW RESOLUTION IMAGES OF BIOLOGICAL CELLS

This section proposes a new method for the improved monitoring of membrane-potential oscillations in a time-series data (i.e. a video). This method consists of the following three steps:

- Firstly, detection of changes is achieved by decomposing images from the time-series into a series of high-frequency differences and low-frequency alpha-trimmed mean images.
- $\Theta$ -mapping is then applied on each high-frequency image in order to detect and remove noise.
- Finally, output images are constructed by adding the filtered images of differences to the low-frequency images.

In the continuation, these three steps are explained in detail.

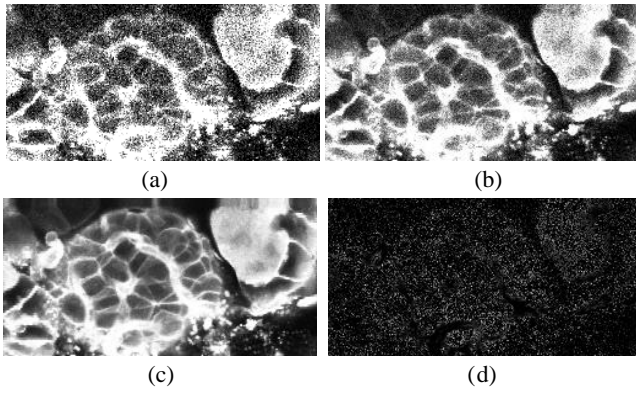
### 3.1 Image smoothing

The first step of the proposed method considers smoothing within the time-domain in order to obtain a representative image of the cells' structures and, consequentially, identify potential candidates for the noise pixels. Since no changes are expected in the positions of the cells over a short period of time, changes between the successive images can be related to the changes in the membrane potential and the presence of noise. Smoothing the images through the time-domain should, therefore, allow for efficient estimation of the image's structural part, whilst the textural part (i.e. the image of differences between the original and smoothed image) should contain most of the noise. It can, furthermore, be claimed that short, high-contrasted changes (e.g. high increases or decreases of intensity) are most-probably introduced by the impulse noise. This, consequentially, leads to applying a band-pass alpha-trimmed mean filter to each pixel within the time-domain. Let a time-series  $f = \{f_t\}$ , where  $t \in [0, T]$ , and a function  $\max^{(n)}\{f_t[p]\}$  estimate the  $n$ th largest value contained in the set of values  $\{f_t[p]\}$ . The alpha-trimmed mean filter  $\Lambda_\alpha^w(f_t)$  acting on  $f_t$  with a window size  $w$  is at a pixel  $p$  given by [13]

$$\Lambda_\alpha^w(f_t)[p] = \frac{1}{1+2(w-\alpha)} \sum_{n=\alpha}^{2w+1-\alpha} \max^{(n)}\{f_{t'}[p]\}, \quad (5)$$

where  $t' \in [t-w, t+w]$ , and  $\alpha$  defines the number of trimmed values. Each  $f_t$  is then decomposed into a high-frequency  $f_t^S$  (structural) and low-frequency  $f_t^T$  (textural) part, where  $f_t^S = \Lambda_{10}^{50}(f_t)$  and  $f_t^T = \Lambda_1^2(f_t - f_t^S)$ . Note that trimming values  $\alpha$  (and consequentially  $w$ ) were defined according to the oscillations in the average image intensity (see Figure 1). Namely,  $\alpha = 1$  was selected when constructing  $f_t^T$  since no significant oscillations can be noticed over such a short period of time. On the other hand,  $\alpha = 10$  was selected when constructing  $f_t^S$  since all the oscillations are up to 10 frames long. In this way, we ensure that the signal is entirely contained in  $f_t^T$ , that is further processed. The obtained  $f_t^S$  and  $f_t^T$  are shown in Fig. 1.





**Figure 1:** Input frame 55 (a) filtered with  $w=5$ ,  $\alpha = 1$  (b), filtered with  $w=50$ ,  $\alpha = 10$ , and the obtained  $f_t^S$  (c) and  $f_t^T$  (d).

### 3.2 Noise filtering

Although the alpha-trimmed averaging performed in the previous step sufficiently reduced the impulse noise from the original series of images, it does not remove the noise that is present for more than a single frame. Additional filtering is, therefore, performed within the space-domain of the textural image  $f_t^T$ . In this case, noise may be described as a set of small compact regions with high contrast against their surroundings. Thus,  $\Theta$ -mapping provides an efficient approach for its identification. For this purpose, the definition of the underlying DAP is considered first.

In order to avoid relying on any particular shape when estimating the most-contrasted connected-components, area attribute  $A$  was used and the same area threshold vector  $\mathbf{a}$  is defined for all  $f_t^T$ . Due to the low image resolution, sufficient results were obtained using relatively small area-zones defined as  $\mathbf{a} = \{3 * i\}$ , where  $i \in [0,10]$ . The shape-compactness parameter of the most-contrasted connected-component was then estimated by  $\Theta$ -mapping using shape-based filtering support. Thus, the value of  $P(f_t^T)$  at a particular pixel  $p$  is defined as [11]

$$P(f_t^T)[p] = \frac{A(C_{max}^p)}{9\pi * DT(C_{max}^p)}, \quad (6)$$

where  $A(C_{max}^p)$  is the area of  $C_{max}^p$  and  $DT(C_{max}^p)$  is the average value of the well-known distance transformation performed on  $C_{max}^p$ . A set of noise pixels  $p^*$  is then defined by

$$p^* = \{p | R(f_t^T)[p] > 10.0, P(f_t^T)[p] < 10.0\}, \quad (7)$$

where the threshold values were defined according to the amplitudes of the oscillations in the average intensity graph (see Figure 1). Finally, the same filtering principle was performed for the removal of the dark noise by applying  $\Theta$ -mapping on the negative response vector (obtained by an ordered set of area closings  $\Phi_a^A(f)$ ).

### 3.3 Image reconstruction

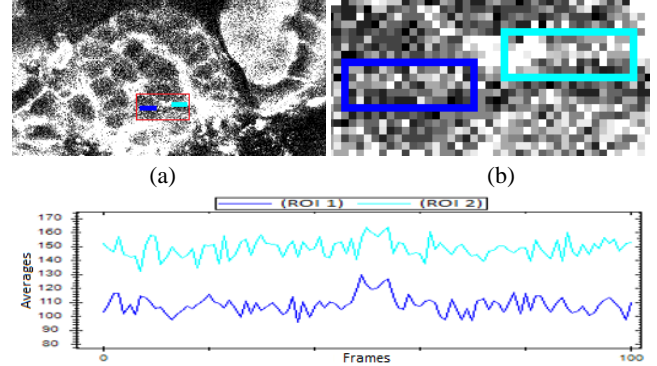
In the last step of the method, textural images  $f_t^T$  are denoised first and then added to the structural images  $f_t^S$  in order to obtain a filtered version of the original time-series  $f'$ . First, the filtered textural image  $f_t^{T'}$  is obtained by

$$f_t^{T'}[p] = \begin{cases} f_t^T[p]; & p \notin p^* \\ f_t^T[p] - R(f_t^T)[p]; & p \in p^* \end{cases} \quad (8)$$

A convolution with a  $3 \times 3$  Gaussian kernel is then performed on  $f_t^{T'}$  in order to smooth out any possible irregularities introduced by eq. 10 and the output can finally be given as  $f'_t = f_t^S + f_t^{T'}$ .

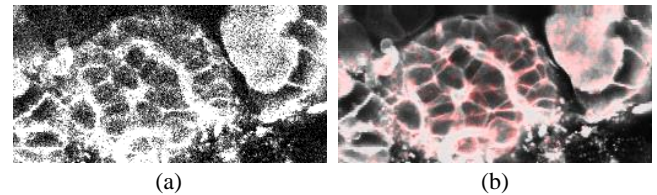
## 4. RESULTS

The proposed method is compared with the existing approach for the detection of changes in membrane potential, where a region of interest (ROI) is selected in order to monitor changes in the average intensity (see Fig. 2). Although averaging improves the resistance of the method against noise, it does not remove it (i.e. in Fig. 2c only the middle peak represents actual signal). However, this approach does not allow monitoring a single pixel.

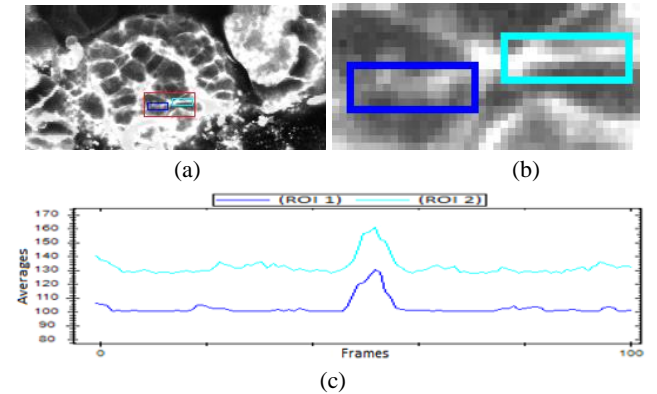


**Figure 2:** Input frame 55 with ROIs (a) magnification of region containing ROIs (b) and corresponding chart of average values within selected ROIs (c).

In figure 2a, frame 55 from the input video is shown with two selected ROIs (magnification in Fig. 2b). The corresponding time series of average intensities are shown in Fig 2c. A significant change in the membrane potential can be recognised between frames 45 and 55, as it is similar in both regions, whilst significant differences in oscillations can be observed outside of this range. These are characteristics should, therefore, be preserved by the proposed method, whilst visualisation of the signal propagation should be enhanced.



**Figure 3:** Input frames 55 (a), enhanced after noise filtering (b). The results of the method are shown in Fig. 3, where frame 55 is used for the comparison. Significant denoising can be observed, whilst visualising the signal in the red channel emphasises the representations of changes for improved analysis.



**Figure 4:** Output frame 55 (a) with ROIs (b) and corresponding chart of average values within selected ROIs (c).

Finally, the method has been evaluated in terms of monitoring the average intensity in ROIs. An important improvement achieved by the proposed method can be observed in Fig. 4. Fig. 4a shows output frame 55 where significant de-noise has been achieved and the magnifications of those area containing selected ROIs are shown in Fig. 4b. By comparing the obtained time-series charts (Fig. 4c) with the original (Fig. 2c), a significant noise reduction can also be observed in this case, whilst even the actual impulse in the membrane potential has been emphasised. Consequentially, further analysis can significantly be improved.

## 5. CONCLUSION

This paper proposed an improved method for monitoring changes in the membrane potentials of biological cells. This method is able to remove the impulse noise by alpha-trimmed mean filtering applied on a series of images within the time-domain. The derived band-pass filter, moreover, allows for accurate extraction of the actual signal, while mapping has proved to be an efficient tool for the detection of noise within the space-domain. By using this extraction of the signal in a separate band, improved visualisation can be achieved, where the signal can be enhanced and mapped into different coloured channels (e.g. red). Future work will consider an analysis of the time-series charts of average intensities for an automatic definition of the used threshold values.

## 6. ACKNOWLEDGEMENTS

This work was supported by the Slovenian Research Agency under grants P2 – 0041 and through Centre of Excellence for Integrated Approaches in Chemistry and Biology of Proteins. Present paper was produced within the framework of the operation entitled “Centre for Open Innovation and Research of the University of Maribor”. The operation is cofounded by the European Regional Development Fund and conducted within the framework of the Operational Programme for Strengthening Regional Development Potentials for the period 2007 – 2013, development priority 1: “Competitiveness of companies and research excellence”, priority axis 1.1: “Encouraging competitive potential of enterprises and research excellence”.

## 7. REFERENCES

[1] Ashcroft FM, Harrison DE, Ashcroft SJ (1984) Glucose induces closure of single potassium channels in isolated rat pancreatic beta-cells. *Nature* 312: 446-448.

[2] Association AD (2012) Diagnosis and Classification of Diabetes Mellitus. *Diabetes Care* 35: S64-S71.

[3] Atwater I, Ribalet B, Rojas E (1979) Mouse pancreatic beta-cells: tetraethylammonium blockage of the potassium permeability increase induced by depolarization. *The Journal of Physiology* 288: 561-574.

[4] Dean PM, Matthews EK (1968) Electrical Activity in Pancreatic Islet Cells. *Nature* 219: 389-390.

[5] Henderson JR (1969) WHY ARE THE ISLETS OF LANGERHANS? *The Lancet* 294: 469-470.

[6] Hellman B (1959) Actual distribution of the number and volume of the islets of Langerhans in different size classes in non-diabetic humans of varying ages. *Nature* 184(19): 1498-1499.

[7] Kim A, Miller K, Jo J, Kilimnik G, Wojcik P, et al. (2009) Islet architecture: A comparative study. *Islets* 1: 129-136.

[8] Meissner HP (1976) Electrophysiological evidence for coupling between [beta] cells of pancreatic islets. *Nature* 262: 502-504.

[9] Miller EW, Lin JY, Frady EP, Steinbach PA, Kristan WB, Jr., et al. (2012) Optically monitoring voltage in neurons by photo-induced electron transfer through molecular wires. *Proceedings of the National Academy of Sciences of the United States of America* 109: 2114-2119.

[10] Mongus D and Žalik B (2013) Computationally efficient method for the generation of a digital terrain model from airborne LiDAR data using connected operators *IEEE J of Selected Topics in Applied Earth Observations and Remote Sensing* in press.

[11] Montero RS and Bribiesca E (2009) State of the art of compactness and circularity measures. In *International Mathematical Forum* 4(25-28): 1305-1335.

[12] Ouzounis GK, Pesaresi M, and Soille P (2012) Differential area profiles: Decomposition properties and efficient computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(8): 1533–1548.

[13] Oten R and Figueiredo RJ (2004) Adaptive alpha-trimmed mean filters under deviations from assumed noise model. *IEEE Transactions on Image Processing* 13(5): 627-639.

[14] Salembier P and Wilkinson MHF (2009) Connected operators: A review of region-based morphological image processing techniques. *IEEE Signal Processing Magazine* 136(6): 136–157.

[15] Serre-Beinier V, Le Gurun S, Belluardo N, Trovato-Salinario A, Charollais A, et al. (2000) Cx36 preferentially connects beta-cells within pancreatic islets. *Diabetes* 49: 727-734.

[16] Stožer A, Dolenšek J, Rupnik MS (2013) Glucose-Stimulated Calcium Dynamics in Islets of Langerhans in Acute Mouse Pancreas Tissue Slices. *PLoS ONE* 8: e54638.

[17] Stožer A, Gosak M, Dolenšek J, Perc M, Marhl M, et al. (2013) Functional Connectivity in Islets of Langerhans from Mouse Pancreas Tissue Slices. *PLoS Comput Biol* 9: e1002923.

[18] Taylor SI (1999) Deconstructing Type 2 Diabetes. *Cell* 97: 9-12.

[19] Unwin N, Whiting D, Gan D, Jacqmain O, Ghyoot G (2012) *IDF Diabetes Atlas*. Brussels: International Diabetes Federation, p. 143.

## About the author

**Denis Špelič** is an assistant researcher at Faculty of Electrical Engineering and Computer Science at University of Maribor.

**Jurij Dolenšek** is an assistant researcher at Faculty of Medicine at University of Maribor.

**Andraž Stožer** is an assistant researcher at Faculty of Medicine at University of Maribor.

**Marjan Slak Rupnik** is a professor at Faculty of Medicine at University of Maribor.

**Borut Žalik** is a professor at Faculty of Electrical Engineering and Computer Science at University of Maribor.

**Domen Mongus** is an assistant professor at Faculty of Electrical Engineering and Computer Science at University of Maribor.

# Human Body Shape Modeling Using 3D Range Data for Healthcare Applications and Beauty Services

Igor Goncharenko  
3D Incorporated  
Yokohama, Japan  
goga\_3d@yahoo.com

## Abstract

The paper describes a system to model variations of body shapes by individual 3D scan data and to provide the modeling results to the users via the Internet. The clients of the system are involved into unloading dieting programs and exercises in sport clubs. After scanning and data processing, they can control modeling of their prognostic body shapes in case of weight gain/loss. The client interface allows the users to browse the processed data in a 3D GUI, to simulate body shapes with new desired body measurements (such as weight, waist, chest circumferences, etc.), and to generate printed reports. Modeling can be controlled by a few body dimensions, or even by one measurement.

**Keywords:** 3D Body Scanners, Digital Human Modeling.

## 1. INTRODUCTION

Recent advances in the development of high-resolution and accurate 3D scanners for whole-body shape capturing make it attractive to utilize the scan data in many application fields, e.g., in individual design of car interiors, clothing, anthropometry, healthcare, social studies, realistic 3D media content creation, aerospace, defense, and national surveys. Formerly *body dimensions* were widely used for apparel design, ergonomic assessment of working/operational spaces, and reverse engineering. The data are, for instance, weight, height, leg length, waist, hip or chest circumferences, etc. Body dimensions are defined as the distances between some specific points on human body surface (*landmarks*), or circumferences measured at the level of the landmarks. The landmarks are anatomically defined features, for example, top of the head, tip of the spinous process of the 7<sup>th</sup> cervical vertebra, the most lateral point of acromion, etc [8]. With modern 3D scanning technologies, it becomes possible to automatically locate the landmarks and to retrieve body dimensions from accurately scanned range data. And, the main advantage of 3D scanning is that it provides a highly detailed individual's body *surface* shape, which can be used in further human-oriented modeling.

Human body shape modeling (HBSM) utilizing 3D scan data is a rapidly evolving multidisciplinary area [7], applicable also to biomedical applications [11]. The beauty services and healthcare applications include: anthropometric surveys; body deformity, asymmetry and obesity control; rehabilitative and training monitoring; weight control ([6],[11]). Now, even some visual cognition factors can quantitatively be defined with the aid of HBSM, for instance, woman body attractiveness [10], or adolescent body perception [1]. Various methods and tools are used in HBSM depending on application fields and input 3D scan data types, however, controls of HBSM are still traditional body dimensions mentioned above, because ergonomics designers (or,

HBSM end users) clearly understand the nature of the measurements and can easily manipulate with them during modeling. It is also important to provide the end users of HBSM systems the ability to control the modeling process by a limited number of variables to avoid ambiguity and complexity of control, especially, if modeling is done though the Internet in interactive way. Moreover, in some cases a complete set of body dimensions is not provided with 3D scan data.

Recently, Internet-based applications using 3D scan data appeared mostly for apparel design. In [4] several virtual dressing systems are described (MyShape, Virtual try-on, Lands' End, My Virtual Model). These systems use "static", or pre-defined 3D models from databases, without on-line body shape modeling. Israeli company OptiTex presented a Web-based garment fitting system with 3D preview of a synthetic avatar created by user's body dimensions retrieved from individual 3D scan data. It would be worthwhile to implement Web-based systems allowing the end-users to manipulate with their *personal* 3D scan data, to model, for instance, the body shape after weight loss/gain, after physical or rehabilitative training, and to display the virtually corrected postures of the models. The users of such systems would be motivated not only in taking care on their health and physical exercises, but also in participating in 3D scanning process itself and contributing into 3D scan databases.

The purpose of this work is to develop a Web-based system to realistically model variations of prognostic body shapes by individual 3D scan measurements and to provide the results of modeling to the users via the Internet through a 3D graphical interface. The paper is organized as follows. First, in Section 2, input data processing methods and human body shape modeling techniques, which are used in the system, are described. In Section 3, we present some statistical results allowing us to reduce the number of body dimensions to control body shape modeling. Section 4 describes the Web system's implementation details such as server-side functionality, client interface with 3D graphics, etc. Finally, conclusions are summarized in Section 5.

## 2. MODELING METHOD

### 2.1 Data

Nowadays, several active and passive optical 3D whole-body scanning technologies are used, for example, optical laser scanning, photogrammetry based on structured light, modulated light, passive stereoscopy, etc. The success of CAESAR Project (see, e.g., [2]) focused attention of professionals on the industrial-type equipment providing harmless scanning with high quality output data (with about million of 3D points). Such 3D scanning systems are produced, for instance, by Cyberware, Hamamatsu Photonics, Human Solutions/Vitronic, [TC]<sup>2</sup>, and other manufacturers. The trend of the 3D body scanners' development

is characterized by decreasing the cost of hardware and time of scanning (to few of seconds), easiness of installation and operation, truly 360 degree scanning with low noise and data missing, nearly millimeter range accuracy. It results in satisfactory quality of the produced data with high level of details of face and body parts of scanned subjects, ability to automatically extract dozens of body dimensions and landmarks. We mostly used data collected with the Bodyline Scanner from Hamamatsu Photonics [5].

The standard manufacturer's preprocessing software, Bodyline Manager, allows 54/55 (male/female) auto-landmarks, and about one hundred body dimensions for one data range scan. Original data are recorded as a polygonal model in Wavefront graphical format (OBJ) with several hundred thousand vertices, accompanied by personal information (age, weight, height, BMI, gender, etc). Even Bodyline Manager has an interactive functionality to correct landmark position by a 3D editor, our processing software (see Section 2.2) was developed with the assumption that location of some landmarks may not be correct. We also considered the case, when landmarks and body measurements are not given at all. The last case was tested with the range data collected by other body scanner [3].

Several thousand subjects participated in 3D body scanning campaigns managed by Nihon Unisys. The data were collected with the system "3D-Navi" from Japanese males and females aged from 15 to 96 years old. On the server side of the system, 3D scan data and measurements were sorted into ten gender-age groups: younger than 30 years old, from 30 to 40, between 40 and 50, from 50 to 60, and older than 60 years old for both, male and female subjects. This grouping reduces intra-population variations, because it is known, for instance, that on the average the younger generation is taller, older generation tends to lose muscularity, etc. For operational use, statistical analyses were conducted separately for the sex/age groups, and overall inter-population statistics were investigated experimentally.

## 2.2 3D model creation

The data collected with 3D scanners are characterized by significant variations of body shapes, postures, number of vertices, presence of holes and noise in the data. Nowadays, the most proven approach for HBSM is based on template models and reconstruction of body surfaces from eigen-spaces. Pioneered in [2], the method uses Principal Component Analysis (PCA), which captures the body shape variations from an "average", or template shape. In the method (utilizing landmarks) all the vertices of a template model are non-rigidly morphed until the best fitting to the point cloud of the input 3D scan data. The resulting fitted model is called *homologous*. Therefore, there is exact correspondence of vertices for any pair of homologous models, or a homologous model and template. This property makes it possible to apply PCA to model shape variations. The approach also resolves hole (missed data) interpolation, and it has variants for landmark-less fitting. If the template is accompanied by a skeleton model with body mesh skinning weights, the "bone-skin" technique (widely used in computer graphics animation) can be applied for fitting if the postures of template and scan data are significantly different. We implemented an HBSM method based on male (13379 vertices) and female (16780 vertices) "Dhaiba" templates [9] in the form of closed polyhedra instrumentally connected with the link-joint skeleton models and landmark sets (Figure 1).

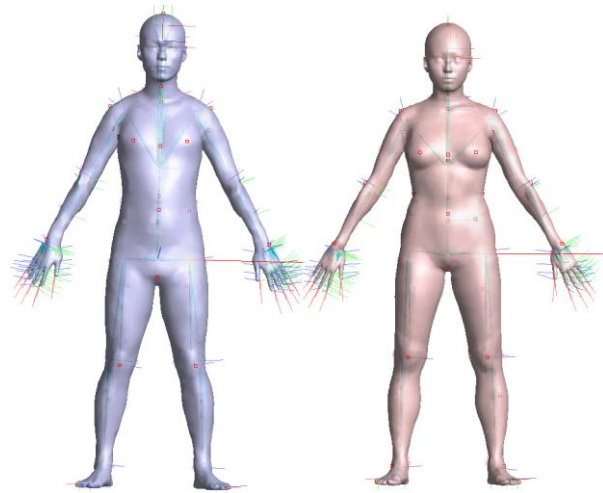


Figure 1: "Dhaiba" template models.

The process of model creation is as follows. First, the homologous model is created iteratively for the posture of original 3D scan data, and the joint centers and link lengths are estimated during the iterations. Finally, the subject-specific skeleton is completely defined. To avoid posture variations during modeling, the homologous model is created for the standard template posture, and the main body part is extracted. The overall process is depicted in Figure 2.

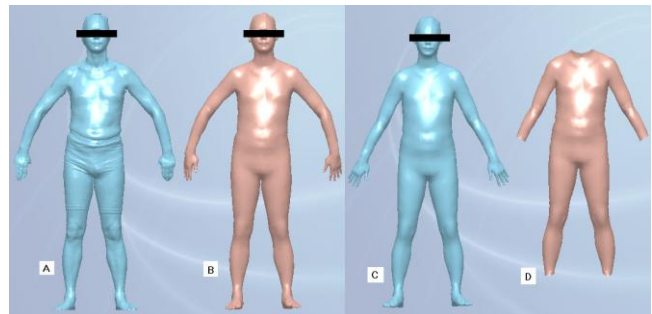


Figure 2: Original scan data (A) and homologous models in original (B) and standard (C) postures.

## 2.3 Body shape modeling

For Web-based modeling providing the customer his own prognostic body shape view in 3D, the following scenarios were considered: visualize the user prognostic body shape in case of weight loss/gain; visualize the shape in case of increase/decrease of the waist circumference; visualize the body model in "correct" (standard) posture without body asymmetry and deformations (which can appear, for instance, due to early stage of scoliosis); estimate the required weight loss for the desired waist decrease.

In all the above scenarios, it is quite natural to vary weight and/or body dimensions (except height and other measurements, whose variations lead to significant bone deformations). We suppose that these variations mostly result in changes of the main body part shape (Figure 3D). It was also supposed that head, face, hands, and feet variations are small in case of weight loss. The main body part surface is composed from a smaller number of vertices

( $N=2618$  in the case of our male homologous model), and statistical calculations can be done much faster for it.

Similarly to [2], for each statistical group (collection of  $K$  main body homologous models, where each body shape is composed from  $N$  vertices), we describe the body shape of the  $k$ -th individual ( $k=1, \dots, K$ ) by  $3N$ -dimensional vector

$$a_k = (x_{1k}, y_{1k}, z_{1k}, x_{2k}, y_{2k}, z_{2k}, \dots, x_{Nk}, y_{Nk}, z_{Nk})^T.$$

Calculating average shape  $a_0$  and applying PCA, we can find the eigenvector decomposition (basis of vectors  $e_i, i=1, \dots, K-1$ ) for variation of any shape  $a_k$  from the collection:

$$a_0 = \frac{1}{K} \sum_{k=1}^K a_k, \quad a_k - a_0 = \sum_{i=1}^{K-1} PC_{ki} e_i, \quad PC_{ki} = \frac{(a_k, e_i)}{(e_i, e_i)}.$$

Each principal component (PC) cannot clearly be associated with a single anthropometric body measurement such as height, weight, or waist circumference, it is rather influenced by combination of the measurements and variations of postures. Each body shape  $a_k$  has  $L$  supplementary measurements  $\{d_{k1}, d_{k2}, \dots, d_{kL}\}$ , derived from the range data and measured with scale. Supposing that scores of the  $i$ -th PC ( $i=1, \dots, K-1$ ) depend on  $L$  ( $L < K$ ) multiple measurements  $\{d_1, d_2, \dots, d_L\}$ , we may propose control of body shape modeling through them for the case of simple linear dependency:

$$PC_i(d_1, d_2, \dots, d_L) = b_{i0} + \sum_{j=1}^L b_{ij} d_j$$

Taken into account that PC scores can directly be reconstructed from the homologous model, we can compare them with ones obtained from the above linear model to clarify how accurate the assumption of linearity is. Figure 4 shows the comparison for  $PC_1$  for one of the scanned inter-population female group for 533 subjects aged from 16 to 96 years old. (In the Figure, gray squares depict correlation without skeleton measurements, and black triangles show correlation with skeleton measurements.) On the average, for this population the results of linear modeling are very good for lower PCs. Particularly, correlation is 0.996 for  $PC_1$  if skeleton measurements are taken into account (Figure 4).

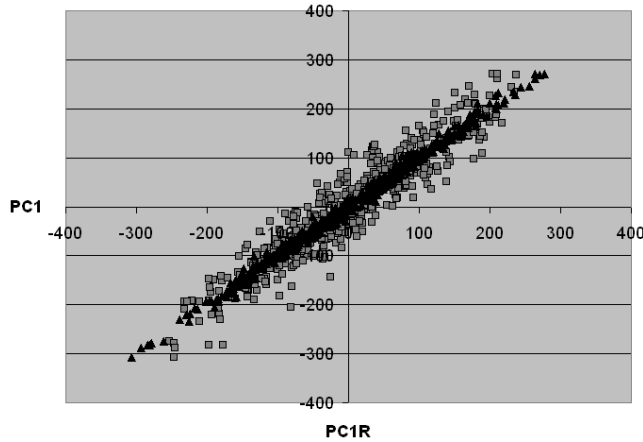


Figure 3: Correlation of measured and modeled  $PC_1$ .

For the case of individual body simulation with available homologous model  $a_k$  and measurement vector  $d_k = (d_{k1}, d_{k2}, \dots, d_{kL})^T$ , it is preferable to use increments of the measurements. Then, denoting "new" measurement set as  $d = d_k + \delta d$ , where  $\delta d = (\delta d_1,$

$\delta d_2, \dots, \delta d_L)^T$  is the measurement increment, we can use our regression results to simulate new shape in terms of the increments:

$$a_{sim}(d) = a_k + \sum_{i=1}^{K-1} (PC_i(d) - PC_i(d_k)) e_i.$$

For high order eigenvalues with low significance, variation magnitudes are significantly decreased. Limiting the value  $K$ , the last formulae can be used for fast modeling in real-time on the client sites of Web-based systems.

## 2.4 Reduction of the number of controls

As the developed body shape modeling method was supposed to be implemented for interactive control through the Web, it is reasonable to make such control more user-friendly and simple. If the number  $L$  of controlling measurements is large enough (e.g., basic software for our 3D scanner can provide up to 72 body dimensions) and the user is allowed to vary them independently, the resulting simulated body shapes would be very unrealistic and the control process itself would be confusing and difficult for the users. Therefore, more simplified and robust method of modeling was proposed. Once the developed modeling system is mostly intended for personal body weight control, it was decided to select 16 measurements: weight, height, BMI, and 3D scanner's measurements obviously dependent from weight gain/loss.. The body dimensions are mostly the circumferences of body parts (neck, torso, arms, and legs) measured in horizontal cross sections.

Additional 28 skeleton measurements (bone lengths and distances between joints) calculated during building the homologous model were also used. The skeleton measurements, when being kept constant in individual shape modeling, result in constraining effect and prevent bone deformations, and as a consequence, the modeled shapes look much more realistic, and the regression errors (see, e.g., Figure 4) become smaller. The final composition of body shapes from PCs calculated with the basic measurement set and the complete set appeared very similar for the task of weight variations. Being included in the model, skeleton data are not changed in modeling.

After analysis of measurements it was found that correlations of symmetric pairs of arm and leg circumferences are high. Also, BMI measurement is redundant, because it can explicitly be calculated through weight and height. Height should not be varied by the users, when they model their body shapes. Based on the above speculations, a reduced set of control measurements was found.

However, change in one measurement requires adjustment of all others simultaneously for realistic body shape view. It was found that horizontal body circumferences are highly correlated with weight. Based on this fact, we can approximately model the changes of the measurements by weight change in accordance with the following scheme. Suppose, all the measurements depend on weight  $w$  linearly:

$$d_i(w) = d_i + c_i(w - d_2), \quad i = 1..16,$$

where  $d_2$  is weight measured on the day of 3D body scanning together with all other measurements  $d_i$ . Thus, it is possible to predict the change of waist circumference by the change of weight, and vice versa. Coefficients  $c_i$  are defined by linear regression through all the data for each statistical group. The standard deviations estimated from the regression are small enough to reconstruct all the measurements even at high gain or loss of weight.

### 3. WEB-BASED IMPLEMENTATION

The methods described in the above Section were implemented in our Web-based system “3D-Navi” managed by Nihon Unisys Ltd. Also, the methods were realized in a standalone application, but there are many disadvantages of this approach, because the collected 3D scan data are commercially sensitive and should not arbitrarily be accessed or updated by the users. Also, there are serious ethical reasons (e.g., the requirement of non-disclosure of almost undressed personal body shapes) restricting a stand-alone implementation. Therefore, it was decided to implement centralized control of data access from Web clients with a secure authorization mechanism.

In order to motivate people in participating in 3D scanning, the main scenario of the usage of the “3D-Navi” Web-based system was supposed to provide the subjects with ability to observe their current personal body shapes, model body shapes in cases of weight loss or gain, visualize their virtually regal carriage, and, therefore, to stimulate them in physical training, exercises, and dieting. To realize the above scenario, “3D-Navi” was designed to provide user-friendly and secure access to personal data, to quickly react on the users’ requests from the client side, to intuitively convenient 3D browse the whole body shape and selected body parts, to generate reports with 3D views of current and modeled body shapes and body dimensions in textual form. Real-time rendering of 3D contents on the client side (not only homologous models, but also original 3D scans with up to one million vertices) implies high requirements on the client system performance. Also, on-line modeling requires the efficiency of server-side calculations and network transfer.

In case of gradual increase of collected 3D scans in the database, the data required for one body shape modeling, namely, eigenvectors and PCs, can be much more bulky than the size of file with modeled 3D body shape. Thus, it was decided to implement modeling as an optimized server-side application without cutting of higher order eigenvectors. PCA analysis is periodically updated on the server by administrator’s request. Modeled body shapes are transferred to the client side in a compressed form of the Wavefront graphical format.

To support the variety of browsers and platforms, the client side sub-system was implemented as an HTML form with JavaScript control calling Java applet for 3D visualization in a panel inside the main frame of the client window. 3D graphics and navigation in 3D is implemented in Java3D on top of OpenGL, which provide real-time rendering of modeled shapes. The applet also provides a variety of service functions, such as model loading, high quality off-screen image rendering for modeling session report generation, modeling measurement settings, etc. As described in the above Section, modeling can be done in several modes: with measurement masking, independent, and setting by one measurement.

In the 3D window the user can visualize up to two models (for example, original homologous and prognostic with over-weight of 20kg is shown in Figure 4), rotate, scale, align them, or start auto-rotation mode. Different body parts can be colored with specially assigned colors for displaying the body part surface difference. The important feature of the system is overlapping the shapes in a semi-transparent mode, which displays the difference of the overall shapes or body parts.

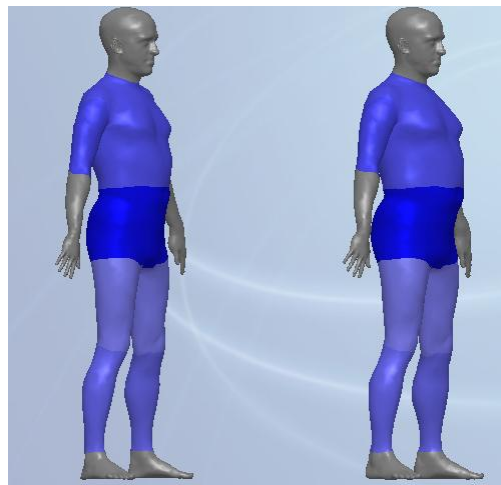


Figure 4: 3D view of client GUI.

### 4. CONCLUSION

Web-based “3D-Navi” system’s design and implementation are described in this paper. The system is intended for remote access to individual 3D scan data and personalized body shape modeling by interactive changes of limited number of body dimensions (or, even one such measurement). Typical scenarios are, for instance, visualization of the desired user’s body shape after weight loss, or prognostic shape after wished waist decrease. The system is supposed to motivate the people in physical exercises and follow to unloading diets. The system is supported by real-time graphical rendering and browsing of the body shapes and/or body parts in a 3D window on the Web. Simple, intuitive and user-friendly 3D navigation allows easy comparison of current and prognostic body (or, body part) shapes. Also, the system allows to generate printed reports of the results of scanning, measurements, and body shape modeling.

For the Web-based system, special rapid methods of 3D range data preprocessing, canonical model creation with corrected posture, and statistical analysis were implemented. The methods are based on templates (body meshes with associated skeleton) with high level of details for face, hands, and feet. The individual features of face are well transferred to the modeled shapes. To simplify modeling by user control through a limited set of anthropometric data, an original method of measurement settings was proposed and implemented in the system.

The system was used commercially in Japan for a beauty service company. In future works, it would be also worthwhile to collect statistical data separately for persons actively participating in body building in sports clubs, because eigen-spaces for muscular people can be different for subjects with overweight.

### 5. REFERENCES

- [1] Aleong, R. and Duchesne, S. , ‘Assessment of adolescent body perception: Development and characterization of a novel tool for morphing images of adolescent bodies’, *Behavior Research Methods*, Vol. 39 No.3, 2007, pp.651-668.
- [2] Allen, B., Curless, B. and Popovic, Z., ‘The space of human body shapes: reconstruction and parameterization from range

- scans', *ACM Transactions on Graphics (ACM SIGGRAPH 2003)*, Vol.22 No.3, 2003, pp.587-594.
- [3] Cartesia: *Body Scanner* [online]. Available at: <http://www.space-vision.jp/E-HOME.html> (Accessed 15 June 2013).
- [4] Duffy, V. (Ed.), *Digital Human Modeling: Proc. ICDHM 2009*. Lecture Notes in Computer Science, Vol. 5620, Springer Publ., 2009, ISBN 978-3-642-02808-3, 767p.
- [5] Horiguchi, C. , 'BL (Body Line) scanner: the development of a new 3D measurement and reconstruction system', *International Archives of Photogrammetry and Remote Sensing*, Vol.32, B5, 1998, pp.421-429.
- [6] Kouchi, M. and Mochimaru, M. , 'Simulation of the body shape after weight change for health-care services' in *AHFE 2010: Proceedings of the 3rd International Conference on Applied Human Factors and Ergonomics*. Miami, Florida, USA, 2010, pp. 40-44.
- [7] Lerch, T., MacGillivray, M. and Domina, T. , '3D laser scanning: a model of multidisciplinary research', *Journal of Textile and Apparel, Technology and Management*, Vol. 5 No. 4, 2007, pp. 1-8.
- [8] Martin, R. and Knussmann, R. *Anthropologie, Band I*, Gustav Fischer Verlag Publ., Stuttgart, Germany, 1988, ISBN 3-437-30505-0.
- [9] Mochimaru, M. *Dhaiba: functional human models to represent variation of shape, motion and sub subjective assessment*, SAE Technical Paper, 2006-01-2345. 2006
- [10] Smith, K., et al., 'An analysis of body shape attractiveness based on image statistics: Evidence for a dissociation between expressions of preference and shape discrimination', *Visual Cognition*, Vol. 15 No.8, 2007, pp 927-953.
- [11] Treleaven, P. and Wells, J., '3D body scanning and healthcare applications', *Computer*, Vol. 40 No.7, 2007, pp. 28-34.

## About the author

Igor Goncharenko is a Chief Researcher at 3D Incorporated, Yokohama, Japan.

# Heterogeneous System Architecture (HSA): Software Ecosystem for CPU/GPU/DSP and other accelerators

Timour Paltashev  
 Graphics IP Engineering Division  
 Advanced Micro Devices, Sunnyvale, California, U.S.A.  
 {timour.paltashev}@amd.com

## Abstract

This STAR report describes the essentials of Heterogeneous System Architecture (HSA) with introduction and motivation for HSA, architecture definition and configuration examples. HSA performance advantages are illustrated on few sample workloads. Kaveri APU - first AMD HSA-based product is briefly described.

**Keywords:** GPU, CPU, DSP, APU, heterogeneous architecture.

## 1. INTRODUCTION

HSA is a new hardware architecture that integrates heterogeneous processing elements into a coherent processing environment. Coherent processing as a technique ensures that multiple processors see a consistent view of memory, even when values in memory may be updated independently by any of those processors. Memory coherency has been taken for granted in homogeneous multiprocessor and multi-core systems for decades, but allowing heterogeneous processors (CPUs, GPUs and DSPs) to maintain coherency in a shared memory environment is a revolutionary concept. Ensuring this coherency poses difficult architectural and implementation challenges, but delivers huge payoffs in terms of software development, performance and power. The ability for CPUs, DSPs and GPUs to work on data in coherent shared memory eliminates copy operations and saves both time and energy. The programs running on a CPU can hand work off to a GPU or DSP as easily as to other programs on the same CPU; they just provide pointers to the data in the memory shared by all three processors and update a few queues. Without HSA, CPU-resident programs must bundle up data to be processed and make input-output (I/O) requests to transfer that data via device drivers that coordinate with the GPU or DSP hardware. HSA allows developers to write software without paying much attention to the processor hardware available on the target system configuration with or without GPU, DSP, video hardware and other types of specialized compute accelerators.

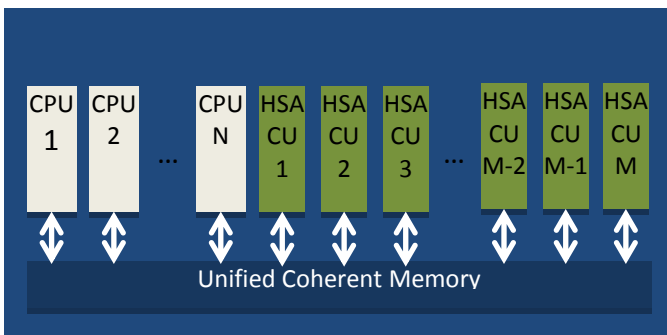


Figure 1: Generic HSA Accelerated Processing Unit (APU)

Fig.1 depicts generic HSA APU with multiple CPU cores and accelerated compute units (CU) which may include any type.

## 2. HSA ESSENTIAL FEATURES FOR USERS

Essential HSA features include:

- Full programming language support
- User Mode Queueing
- Heterogeneous Unified Memory Access (hUMA)
- Pageable memory
- Bidirectional coherency
- Compute context switch and preemption

**Shared page table support.** To simplify OS and user software, HSA allows a single set of page table entries to be shared between CPUs and CUs. This allows units of both types to access memory through the same virtual address. The system is further simplified in that the operating system only needs to manage one set of page tables. This enables Shared Virtual Memory (SVM) semantics between CPU and CU.

**Page faulting.** Operating systems allow user processes to access more memory than is physically addressable by paging memory to and from disk. Early CU hardware only allowed access to pinned memory, meaning that the driver invoked an OS call to prevent the memory from being paged out. In addition, the OS and driver had to create and manage a separate virtual address space for the CU to use. HSA removes the burdens of pinned memory and separate virtual address management, by allowing compute units to page fault and to use the same large address space as the CPU.

**User-level command queuing.** Time spent waiting for OS kernel services was often a major performance bottleneck in prior throughput computing systems. HSA drastically reduces the time to dispatch work to the CU by enabling a dispatch queue per application and by allowing user mode process to dispatch directly into those queues, requiring no OS kernel transitions or services. This makes the full performance of the platform available to the programmer, minimizing software driver overheads.

**Hardware scheduling.** HSA provides a mechanism whereby the CU engine hardware can switch between application dispatch queues automatically, without requiring OS intervention on each switch. The OS scheduler is able to define every aspect of the switching sequence and still maintains control. Hardware scheduling is faster and consumes less power.

**Coherent memory regions.** In traditional GPU devices, even when the CPU and GPU are using the same system memory region, the GPU uses a separate address space from the CPU, and the graphics driver must flush and invalidate GPU caches at required intervals in order for the CPU and GPU to share results. HSA embraces a fully coherent shared memory model, with unified addressing. This provides programmers with the same coherent memory model that they enjoy on SMP CPU systems.



This enables developers to write applications that closely couple CPU and CU codes in popular design patterns like producer-consumer. The coherent memory heap is the default heap on HSA and is always present. Implementations may also provide a non-coherent heap for advance programmers to request when they know there is no sharing between processor types.

The HSA platform is designed to support high-level parallel programming languages and models, including C++ AMP, C++, C#, OpenCL, OpenMP, Java and Python, as well as few others. HSA-aware tools generate program binaries that can execute on HSA-enabled systems supporting multiple instruction sets (typically, one for the LCU and one for the TCU) and also can run on existing architectures without HSA support.

Program binaries that can run on both CPUs and CUs contain CPU ISA (Instruction Set Architecture) for CPU unit and HSA Intermediate Language (HSAIL) for the CU. A *finalizer* converts HSAIL to CU ISA. The finalizer is typically lightweight and may be run at install time, compile time, or program execution time, depending on choices made by the platform implementation.

HSA architecture example platform is depicted on Figure 2.

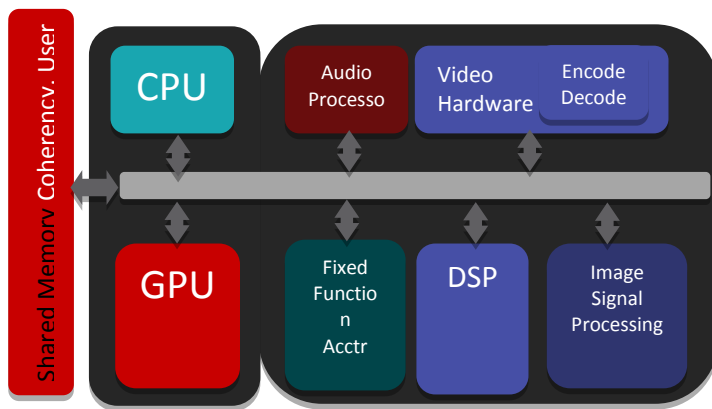


Figure 2: HSA architecture example platform.

### 3. HSA IMPLEMENTATION AND CONCEPTS

**Unified Programming Model.** General computing on GPUs has progressed in recent years from graphics shader-based programming to more modern APIs like DirectCompute and OpenCL™. While this progression is definitely a step forward, the programmer still must explicitly copy data across address spaces, effectively treating the GPU as a remote processor.

Task programming APIs like Microsoft’s ConCRT, Intel’s Thread Building Blocks, and Apple’s Grand Central Dispatch are recent innovations in parallel programming. They provide an easy to use task-based programming interface, but only on the CPU. Similarly, Thrust from NVIDIA provides a similar solution on the GPU.

HSA moves the programming bar further, enabling solutions for task parallel and data parallel workloads as well as for sequential workloads. Programs are implemented in a single programming environment and executed on systems containing both CPUs and CUs.

HSA provides a programming interface containing queue and notification functions. This interface allows devices to access load-balancing and device-scaling facilities provided by the higher-level task queuing library. The overall goal is to allow

developers to leverage both CPU and CU devices by writing in task-parallel languages, like the ones they use today for multicore CPU systems. HSA’s goal is to enable existing task and data-parallel languages and APIs and enable their natural evolution without requiring the programmer to learn a new HSA-specific programming language. The programmer is not tied to a single language, but rather has available a world of possibilities that can be leveraged from the ecosystem.

**Queuing.** HSA devices communicate with one another using queues. Queues are an integral part of the HSA architecture. CPUs already send compute requests to each other in queues in popular task queuing run times like ConCRT and Threading Building Blocks. With HSA, both CPUs and CUs can queue tasks to each other and to themselves.

The HSA runtime performs all queue allocation and destruction. Once an HSA queue is created, the programmer is free to dispatch tasks into the queue. If the programmer chooses to manage the queue directly, then they must pay attention to space available and other issues. Alternatively, the programmer can choose to use a library function to submit task dispatches.

A queue is a physical memory area where a producer places a request for a consumer. Depending on the complexity of the HSA hardware, queues might be managed by any combination of software or hardware. Queue implementation internals are not exposed to the programmer.

Hardware-managed queues have a significant performance advantage in the sense that an application running on a CPU can queue work to a CU directly, without the need for a system call. This allows for very low-latency communication between devices, opening up a new world of possibilities. With this, the CU device can be viewed as a peer device, or a co-processor.

CPUs can also have queues. This allows any device to queue work for any other device.

### 4. CONCLUSION

The current state of the art of GPU/DSP and other high-performance computing is not flexible enough for many of today’s computational problems.

HSA is a unified computing framework. It provides a single address space accessible to both CPU and GPU (to avoid data copying), user-space queuing (to minimize communication overhead), and preemptive context switching (for better quality of service) across all computing elements in the system. HSA unifies CPUs and GPU/DSPs into a single system with common computing concepts, allowing the developer to solve a greater variety of complex problems more easily.

### 5. REFERENCES

- [1] Heterogeneous System Architecture: A Technical Review, Advanced Micro Devices, Rev. 1.0.
- [2] <http://developer.amd.com/resources/heterogeneous-computing/>

### About the author

Timour Paltashev is a professor at Northwestern Polytechnic University, College of Engineering and Senior Manager in Advanced Micro Devices. His contact email is [timpal@mail.npu.edu](mailto:timpal@mail.npu.edu).

# Rendering of transparent optically anisotropic objects

Victor A. Debelov, Dmitry S. Kozlov

Institute of Computational Mathematics and Mathematical Geophysics SB RAS,

Novosibirsk State University, Novosibirsk, Russia

{debelov, kozlov}@oapmg.sccc.ru

## Abstract

At the present time, computer graphics demonstrates a lot of achievements in physically correct rendering of opaque objects and optically isotropic transparent and translucent objects (such as glass, diamond, or water).

This paper examines the current state of art in the algorithms for physically accurate calculation of images of 3D scenes including transparent optically anisotropic objects (feldspar, KDP, etc.). It should be noted that very little work has been made in this field, with the most important of it in this century. Only rendering of transparent optically anisotropic objects (without absorption) is considered here.

**Keywords:** *photorealistic rendering, transparent objects, optical dispersion, optical anisotropy, birefringence, polarized light, crystals.*

## 1. INTRODUCTION

There exist several major algorithms for rendering of 3D scenes [1, 8, 39]: scanline algorithms like OpenGL and DirectX, ray casting, Whitted style ray tracing (WSRT), radiosity, and Monte Carlo ray tracing. Specific rendering algorithms are their combinations and modifications. It is well-known that all algorithms require some specifications of the materials of scene objects. The material of an object determines how light interacts with this object. A description of the parameters of light coming to a point of a scene or a scene object, that is, the directions of reflection and the fraction of light energy reflected, is called a Local Model of Light Interaction (LMLI) with an object. The recent developments of computer graphics and increase in computer power make it possible to use more complex materials and more sophisticated LMLIs to correctly characterize this interaction from a physical point of view.

Major attention will be given to crystalline scene objects: monocrystals and crystalline aggregates, although the algorithms being considered can be applied to any transparent medium with a piecewise-smooth boundary. It should be noted that the behavior of many natural objects that are normally optically isotropic can be anisotropic under certain physical conditions (external fields, etc.), that is, they demonstrate birefringence, for instance, even vacuum [35]. Therefore, an intention to more and more correctly render the various scenes may require to use the rendering algorithms considered here even for typical materials, such as glass. It is appropriate to recall the following Kajiya's statement [12]: "A thread that runs throughout computer graphics is the quest for detail. Nature presents a nearly infinite complexity and richness of form over an enormous range of scales. In image synthesis it is our task to make convincing pictures of such natural phenomena: thus how to represent this range of scales becomes a central problem."

The present paper is a condensed version of our talk. It is organized as follows: Section 2 provides a scheme of rendering based on ray tracing. Section 3 gives a brief insight into the evolution of the

materials. Section 4 deals with a Local Model of Light Interaction with crystals, and Section 5 presents developments of algorithms to render transparent optically isotropic and anisotropic objects. Section 6 is devoted to testing/verification. Section 7 discusses some rendering problems closely related to those described above.

## 2. RENDERING

Consider the problem of photorealistic rendering in the following statement.

A scene is a set of objects, surfaces, and light sources. They are located in the space of the scene. An object is given in some way, and has a closed boundary. Each boundary divides two media with specific optical characteristics. We calculate the scene image for a given location of the camera with the following input data:

- Procedures for finding the point of intersection of the ray with the object boundaries (or scene surfaces).
- At the boundary point, there exists a normal, and it can be calculated.
- At the boundary point, the materials of both media are specified.
- An LMLI is specified for each type of boundary between different objects/media.
- A material is specified at each point of every particular medium.
- A material is specified for the medium filling the scene space. The most-used approach is to assume that the scene medium is vacuum.

The process of rendering is based on the construction of light ray paths: from a source or camera to an object, from one object to another, or inside transparent or translucent objects. The most important characteristic of rays of the paths is the energy transferred by them. The resulting photorealistic image is obtained by calculating this energy.

## 3. TRADITIONAL MATERIALS

### 3.1 Bidirectional reflectance distribution function

A popular local model for characterizing the material of points of opaque surfaces is the bidirectional reflectance distribution function  $BRDF(\omega_i, \omega_r)$ , which shows the portion of energy coming from direction  $\omega_i$  and reflected in direction  $\omega_r$ . This is *isotropic reflection*, since it depends only on these angles. It has the property of Helmholtz reciprocity:  $BRDF(\omega_i, \omega_r) = BRDF(\omega_r, \omega_i)$ . The existing BRDFs can be roughly subdivided into those obtained from simulation and from geometrical and wave optics. In most photorealistic rendering algorithms, it is sufficient to specify the material only at the points of boundaries between the objects/media. There are more general BRDFs, for instance, anisotropic reflection ones [12], where, in addition to the angles of incidence and reflection, some special directions associated with the surface itself are taken into account.

The phenomenological BRDFs can be subdivided into:

- *Intuitive BRDFs*. Example: Phong's specular reflection formula [39]. It is clear intuitively (according to Phong) that  $\cos^{\text{shininess}}(\alpha)$  makes it possible to rather realistically display highlights in some applications.
- *Simulation BRDFs*, obtained in natural experiments with the material under study in a special setup that makes it possible to estimate even anisotropic reflection [38].

*Modeling BRDFs* are obtained, as a rule, on the basis of optical laws.

Within the class of opaque objects, some successful attempts have been made to create BRDFs for objects with subsurface light scattering (skin, marble, and even leaves) [9, 11].

For transparent and translucent objects, the material is described, in addition to the BRDF, also by the bidirectional transmission distribution function (BTDF), and the total behavior of light is represented by the bidirectional scattering distribution function (BSDF) at a surface point [22].

### 3.2 Spectral rendering

The recent increase in computer efficiency caused the practical usage of spectral rendering (for instance, MaxwellRender – <http://www.maxwellrender.com/>). In fact, a phenomenon like light dispersion cannot be simulated within the framework of a color model RGB to illustrate Newton's experiment on white light splitting, well-known from school physics. Here, more complicated specifications of materials in the form of spectra –  $BSDF(\lambda, \omega_i, \omega_r), \lambda = 380..780nm$  are used, that is, the light scattering function is specified for some set of waves from the visible spectrum. The spectra of light sources are also given. For this more precise model, the following problems are solved:

- Transformation of RGB images into spectral representation. This is a practical problem, since most initial data are given in the RGB format.
- Transformation of the spectral representation of an image into the RGB format (*tone reproduction*). The obtained result image in spectral form should be displayed using available RGB-devices with a poorer color range to obtain images producing the same impressions of the observer. Numerous approaches have been proposed to solve this problem (see [6]).

Obviously that researchers focused their attention on the rendering of glass products and diamonds [6, 18, 19] taking into account absorption in the object medium [20]. The laws of optics (the Buger-Lambert-Baer law) were used to derive a model of absorption [28].

### 3.3 Polarized light

Nevertheless, a lot of objects still have to be studied, since no physically correct models are available in computer graphics for the interaction of light with translucent objects and media, as well as with metals. This is due to the fact that the models in use had some assumptions, which were characterized by Hanrahan in the introduction chapter 2 to book [1] (1993) as follows: "The polarization of incident radiation is an important parameter affecting the reflection of light from a surface, but the discussion will be simplified by ignoring polarization". Up to now, only unpolarized light has been considered in commercial programs of rendering. As a rule, nothing is said about the polarization of light. This is a justified approach, since in typical scenes the light (except for solar light) is often weakly polarized [6].

The fact that some people can distinguish polarized light from unpolarized light cannot be ignored [17].

Light polarization in optics is a very important concept. However, of interest to us are only papers on photorealistic rendering. Using various kinds of light polarization, one can obtain a variety of images. Consider a graphic example demonstrating the concept of Brewster's angle (the angle of incidence at which full polarization of reflected light takes place) [27, 28] where the light reflected from a water surface is fully polarized. In this way, *full refraction can be achieved* ([http://ru.wikipedia.org/wiki/Закон\\_Брюстера](http://ru.wikipedia.org/wiki/Закон_Брюстера)).

## 4. LMLI OF CRYSTALS

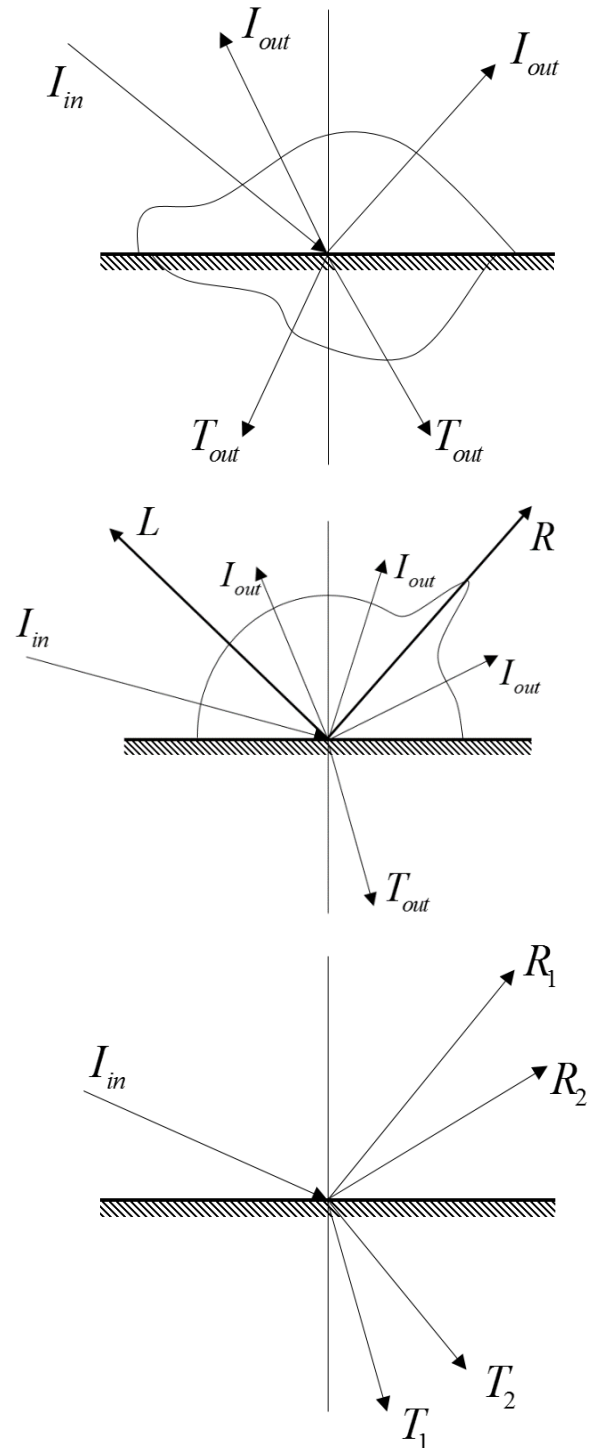


Figure 1: Different models of light ray-surface interaction.

Fig. 1 (top) presents a typical BSDF, which is the most general case of behavior of a ray (or path) after it falls on the surface of a translucent object. The possible directions of reflection  $I_{out}$  and refraction  $T_{out}$  and the portions of energy going in these directions are taken into account. A function that is typical for the Whitted algorithm of ray tracing is shown in the center of Fig. 1 [25, 39]. Here  $L$  is the direction to the point light,  $R$  is the corresponding vector of the reflected ray, and  $T_{out}$  is the vector of the refracted ray. Whereas for the incoming ray  $I_{in}$  the reflection distribution function is fully represented by the outgoing rays  $I_{out}$  in the model, refraction takes place only along the single ray  $T_{out}$ , which depends on  $I_{in}$  and is calculated by the Snell law. Fig. 1 (bottom) shows the most general case of response of a transparent crystal surface to an incoming ray  $I_{in}$ . Reflection in one or two directions of reflection ( $R_1$  and  $R_2$ ) and refraction in one or two directions of refraction ( $T_1$  and  $T_2$ ) are possible. Hence, the incident ray can give a maximum of only four directions of outgoing rays.

Thus, the BSDFs are piecewise-continuous functions of directions. This key property of the functions is a basis of many rendering algorithms. The continuity allows one to perform linear approximations, specify these functions with the help of tables, etc. As for crystals, this is the sum of delta-functions.

As for ray tracing in scene rendering, a ray must contain, in addition to geometrical information, some information about the polarization of light carried by this ray. Several concepts of polarization have been successfully used in solving the problem of rendering: coherence matrices, Stokes vectors, and Mueller matrices (see, for instance, [6, 28]).

From an optical point of view, all transparent and translucent media can be subdivided into isotropic and anisotropic ones. In isotropic (semi)transparent media (glass, diamond, water) all directions of light propagation are equal, whereas the anisotropic media effects on light propagating in different directions will differ from one another. Well-known representatives of anisotropic media are crystals, for instance, feldspar (calcite). Anisotropic media have specific directions, called *optical axes*. *Isotropic* media do not have optical axes. If a medium has only one axis, it is called *uniaxial*, and if it has two axes, it is called *biaxial*. An important property of optically anisotropic media is birefringence (that is, double imaging) and double reflection (see Fig. 1, bottom). Thus, one or two reflected and one or two refracted rays can be generated at the boundary of anisotropic media for a single incident ray.

Fully transparent media are important, because absorption (translucence, medium's color) and optical activity [28, 15, 16] are significant only inside the medium. However, the direction of (reflected and refracted) rays generated by an incident ray is calculated in the same way both in absorbing and transparent media.

The first important works that appeared in the last century are as follows:

- [27] – a mathematical model of the phenomenon of polarization which can be applied in computer graphics is constructed. This model is based on coherence matrices and takes into account a possibility of partial ray polarization, elliptic polarization. A possibility of polarization under reflection from surfaces of dielectrics with incidence under Brewster's angle is also considered. However, in this case

only reflection from isotropic dielectrics is taken into account. The question of the degree of polarization of a ray reflected from the surface of a birefringent crystal still remains open.

- [14] – the problem of ray incidence from an isotropic medium to a uniaxial one is considered. Formulas to calculate the directions of generated rays are presented.
- [15, 16] – these papers are devoted to ray tracing. The directions and degrees of polarization of generated (reflected and refracted) rays are calculated. Thus, a calculation of energy is performed, which makes it possible to correctly calculate the intensity of light energy. Uniaxial absorbing and active crystals are considered.
- [21] – an algorithm to simulate birefringence in uniaxial crystals, specifically, simulation of polarization effects using coherence matrices, is considered. **This is the first paper** which presents an image for a test scene including calcite and a polarizing filter; the *test image is colored*.

The following papers of interest appeared in the 21st century:

- In paper [32], the first attempt is made to formulate the problem of crystal rendering.
- Determination of generated rays (without calculation of polarization) is made in [13, 14, 33, 34, 40, 41].
- Determination of generated rays (with calculation of polarization) is made in [5, 10, 23, 24, 30, 36]. Paper [5] is an extension of [30, 36]. Paper [23] describes a theory, without any experiments.
- A test image is presented and calculated in [5, 10, 24, 30].

Papers [14–16, 23] were used in the development of LMLIs for crystals.

There are a lot of important papers that should be included in this context, but they are typically devoted to finer optical effects and are not mentioned in papers on computer graphics of anisotropic objects.

The calculation formulas are based on the physical laws of electromagnetic wave propagation by solving Maxwell's equations. The fundamental references are books [28, 42].

The derivation of major formulas for the calculation of generated rays is very important. Since the books on crystal optics usually do not care about applications of computer graphics and, therefore, do not contain the required formulas, the formulas are derived by specialists in photorealistic rendering. A geometrical / trigonometrical approach is mainly used, although in some papers matrix calculations are used [15, 16]. A covariant method [42] is used in [5, 30]. It should be noted that many of the papers do not consider the purely isotropic case at all.

The domain of applicability of a local model of light interaction is defined by the available formulas. Each object or medium in a scene can be characterized from an optical point of view in the following way: 0 – isotropic medium, 1 – uniaxial medium, 2 – biaxial medium. Then the types of boundary between objects and/or media can be denoted as  $i \Rightarrow j$ , that is, the LMLI calculates the generated rays if a ray falls from a medium  $i$  to the boundary with a medium  $j$ . All above-mentioned papers are devoted to LMLIs providing formulas for boundaries with one isotropic medium. Only paper [5] proposes a unified LMLI algorithm for any boundary type. In nature, minerals most often have the form of crystalline aggregates (“conjoined monocrystalline parts”). Thus, a LMLI for the boundary between two anisotropic media is necessary.

Also, one should mention an attempt to formalize a material in the form of a VRML [7] extension made in paper [30].

## 5. RENDERING OF CRYSTALS

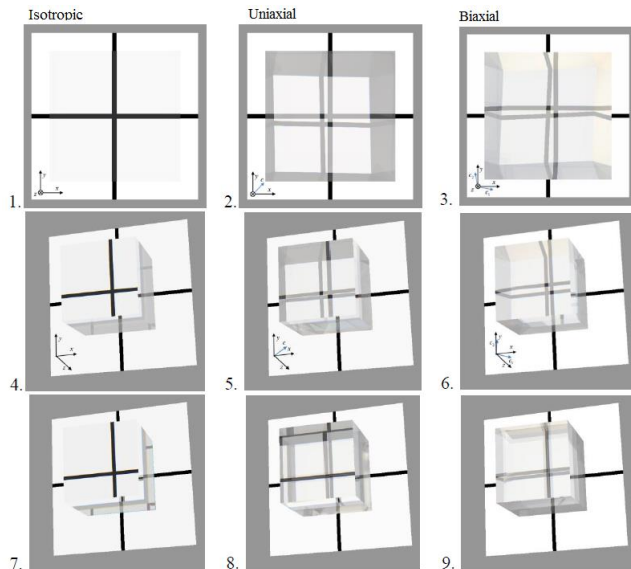


Figure 2: Variants of a virtual scene.

Real rendering of test scenes with crystals is presented in papers [4, 10, 13, 30]. Fig. 2 presents nine variants of a virtual scene consisting of a transparent cube. The cube is lying on a plane white square with a black cross drawn on it. The left column (cells 1, 4, 7) has cubes from isotropic materials, the middle column (cells 2, 5, 8) has cubes from uniaxial materials, and the right one (cells 3, 6, 9), cubes from biaxial materials. In the two upper rows of images (cells 1 – 6) the cube is made from a monocrystal, only for the top row the camera is located right in the center, and in the second row the camera looks from one side. In the lower row (cells 7 – 9) the cubes are aggregates. They consist of two layers - two half-cubes with different optical characteristics: cell 7 – layers with different refraction coefficients, cells 8 and 9 – layers with different directions of optical axes.

- The images in the left-column cells can be obtained using numerous commercial programs (for instance, Maxwell).
- Using algorithms [10, 24], one can obtain images in cells 2 and 5.
- Using algorithm [13], one can obtain "sketches" for images in cells 2, 3, 5, and 6. In this case the colors will not be correct, because energy balance cannot be calculated.
- Using algorithm [5], one can obtain any of the nine images.

Some more remarks should be made to provide a better understanding of the peculiarities of rendering scenes with transparent objects.

In the process of rendering, the following light ray paths are constructed: from a source to an object, from one object to another, inside transparent and translucent objects.

A polarized ray is considered in [5] as a structure containing:  $\lambda$  – light wavelength;  $P_0$  – ray origin point;  $\vec{r}_d$  – direction vector (ray vector);  $\vec{m}$  – phase propagation vector;  $CS$  – system of coordinates of the ray;  $J$  – 2x2 coherence matrix; ray type (isotropic, ordinary, extraordinary, fast, slow [28]).  $CS$  and  $J$  define the degree of ray polarization. All these parameters are calculated after determination of the directions of the generated rays.

With recursive algorithms of backwards tracing like the Whitted algorithm [25], a tree of backwards tracing is constructed. At each node, up to 4 descendants (Fig. 1, bottom), and not 2, as in the case of isotropic objects (Fig. 1, center), are generated. Only paper [31] describes an implementation of tracing and branching into 4 descendants. In the other papers, no attention is given to this question. Once the tree of paths from the camera is constructed, it is passed in the reverse order to collect the energy.

It is much more expensive to calculate the polarization of generated rays rather than the direction of the generated rays themselves. Therefore, polarization is not calculated in the tree construction. In the process of energy collection from the leaves (light sources) to the root, for every ray connecting a descendant node with its predecessor all its generated rays (as for an incident ray) and the degree of their polarization are fully calculated.

It should be noted that under reflection from a diffuse surface the ray becomes unpolarized, regardless of the state of polarization of the incident ray [27].

## 6. VERIFICATION

Traditionally, expert estimates (when one or other effect is correct from the point of view of an expert) are used to verify algorithms of photorealistic rendering. For instance, pioneering paper [21] shows that an image calculated for a photo of a tiger screened by a birefringent crystal demonstrates birefringence. Similarly, in papers [10, 24] it is proposed to compare visually photos with calculated images of corresponding virtual scenes.

A more accurate approach was used in the project Cornell Box [2] to demonstrate the real performance of a radiosity algorithm, where a calculated image and a photo are compared pixel-by-pixel. However, for this it was necessary to exactly specify all materials of the scene surfaces and camera setup.

In papers [4, 29], some problems of verifying the algorithms for rendering of (semi)transparent crystals are considered, and it is proposed to organize a common database of specific tests and rules. Paper [3] gives a sketch of such a base. An example of verification from [5] is given in Figs. 3-5.

## 7. OTHER PROBLEMS AND ALGORITHMS

In the above-mentioned papers, additional optical effects are considered:

- Fluorescence. This is a phenomenon where a substance changes the wavelength of light. In paper [26], some fluorescent paints are simulated.
- Pleochroism. This is a phenomenon where absorption in an anisotropic medium depends on the direction and state of polarization of a ray (see [36, 37]).
- Optical activity. Some crystals (for instance, quartz) rotate the plane of ray polarization during propagation through a crystal [15, 16].

## 8. ACKNOWLEDGEMENTS

This work was partially supported by the Russian Foundation for Basic Research (research project No. 12-07-00386 a).

## 9. REFERENCES

- [1] Cohen, M.F., Wallace, J.R. *Radiosity and Realistic Image Synthesis*. – Academic Press, New York, 1993.
- [2] Cornell Box: <http://www.graphics.cornell.edu/online/box>
- [3] Crystal tests: [http://oapmg.sccc.ru/temp\\_crystal\\_tests/](http://oapmg.sccc.ru/temp_crystal_tests/)

- [4] Debelov V.A., Kozlov D.S. *Rendering of Translucent Objects, Verification and Validation of Algorithms // Proceedings of 20th WSCG International Conference on Computer Graphics, Visualization and Computer Vision, part I, Plzen, Czech Republic, June 26 - 28, 2012, pp. 189-196.*
- [5] Debelov, V.A., Kozlov, D.S. *A Local Model of Light Interaction with Transparent Crystalline Media. // IEEE Transactions on Visualization and Computer Graphics, 2012. <http://doi.ieeecomputersociety.org/10.1109/TVCG.2012.304>*
- [6] Delvin, K., et al. *Tone reproduction and physically based spectral rendering // Eurographics'2002. STAR report.*
- [7] Fellner, D. W. *Extensible image synthesis. Object-Oriented and Mixed Programming Paradigms, – Focus on Computer Graphics. Springer, Wisskirchen P. (Ed.) – 1996, P. 7–21.*
- [8] Foley, J., van Dam, A., Feiner, S., Hughes, J. *Computer Graphics: Principles and Practice // Addison Wesley, 1996.*
- [9] Hanrahan, P., Krueger, W. *Reflection from layered surfaces due to subsurface scattering. In ACM Computer Graphics (SIGGRAPH'93), pages 165–174.*
- [10] Guy, S., Soler, C. S. *Graphics gems revisited. // ACM TOG, Vol. 23, No. 3, pp. 231–238, 2004.*
- [11] Jensen, H. W., Buhler, J. *A rapid hierarchical rendering technique for translucent materials. // ACM TOG, Vol. 21, No. 3, pp. 576–581, 2002.*
- [12] Kajiyu, J.T. *Anisotropic Reflection Models // Proc. SIGGRAPH '85, P. 15-21.*
- [13] Latorre, P., Seron, F. J., and Gutierrez, D. *Birefringence: calculation of refracted ray paths in biaxial crystals // Visual Computer, Vol. 28, No. 4, pp. 341-356, 2012.*
- [14] Lekner, J. *Reflection and refraction by uniaxial crystals. // Condens. Mat. Vol. 3, No. 32, pp. 6121–6133, 1991.*
- [15] McClain, S.C., Hillman, L.W., Chipman, R.A. *Polarization ray tracing in anisotropic optically active media I, algorithms. // Applied Optics, vol. 10, no. 11, pp. 2371-2382, 1993.*
- [16] McClain, S.C., Chipman, R.A. *Polarization ray tracing in anisotropic optically active media II, theory and physics. // Applied Optics, vol. 10, no. 11, pp. 2383-2393, 1993.*
- [17] Minnaert, M. *Light and Color in the Open Air. Dover, 1954.*
- [18] Sun, Y., Fracchia, F. D., Drew, M. S. *Rendering light dispersion with a composite spectral model // In Proc. Of International Conference on Color in Graphics and Image Processing CGIP'2000.*
- [19] Sun, Y., Fracchia, F. D., Drew, M. S. *Rendering diamonds. // In Proc. of 11th Western Computer Graphics Symposium (WCGS) , pp. 9-15, 2000.*
- [20] Sun, Y., Fracchia, F. D., Drew, M. S. *Rendering the phenomena of volume absorption in homogeneous transparent materials // In 2nd Annual IASTED International Conference on Computer Graphics and Imaging (CGIM'99), 283–288. <http://fas.sfu.ca/pub/cs/mark/Cgim99/volumeAbs.ps.gz>.*
- [21] Tannenbaum D.C., Tannenbaum P., Wozny M.J. *Polarization and birefringency considerations in rendering. // Comp. Graphics (SIGGRAPH '94 Proceedings), ACM Press, 1994. PP. 221-222. (Extended version available on CD-ROM – 9 P.)*
- [22] Veach, V, E. *Robust Monte Carlo methods for light transport simulation // PhD thesis. Stanford University. 1997.*
- [23] Wang Y. et al. *Complex ray tracing in biaxial anisotropic absorbing media. // Journal of Optics A: Pure and Applied Optics vol. 10(7), p. 075009 (7pp).*
- [24] Weidlich A., Wilkie A. *Realistic Rendering of Birefringency in Uniaxial Crystals // ACM TOG, Vol. 27, No. 1, pp. 6:1-6:12, 2008.*
- [25] Whitted, T. *An Improved Illumination Model for Shaded Display // Commun. ACM, Vol. 23, No. 6, 1980. P. 343-349.*
- [26] Wilkie, A., Tobler, R.F., Purgathofer, W. *Combined Rendering of Polarization and Fluorescence Effects // Proc. of the 12th Eurographics Workshop on Rendering Techniques, London, UK, June 25-27, 2001, 197-204.*
- [27] Wolff, L., Kurlander, D. J. *Ray tracing with polarization parameters. // IEEE Computer Graphics and Applications, Vol. 10, No. 6, PP. 44-55, 1991.*
- [28] Борн М., Вольф Э. // *Основы Оптики. Москва, "Наука", 1973.*
- [29] Дебелов В.А., Козлов Д.С. *Верификация алгоритмов фотореалистического рендеринга кристаллов // Тр. Графикон-2010, Санкт-Петербург.*
- [30] Дебелов В. А., Козлов Д. С. *Локальная модель взаимодействия света с изотропными и одноосными прозрачными средами // Вестник НГУ, серия "Информационные технологии", Том 10, Выпуск 1. – 2012. – С. 5-23.*
- [31] 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, Protvino-Altai. - Изд-во ИФТИ, Протвино. - С. 15-22.*
- [32] Дебелов В.А., Саттаров М.А. *Проблемы реалистической визуализации кристаллов // Тр. Графикон-2003. Москва.*
- [33] Дебелов В., Саттаров М. *Расчет фотореалистических изображений сцен с кристаллами: проблемы и пути решения // Вычислительные технологии, том 9, Вестник КазНУ им. Аль-Фараби, Серия математика, механика, информатика N 3 (42). – Алматы-Новосибирск, 2004.*
- [34] Дебелов В.А., Саттаров М.А. *Модели и алгоритмы для фотореалистической визуализации сцен с кристаллами. // Тр. Графикон-2005, Новосибирск.*
- [35] Железняков В.В. *Проблемы современной астрофизики // Соросовский Образовательный Журнал, 1996, № 6, С. 83-91.*
- [36] Козлов Д.С. *Алгоритм фотореалистического 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](http://www.ict.nsc.ru/ws/YM2008/14308/Kozlov_YM2008.pdf)*
- [37] Козлов Д. *Алгоритм рендеринга по методу Монте-Карло петрографических шлифов одноосных кристаллов // Тр. Школы молодых ученых. Графикон-2009. Москва.*
- [38] Летунов А.А., Барладян Б.Х. и др. *Прибор для измерения bdf на основе пзс-камеры для использования в компьютерной графике // Тр. Графикон-2001.*
- [39] Роджерс Д. *Алгоритмические основы машинной графики // Пер. с англ. - М.: Мир, 1989.*
- [40] Саттаров М.А. *Моделирование двойного лучепреломления в одноосных кристаллах // Тр. Конференции Молодых Ученых ИВМиМГ СО РАН. – Новосибирск. – 2004. – С. 176–182.*
- [41] Саттаров М.А. *Моделирование двойного лучепреломления в двуосных кристаллах // Программа и тезисы докладов V Всероссийской конф. молодых ученых по*

математическому моделированию и информационным технологиям. – Новосибирск, ИВТ СО РАН, 2004. – С. 29. См. также <http://www.ict.nsc.ru/ws/YM2004/8596/Sattarov.htm>

[42] Фёдоров Ф.И., Филиппов В.В. *Отражение и преломление света прозрачными кристаллами* // Минск, "Наука и техника", 1976.

### About the authors

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 the Novosibirsk State University. His contact email is [debelov@oapmg.sccc.ru](mailto:debelov@oapmg.sccc.ru).

Dmitry S. Kozlov is a Ph.D. student of the Novosibirsk State University and engineer of the Institute of Computational Mathematics and Mathematical Geophysics SB RAS. His contact email is [kozlov@oapmg.sccc.ru](mailto:kozlov@oapmg.sccc.ru).



Figure 4: Calculated image of a virtual scene.



Figure 3: Photo of a real scene.

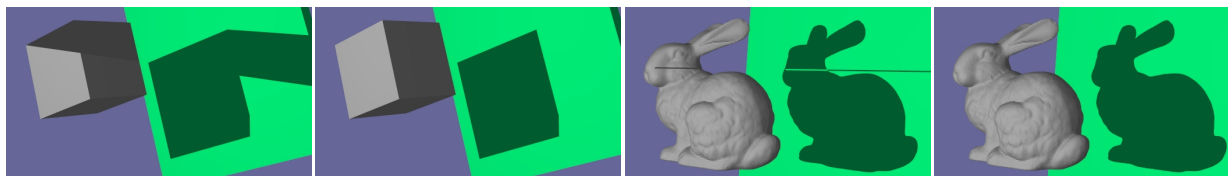


Figure 5: Pixel-by-pixel difference of images.

# Robust Silhouette Shadow Volumes on Contemporary Hardware

Jan Pečiva, Tomáš Starka, Tomáš Milet, Jozef Kobrtek, Pavel Zemčik

Faculty of Information Technology Brno University of Technology Czech Republic \*



**Figure 1:** The Figure shows the difference between the original algorithm and our robust algorithm. The right image of each couple shows result of our robust algorithm. The first couple of images shows very simple model, where artefacts are most visible. The second couple shows artefacts on more complex model, which could appear in real applications.

## Abstract

The paper describes an algorithm, which produces shadow volumes for an arbitrary triangle model without visual artifacts. The algorithm has been implemented, optimized, and evaluated for a number of contemporary hardware platforms. The main contribution of the paper is removal of visual artifacts caused by limited precision of floating point arithmetics. The paper also presents an overview of the implementation and result of the optimizations on individual platforms. Finally, the conclusions are drawn and the future work is outlined.

**CR Categories:** I.3.3 [Computer Graphics]: Three-Dimensional Graphics and Realism—Display Algorithms I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Radiosity;

**Keywords:** shadow volumes, silhouette, OpenCL, GPGPU, geometry shader

## 1 Introduction

Shadow volumes (SV) method is a traditional and popular method for shadow casting in computer graphics. It has been introduced in 1977 [Crow 1977]. Later on, in 1991, SV algorithm was implemented in stencil buffer hardware [Heidmann 1991]. This implementation is generally called z-pass method. This method, however, is not robust. It produces incorrect shadows when the observer is inside a SV itself. This problem was addressed in 2002 through a method called z-fail [Everitt and Kilgard 2002].

An alternative for rendering shadows is shadow mapping [Williams 1978]. It is very frequently used as it generally offers high performance; however, the shadow maps approach suffers from visual imperfections caused by the limited shadow map resolution. The shadow map approach is massively used in game industry where high performance is critical and scenes can be adjusted so that visual artefacts are not too visible or do not occur. On the other hand, these features limit the applicability of the approach in e.g. CAD systems, where the scene is given by the model being constructed and may be incompatible with the requirements of the shadow maps approach to work well.

While the SV approach produces per-pixel correct results, they are affected by some performance issues. In its simple form, the applications were using a SV generated for each triangle of the ob-

ject geometry, which could be very rasterization demanding. This fact lead to development of more sophisticated methods, which construct a SV only from the possible silhouette edges of the occluder geometry, that has positive impact on fill-rate. The first one to propose an algorithm of silhouette extractions using the consumer graphics hardware was [Brabec and Seidel 2003]. Later on, [McGuire et al. 2003] managed to implement the algorithm solely using vertex shaders. Furthermore, [van Waveren 2005] extracted the silhouette edges using the SSE2 instructions on CPU. Finally in [Stich et al. 2007] the geometry shaders were used for silhouette extraction.

Many of the of these algorithms expect 2-manifold objects as their input. More general objects were considered by [Bergeron 1986], who focused on manifold objects with a boundary edge case. [Aldridge and Woods 2004] further removed constraints on the input model but oriented non-manifold meshes are expected. Finally, [Kim et al. 2008] presented the algorithm that works with any non-manifold mesh objects. The overview of above methods can be found in [Kolivand and Sunar 2013]

Unfortunately, during the implementation of the algorithm presented in [Kim et al. 2008] we found out that the algorithm is not completely robust and often produces result with visual artefacts. This is caused by limited precision of floating point arithmetics. Thus, we developed the proposed robust algorithm, based on [Kim et al. 2008], that is free from visual artifacts. Additionally, we optimized the algorithm for a number of contemporary hardware platforms, such as modern CPUs and GPUs. The paper describes the algorithm, presents an overview of the implementation of the proposed improvements and optimizations and assesses performance. Finally, the conclusions are drawn and the future work is outlined.

## 2 Algorithm description and Robustness improvement

The algorithm, described in [Kim et al. 2008], generates the output shadow silhouette based on the triangular mesh representing the model and the position of light source.

### 2.1 Description of the algorithm

The algorithm can be briefly described as follows:

**Input:** model represented as a triangular mesh and light source position.

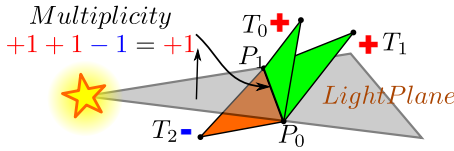
\*e-mail: {pecita, starka, imilet, ikobrtek, zemcik}@fit.vutbr.cz



**Output:** silhouette represented by a set of edges selected from the input model.

**Algorithm:**

1. The triangular model is converted into an edge representation. Every edge occurs only once even though it is shared among more triangles. Each edge in the new representation is described by its vertices and list of all opposite vertices (OVs). The OVs are vertices of the triangles sharing the edge that do not belong to the Edge. See Figure 2.
2. For each edge from the edge set, an oriented light plane (LP) is evaluated from edge vertices and the position of the light source.
3. For each OV belonging to the edge, multiplicity is calculated as +1 or -1 depending on which side of LP the OV lies. If the OV lies exactly on the LP, its multiplicity is 0. The final multiplicity of an edge is given by the sum over the multiplicities of every OV. See Figure 2.
4. Finally, the set of edges forming the silhouette is a subset of all the edges such that their multiplicity is not 0.



**Figure 2:** Multiplicity of Edge P0-P1 for the Opposite Vertices (OVs) T0-T2.

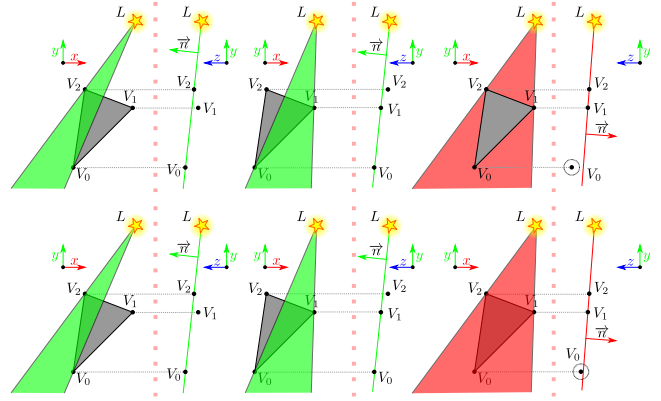
## 2.2 Implementation and problems

The above mentioned algorithm processes the model with the "by edge" approach. The multiplicity could also be calculated with the "by triangle" approach (with identical results). In parallel implementation, the "by edge" approach, used in this paper, is better than "by triangle" implementation, although the later may seem more natural. The main reason is that the edges are independent to each other, so this avoids concurrent memory writes. While the "by triangle" approach would lead to usage of atomic operations. Therefore, the "by edge" implementation is exploited.

The algorithm assumes that the evaluation of multiplicity is consistent within each triangle. Unfortunately, this is not the case for floating point arithmetics used in HW. Let us consider an example as shown in Figure 3. In the "by edge" approach, the multiplicities could be evaluated inconsistently for the triangle which is (almost) parallel to the LP. While the error was demonstrated for a single triangle model, such error can occur in a more complex model for individual triangles and ruin the whole silhouette leading into visible artefacts in shadows (see Figure 1).

## 2.3 The proposed robust algorithm

The proposed algorithm resolves the above issue connected with the inconsistency of triangle edges multiplicity evaluation. The main idea of the improvement is that the triangles, where the inconsistency can occur, are removed from the silhouette calculation. Because these triangles are (almost) parallel with the LP (their shadow volume would be zero), they cannot affect the shape of the resulting shadow. In fact, the removal of the triangles is equivalent to evaluation of its edges multiplicity to 0 which would occur in triangles parallel to the LP if the precision was not limited.



**Figure 3:** The grey triangle generates a SV. The green and red triangles represent the two orientations of the light plane (LP). The lower part shows erroneous calculation. In the upper right part, triangle is in front of LP with respect to the viewer. In the lower right, this is incorrectly evaluated, therefore the vertex V0 is assumed to be behind.

The question is "What is the most efficient way to remove such triangles?". Note that while the "by triangle" approach permits to simply discard the computed triangle, the "by edge" approach does not. One possible solution would be to evaluate "how close to parallel" the triangle is to the LP but in this case, the evaluation would have to be consistent for each triangle edge leading more or less to the same problem. Therefore, a solution was taken to "simulate" evaluation of the "other two edges" of the triangle formed by the edge and each OV. Our modification to the original algorithm is modification of step 3:

3. For each OV belonging to the edge a triangle is formed from OV and the edge. The multiplicity is evaluated for **every side** of this triangle and its remaining vertex as +1 or -1 depending on which side of LP the vertex lies. If the vertex lies exactly on the LP, its multiplicity is 0. If the evaluation of the multiplicity is inconsistent, the triangle is disregarded (the OV multiplicity is set to 0). Note also, that the same order of vertices and edges in triangles must be preserved for each edge evaluation. The final multiplicity of an edge is given by the sum over the multiplicities of every OV. See Figure 2.

The actual multiplicity evaluation for the edge AB for the light source position  $\mathbf{L}$  and set  $\mathcal{D}$  of all OV, each in homogeneous coordinates, is as follows:

The LP itself is defined as:

$$\begin{aligned} \mathbf{V} &= (\mathbf{L}_x - \mathbf{A}_x \mathbf{L}_w, \mathbf{L}_y - \mathbf{A}_y \mathbf{L}_w, \mathbf{L}_z - \mathbf{A}_z \mathbf{L}_w) \\ \mathbf{N} &= \text{normalize}((\mathbf{A} - \mathbf{B}) \times \mathbf{V}) \\ \mathbf{LP} &= (\mathbf{N}_x, \mathbf{N}_y, \mathbf{N}_z, -\mathbf{N} \cdot \mathbf{A}) \end{aligned} \quad (1)$$

The multiplicity of the edge is:

$$m = \sum_{\mathbf{o} \in \mathcal{D}} \text{sgn}(\mathbf{LP} \cdot \mathbf{o}) \quad (2)$$

Where  $|m|$  denotes the number of times the side of SV, extruded from this particular edge, is actually drawn/rendered.

Of course, the evaluation of the above set of expressions, for each edge of the triangle (instead of just once for each triangle), introduces a computational overhead. While some subexpression results can be reused, a significant overhead remains. However, it turns

	GF650Ti	AMD7950	HD3000	HD4000
CPU	4.9 / 5.0	12.9 / 13.9	9.9 / 11.2	6.3 / 5.9
AVX+OMP	4.9 / 5.1	11.8 / 13.5	10.0 / 12.4	6.6 / 8.6
GS	30 / 40	92 / 98	n/a	4.4 / 8.7
OpenCL	42 / 41	83 / 80	n/a	7.6 / 7.7

**Table 1:** Results of experiment (the first value shows FPS of the robust implementation, the second value is the original implementation)

out that the cost of additional arithmetics, especially in case of exploitation of powerful computational platform, is less costly than increased memory traffic or synchronization operations needed in alternative approaches.

### 3 Experiments and Results

The experiments were conducted in order to evaluate the achieved results, to assess feasibility of exploitation of the presented approach at the above mentioned platforms and in different applications, and to verify that the approach works well.

The performance of the algorithm was tested under these conditions: the robust version compared to the original version with no robustness, objects of simple shape compared to more complex shapes with similar number of triangles, varying levels of geometric complexity of the same object and scenes containing several separated objects.

#### 3.1 Robust versus traditional implementation

The purpose of this test was to evaluate whether at all and how much the implementation of the robustness of the algorithm adversely affects performance. The conditions for the test were made similar to the real applications conditions in terms of the size of the scene ( $\sim 10^5$ - $10^6$  triangles). The test was performed on a scene that did not exhibit significant occurrence of visual artefacts caused by the traditional implementation not being robust.

The results of experiments are shown in the Table 1. Note that performance of the robust algorithm was slightly worse than in the traditional implementation (the adverse effect of additional calculations was less than 10% at all platforms) but this is true only in case of scenes exhibiting no or little visual artefacts. In scenes with larger amount of artefacts, performance of the robust implementation was mostly better and the larger the amount of artefacts, the worse the traditional implementation perform also from the computational time point of view probably due to the fact that the artefacts caused increase in the fill rate. Overall, quite surprisingly, the decrease of performance in the robust method is not a problem on any platform.

#### 3.2 Simple versus complex shapes

The purpose of the next test was to evaluate how the shape of the objects influences the rendering times. Therefore, scene consisting of simple shapes, spheres, and scene consisting of complex shapes, bunnies, similarly large in terms of triangles, were compared. The size of the scene in this case was about the usual application size ( $\sim 6.5 \cdot 10^5$  triangles). One measurement was performed for the scene consisting of more (10) objects, one for the scene consisting of a single object but with the same complexity as in the previous case; this was done in order to check whether the number of objects affects the calculation speed.

	GF650Ti		AMD7950		HD3000		HD4000	
CPU	4.9	5.0	12.9	7.8	9.9	<b>11.0</b>	6.3	5.4
	5.5	5.5	13.3	9.6	12.2	<b>10</b>	9.6	5.5
AVX+OMP	4.9	5.1	12.9	4.3	9.9	6.2	6.3	6.0
	5.5	5.5	12.8	5.2	<b>12.8</b>	5.1	<b>11.9</b>	<b>6.2</b>
GS	30	30	<b>92</b>	95	n/a	n/a	4.4	n/a
	34	34	<b>156</b>	124	n/a	n/a	6.0	n/a
OpenCL	<b>42</b>	<b>52</b>	83	<b>96</b>	n/a	n/a	<b>8.1</b>	<b>8.2</b>
	<b>54</b>	<b>67</b>	142	<b>147</b>	n/a	n/a	11.3	6.0

**Table 2:** Results for one particular GPU and one platform consisting of 4 tests on different scenes: 10 individual bunnies (top left), 10 baked bunnies (top right), 10 individual spheres (bottom left), and 10 baked spheres (bottom right).

	GF650Ti		AMD7950		HD3000		HD4000	
CPU	3.9	35	5.1	81	<b>5.3</b>	<b>34</b>	<b>37</b>	20
	3.4	27	8.4	61	6.5	32	4.4	18.5
AVX+OMP	3.9	35	5.7	81	4.9	32	4.2	<b>23</b>
	3.4	28	9.3	63	<b>6.7</b>	<b>34</b>	<b>5.0</b>	<b>24</b>
GS	23	165	85	<b>650</b>	n/a	n/a	3.7	15
	<b>18.7</b>	<b>100</b>	<b>84</b>	<b>482</b>	n/a	n/a	3.1	19.3
OpenCL	<b>43</b>	<b>192</b>	<b>91</b>	430	n/a	n/a	<b>6.0</b>	19.3
	13.4	26	22	54	n/a	n/a	4.4	9.1

**Table 3:** Computational performance based on the number of triangles in the scene. Results for one particular GPU and one platform consisting of 4 tests on different scenes: one sphere with  $10^6$  (top left), one sphere with  $10^5$  triangles (top right),  $10 \times 10$  spheres each with  $10^4$  (bottom left), and  $10 \times 10$  spheres with  $10^3$  triangles (bottom right).

#### 3.3 Number of triangles in the scene

The consequent test was focused on the behavior of the algorithm in rendering of the scene depending on the number of triangles in the scene. The goal was to learn how the algorithm performs when the number of triangles changes from relatively low ( $\sim 10^5$  triangles) to relatively high ( $\sim 10^6$  triangles).

It can be seen that, as expected, performance of the algorithm decreases with the increased size of the object in terms of triangles. However, what was not as expectable is the fact that on different platforms, performance does not decrease uniformly and also that performance is not affected uniformly in different implementations on the same platforms. Additionally, in cases where the number of triangles is large, performance is also adversely affected by subdivision of the scene into separate objects as described below.

#### 3.4 Number of isolated objects in the scene

The goal of this final test was to demonstrate how the algorithm performs in dynamic scenes where typically a scene is composed from a number of ( $\sim 10$ - $100$ ) independently movable objects and cannot be represented by a single object to enable for easy independent motion of the objects. The test was performed along with the above mentioned testing of performance with changing number of triangles in the scene. It shows that the number of isolated objects does have impact on results especially in case of some platforms, probably due to synchronization and data transfer issues.

#### 3.5 Hardware platforms

The implementation was tested on following platforms:

- AMD Radeon HD 7950 (driver version: 13.1)
- GeForce 650 Ti (driver version: 314.07)
- HD3000 integrated Intel GPU (driver version: 9.17.10.3062)
- HD4000 integrated Intel GPU (driver version: 9.18.10.3071)

First three GPUs were tested with Intel i7-2600K with 16GB RAM. The HD4000 was tested with Intel i5-3570K with 8GB RAM. All the measurements were carried out on Windows 7 x64 SP1.

### 3.6 Interpretation of results

Some of the results observed in the above tests are especially worth mentioning and they are listed below:

- Performance of "OpenCL" implementation is very good on all the platforms for which OpenCL is available at all. The current solution heavily suffers from the synchronization between OpenCL and OpenGL contexts. This mainly occurs in scenes containing many separate objects because synchronization must be performed for each object. Unfortunately, OpenCL solution is not supported by the HD3000.
- The "Geometry Shader" implementation performs well in scenes with many separate objects. The performance declines with increasing complexity of models more than in OpenCL implementation.
- CPU+AVX+OMP performance is sometimes surprisingly less than the standard CPU implementation. On the other hand, measurements carried out on i5-3570K (Ivy Bridge architecture) showed 4-30% performance increase (16% in average), compared to standard one. Despite the fact that our CPU is capable of processing 8(4) threads concurrently, maximum performance increase is only 30% in its peak. The reason is that not all parts of the algorithm can be parallelized or rewritten using AVX intrinsics.

It is most suitable to use OpenCL implementation (where available) for scenes which do not contain too many (less than 100) objects. Geometry shader solution can be used in situations where the scene contains larger number of separated objects. CPU+AVX+OMP implementation should be used on modern CPUs in situations where the above solutions are not available. Standard CPU implementation should be used otherwise.

## 4 Conclusions

This paper presented a novel approach to Shadow Silhouette Shadow Volumes that leads into a more robust implementation, which has been tested and evaluated on a number of different hardware platforms.

The proposed approach proved to be working well and producing quality shadows with no visual artifacts. At the same time, it exhibits high performance in variety of hardware platforms. As shown in the paper, it can be efficiently implemented in CPU both using the traditional instructions and the SIMD instructions as well as in GPU using Geometry Shaders as well as using OpenCL.

The most efficient achieved implementation was in OpenCL for a scene containing  $10^6$  triangles, followed by the Geometry Shader implementation that is usable also with Intel HD 4000 platform. However, the OpenCL implementation suffers from synchronization slowdowns in case the scene is divided into more independent objects. As for the CPU implementations, while they are generally lower performance than the GPU ones, the SIMD instructions

(AVX) and parallelism boosts performance on the latest CPU architectures.

Overall, the proposed approach performs very well and at the same time it is robust and precise in terms of per pixel precision of the shadows. Therefore, it represents a very good alternative to shadow methods.

Future work includes improvements of the OpenCL implementation in terms of synchronization in scenes containing more objects, general improvements of the triangle tests. The future work also includes more thorough evaluation on more platforms and more scenes.

## Acknowledgements

The work has been made possible thanks to the co-funding by the IT4Innovations Centre of Excellence, Ministry of Education, Youth and Sports, Czech Republic, MŠMT, ED1.1.00/02.0070, V3C - Visual Computing Competence Center, Technology Agency of the Czech Republic, TAČR, TE01020415V3C, and RODOS - Transport systems development centre, Technology Agency of the Czech Republic, TAČR, TE01020155.

## References

- ALDRIDGE, G., AND WOODS, E. 2004. Robust, geometry-independent shadow volumes. In *Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, ACM, New York, NY, USA, GRAPHITE '04, 250–253.
- BERGERON, P. 1986. A general version of crow's shadow volumes. *IEEE Computer Graphics and Applications* 6, 17–28.
- BRABEC, S., AND SEIDEL, H.-P. 2003. Shadow volumes on programmable graphics hardware. *Computer Graphics Forum (Eurographics) 2003*, 433–440.
- CROW, F. C. 1977. Shadow algorithms for computer graphics. In *Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH '77, 242–248.
- EVERITT, C., AND KILGARD, M. J. 2002. Practical and robust stenciled shadow volumes for hardware-accelerated rendering.
- HEIDMANN, T. 1991. Real shadow real time. *IRIS Universe*, 28–31.
- KIM, B., KIM, K., AND TURK, G., 2008. A shadow volume algorithm for opaque and transparent non-manifold casters.
- KOLIVAND, H., AND SUNAR, M. S. 2013. A survey of shadow volume algorithms in computer graphics. *IETE Tech Rev 2013* 30, 38–46.
- MCGUIRE, M., HUGHES, J. F., EGAN, K., KILGARD, M., AND EVERITT, C. 2003. Fast, practical and robust shadows. Tech. rep., NVIDIA Corporation, Austin, TX, Nov.
- STICH, M., WÄCHTER, C., AND KELLER, A. 2007. Efficient and robust shadow volumes using hierarchical occlusion culling and geometry shaders. In *GPU Gems 3*, Addison Wesley Professional, H. Nguyen, Ed., 239–256.
- VAN WAVEREN, J., 2005. Shadow volume construction.
- WILLIAMS, L. 1978. Casting curved shadows on curved surfaces. *SIGGRAPH Comput. Graph.* 12, 3 (Aug.), 270–274.

# Mathematical Models of Fingerprints on the Basis of Lines Description and Delaunay triangulation

Vladimir Gudkov, Chelyabinsk State University, Chelyabinsk, Russia  
diana@sonda.ru

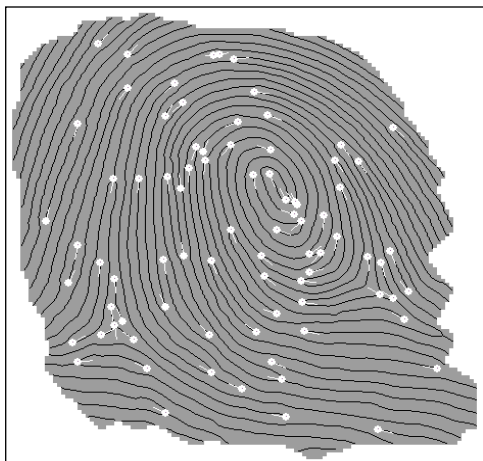
Marina Gavrilova, University of Calgary, Canada  
marina@cpsc.ucalgary.ca

**Abstract.** This paper presents a new mathematical model of fingerprints based on representing ridge lines as topological vectors and utilizing Delaunay triangulation for identification. The ridges are stored in the templates with the list of minutiae. Templates are used to identify the fingerprint. This leads to up to 10 times speed-up in processing time while retaining the high degree of identification precision.

**Keywords:** *Fingerprint, minutiae, topology vector, Delaunay triangulation.*

## 1. INTRODUCTION

Fingerprint images (FI) identification is realized on the basis of the templates, containing the pattern features. The basis for template identification is minutiae representation in the form of beginnings and endings, junctions and bifurcation of lines [1, 3, 4]. These can be detected by gray image, though in the process of template creation they are guided by the lines skeleton [2, 7, 8] (fig. 1).



**Fig. 1.** Skeleton and minutiae points

The mathematical model of the image should depend on necessary and sufficient quantity of features [2]. Minutiae and ridge count between minutiae are reputed as informative in dactyloscopy [7]. However, this count is not present in majority of mathematical models, used in biometrics for automatic proof of pattern uniqueness [8]. Each of such models is focused on increasing the identification accuracy, however suffers from drawbacks of only taking into account limited topological information [5, 6]. For example, classical ridge count, which should be counted along the straight line according to criminalistics, is rarely used due to inherent complexity of processing patterns with curvatures in the area of loops, deltas and whorls [7, 10]. This paper addresses the problem by proposing novel representation and treatment of curved ridge lines through topological vector representation and Delaunay triangulation based methodology.

## 2. PRELIMINARIES

Voronoi diagram and Delaunay triangulation methods continue to receive compelling attention in the various areas of research, and most recently, in the area of biometrics. From an early article framing computational geometry research as one of powerful vehicles to enhance biometric recognition methods [1, 3], to gamut of current research utilizing geometric properties of biometric data sets [15], the number of attempts to extract geometric information and apply topology to solve biometric problems has increased significantly. There has been research on application of topological methods, including Voronoi diagrams, for variety of biometric recognition systems [3, 6, 17]. In correlation-based matching, two fingerprint images are superimposed and the correlation between corresponding pixels is computed for different alignments [13]. During minutiae-based matching, the set of minutiae are extracted from the two fingerprints and stored as sets of points in the two dimensional plane [3, 4, 15]. While minutiae-matching is most widely used approach in fingerprint recognition, ridge feature-based matching based on orientation map, ridge lines and ridge geometry is frequently overlooked due to complexity of matching algorithm implementation [4].

There have been a number of attempts to utilize Voronoi diagrams in biometric research in other application domains. Voronoi diagrams were used for face partitioning onto segments and facial feature extraction in [17]. A method for binary fingerprint image denoising based on Distance Transform realized through Voronoi method was introduced in [14]. Bebis et. al. [1] used the Delaunay triangle as the comparing index in fingerprint matching. Their method works under assumption that at least one corresponding Delaunay triangle pair can be found between the input and template fingerprint images. As it has been shown in [16], this assumption simply does not hold due to low quality of fingerprint images, distortion in conditions under which fingerprint is obtained, or poor performance of the feature extraction algorithm. Another research supports this position by showing that even small local deformation can cause global deformation up to forty five in edge length [5]. The research presented in this paper takes advantage of additional information, which is ignored by fingerprint matching algorithms – ridge geometry. It is based on ridge line representation as topological vectors and a clever utilization of Delaunay triangulation, which results in increased speed and high recognition capability of the system. The method was fully integrated in a commercial software system and is described by a number of patents [10-12].

## 3. PROPOSED METHODOLOGY

The templates as a set of FI features are varied in different software systems. Some templates formats are limited in minutiae quantity [7]. Some features of templates are irrelevant, but it is possible to indicate their common property: the templates have features, being some metrics for minutiae points. These metrics are ridge counts between minutiae and topological vector for minutiae [7, 10, 11].

In this paper, an image template is synthesized in the form of

$$\Gamma: F_0^{(m)} \rightarrow \{L_m, L_l, L_a\}, \quad (1)$$

where  $F_0^{(m)} = [f_0^{(m)}(x, y)]$  – image skeleton (fig. 1);  $L_m$  – minutiae list;  $L_l$  – topological vectors list for lines;  $L_a$  – accelerator 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 and accelerator vectors for line are determined using minutiae, which are read from the skeleton nodes as nodes of the graph. These vectors are derivative from the minutiae. However, all these vectors characterize not the area of a separately selected point, but common properties of line as point's multitude or a line segment. In spite of the fact that the lists  $L_l$  and  $L_a$  characterize FI differently, they are alike in that they represent description for all pattern lines, but not for all points of the lines. This interesting property allows us to synthesize compact templates for very fast matching.

### 3.1. Minutiae list

We now describe how minutiae list is formed for subsequent fingerprint recognition. Let  $M_i$  – is minutiae which is indexed to number  $i$ . The minutiae list  $L_m$  is in the following form

$$L_m = \{M_i = \{(x_i, y_i), \alpha_i, t_i, v_i, \theta_i, p_i, h_i\} | i \in 1..n_1\}, \quad (2)$$

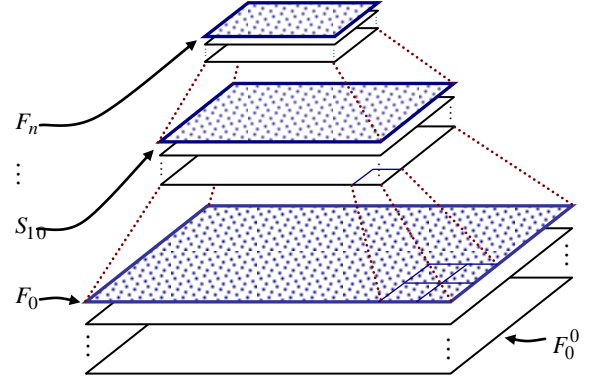
where  $|L_m| = n_1$  – cardinal number;  $(x_i, y_i)$ ,  $\alpha_i$ ,  $t_i$ ,  $v_i$ ,  $\theta_i$ ,  $p_i$ ,  $h_i$  – coordinates, direction and type of minutiae, as well as value and direction of curvature, probability and density of lines about minutiae. We propose to detect minutiae detected only in the informative areas. On the fig. 1 FI informative area is darkened, the skeleton is black.

Coordinates  $(x_i, y_i)$  of minutiae  $M_i$  are determined by coordinates of skeleton node [8, 9]. Direction  $\alpha_i$  is determined with circuit of skeleton nodes for line endings and tree circuits for line bifurcations [10]. Type  $t_i \in \{0,1\}$  is determined with skeleton nodes valence like the nodes of graph [9], where 0 – bifurcation and 1 – ending. Coordinates  $(x_i, y_i)$ , direction  $\alpha_i$  and type  $t_i$  are the basic parameters  $M_i$  [7].

Value  $v_i$  and direction  $\theta_i$  of the curvature are determined by lines direction difference in the neighborhood  $\varepsilon$  of minutiae  $M_i$  [10]. Probability  $p_i$  is calculated as the ratio of the average value of image quality rating in the neighborhood  $\varepsilon$  to the best quality rating in the FI informative area [11]. Lines density  $h_i$  is calculated as the average quantity of lines, located into the neighborhood  $\varepsilon$  on the straight line traced transversely to lines [7]. A value of neighborhood  $\varepsilon$  is assigned to 3-5 line periods.

### 3.2. Topological vectors list

We now describe how topological vectors are used as effective representation of ridge lines. Topological vectors list for lines  $L_l$  is found on the basis of minutiae 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 and shown in fig. 2, which present data informational layers, distributed among its hierarchical structure [2, 7].



**Fig. 2.** Hierarchical structure of layers in pyramid

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

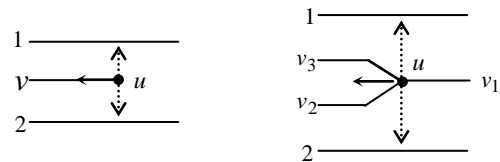
$$L_l = \{V_i = \{(e_j, n_j, l_j)\} | i \in 1..n_2, j \in 1..m_i\}, \quad (3)$$

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  $n_j$ ;  $m_i$  – quantity of links taking into account central line in the form of

$$m_i = 4m + 2. \quad (4)$$

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 (fig. 1). Minutiae directions (angle) point to the area of lines number increase [10]. 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 fig. 3, the projections are shown with dotted lines, and two corresponding nodes of skeleton on the lines 1 and 2 are painted over.



**Fig. 3.** 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 spiral the dissected lines (hereinafter 'links'), which turn clockwise. The section traces the lines curvature [11]. 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 formula (4).

Topological vector is determined by the section. To do this, we 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 projection from minutiae, located on an adjacent line to the right or to the left of the link. The following possible events are detected on the links, shown on the fig. 4 and represented in a binary code.

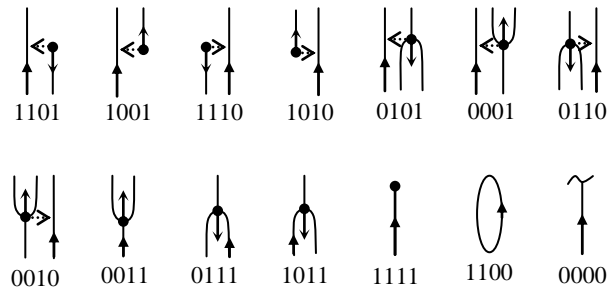


Fig. 4. Events

Minutiae' number initiating the event is associated with the event as the label (fourteen events on fig. 4) detected on the link. The event is associated with the link label. 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**. The event and minutiae' number form ordered pair  $(e_j, n_j)$ . The event is amplified with the link length corresponding to the distance from the section to the position in which the event is detected. **Enlarged topological vector** is formed as follows. 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.

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.

Topological vectors are built for every node of the skeleton  $p_i$  (except minutiae). The process divides the lines into the links, numbered on spiral, turning clockwise. On the fig. 5, examining section for the node A of a skeleton line, which is locked in ending 19, the links are enumerated as 0–17. Topological vector of the node A is shown in the table 1. On the fig. 6, 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 [11] of ending 19 into bifurcation 19 (which can result due to noise in the original images). 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  $m=4$  links  $m_l=18$  for the line are formed according to formula (4).

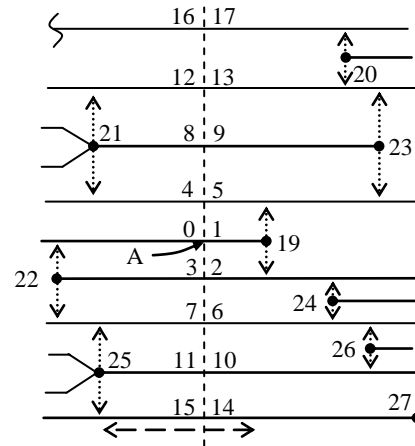


Fig. 5. Section for line with ending

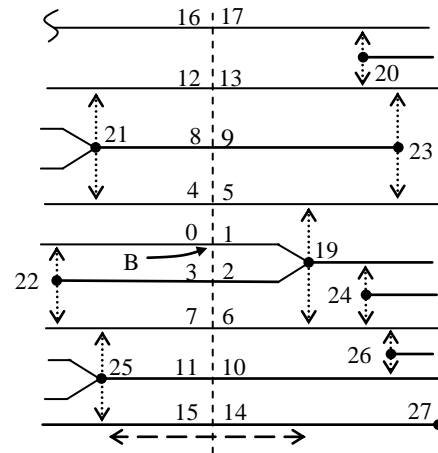


Fig. 6. Section for line with bifurcation

Table 1. Topological vector for A

№	Event	Index	Size
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

№	Event	Index	Size
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	$l_{17}$

The quantity of topological vectors can be enumerated. At the foot of the fig. 5 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 fig. 6. Topological vectors with equal **basic topological vectors** are integrated into one corresponding enlarged topological vector, where the minimum length of link is maximal [11]. At that their quantity is reduced by dozens of times from the value  $n_2 < 1000$  to the value  $n_1 < 100$  according to (3). Vector  $V_i$  automatically characterizes the line or the line segment, but not the minutiae. The image deformation, especially linear, practically does not have any effect upon the content of the basic topological vector. Therefore the vector is named as topological [10-12].

The proposed methodology has 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 [11]. This raises stability and comprehension of mathematical model.

### 3.3. Delaunay triangulation

After the topological vectors has been defined, we propose to utilize Delaunay triangulation for their matching. Delaunay triangulation is built from the minutia list  $L_m$  according to formula (2). By definition, Delaunay triangulation is a triangulation, where circle, circumscribed around any of it's triangle, doesn't contain inside any other point from  $L_m$ . For the node A of skeleton line (look the fig. 5) one of triangles is shown on the fig. 7. Vortexes of that triangle are the minutia  $\{U_i | U_i \in L_m, i \in 1..3\}$  and  $\{U_i\} = \{M_j | M_j \in L_m, j \in \{22, 21, 19\}\}$  accordingly.

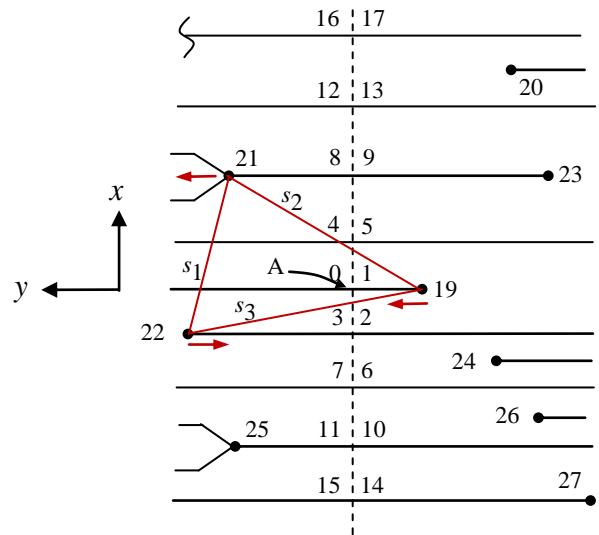
To choose the initial vortex of triangle let's note that node A placed inside the circle, circumscribed around the marked triangle. Topological vector for the point A divides an image with its section into two areas. To the area, containing the most quantity of vortexes of marked triangle, direct an ordinate axis of topological vector, and on the section – abscissa axis. Let's

choose the initial vortex of triangle  $U_s$  with plus ordinate and minimum abscises' meaning. On the fig. 7 minutia number 22 corresponds to vortex  $U_s$ .

Let's go clockwise round the vortexes of the triangle, beginning from the vortex  $U_s$ . In the sequence of minutia 22, 21, 19 let's compare the lengths of triangle sides  $s_1 > s_2, s_2 > s_3, s_3 > s_1$  and form three bits  $\gamma_1, \gamma_2, \gamma_3$ , where  $\gamma \in \{0,1\}$ . They are determined with binary results of comparison. It is possible in total 7 states of those bits: 110, 011, 101, 100, 010, 001, 000. The last corresponds to equilateral triangle.

According to the events in topological vector for the point A (table 1) minutiae number 22 is directed contrariwise to the link course. This fact allows us to show the meaning of bit  $\gamma_4 = 1$ . Therefore, during the clockwise round, over the vertices of triangle sequentially, we calculate  $\gamma_5 = 0$  (is directed along the link course) and  $\gamma_6 = 1$ . Since the minutia distribution is random, there are eight possible states of those three bits.

Node A placed on a line, beginning from minutia 19. If it is directed along ordinate axes of topological vector, let's set the bit  $\gamma_7 = 1$ , otherwise  $\gamma_7 = 0$ .



**Fig. 7.** Delaunay triangulation for A

Calculated bits  $C = \{\gamma_k | k \in 1..7\}$  assume 112 possible states ( $7 \times 8 \times 2$ ). These bits are not correlating. There density of distribution is inhomogeneous. So, we have 112 descriptors which classify topological vectors. Such preliminary classification of topological vectors and minutia on their "equivalence" stable to turns and displacements of the image and speed up fingerprint identification ten times. If  $n_2$  – cardinal number (3), accelerator vectors list is written in the form of

$$L_a = \{C_i = \{\gamma_k | k \in 1..7\} | i \in 1..n_2\}. \quad (5)$$

## 4. CONCLUSION

In this paper, we presented mathematical model of FI on the basis of topological vectors for the ridge lines, which is stored in the template. Topological vectors form linked graph with a high level of redundancy. This allows connecting sub graphs of fragmentary latents of fingerprints. List  $L_l$  according to (3) can be represented in economic format (without links lengths). Minutiae mutation does not 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.

Additionally, the lines in the areas of loops, deltas, whirls and essential curvatures are automatically divided with topological vectors independently from minutiae location, which increases stability of mathematical model.

Robust descriptions are proposed on the basis of Delaunay triangulation, which form 112 classes and allow speeding up fingerprint identification ten times. We tested fingerprints from optical sensor FVC2000 DB3, FVC2002 DB1, FVC2004 DB1, FVC2006 DB2 on the processor Intel Pentium D CPU 3.40GHz.

An image template according consists of lists which are mutually complementary, for them minutiae list is the determining one. The lists are essentially different and do not replace each other, but one of these lists can be excluded from this template. Moreover, topological vectors can be stored compactly (without links lengths). This allows to further optimize memory capacity for the template storage

## 5. REFERENCES

- [1] Bebis, G., Deaconu, T and Georiopoulous, M. Fingerprint identification using Delaunay triangulation, ICII99, Maryland, Nov, pp. 452-459, 1999.
- [2] Gonzales, R. Digital processing of the images / R. Gonzales, R. Woods; translation from English; the editor P. Chochia – M.: Techno sphere, 2006. – 1072 p.
- [3] Jain, A., Hong, L. and Bolle, R. On-line fingerprint verification, IEEE TPAMI, vol. 4, pp. 302–313, 1997.
- [4] Jiang, X., Yau, W.-Y., 2000. Fingerprint minutiae matching based on the local and global structures. In: Proc. 15th Internet. Conf. Pattern Recognition (ICPR, 2000) 2. pp.1042–1045, 2000.
- [5] Kovacs-Vajna, Zs. Miklos A Fingerprint Verification System Based on Triangular Matching and Dynamic Time Warping, IEEE Trans. on PAMI, Vol.22, No.11, pp.1266-1276, 2000.
- [6] Liang, X. and Asano, T. A Linear Time Algorithm for Binary Fingerprint Image Denoising Using Distance Transform, IEICE TRANSACTIONS on Information and Systems, vol. E89-D, no. 4, pp. 1534-1542, 2006.
- [7] Maltoni, D. Handbook of fingerprint recognition / D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar. – London: Springer-Verlag, 2009. – 496 p.
- [8] Mestetskiy, L.M. Continious morphology of binary images / L.M. Mestetskiy.– M.: Fizmatlit, 2009.– 288 p.– ISBN 978-5-9221-1050-1.
- [9] Novikov, F. Discrete Mathematics for programmers: manual / F. Novikov. – St. Petersburg: Piter, 2001. – 304 p.
- [10] 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.
- [11] Patent 236086 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.
- [12] Patent 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.

[13] Ratha, N.K., Karu, K., Chen, S. and Jain, A. A Real-Time Matching System for Large Fingerprint Databases, PAMI Vol.18, No.8, pp. 799-813, 1996.

[14] H. Wang, M. Gavrilova, Y. Luo and J. Rokne An Efficient Algorithm for Fingerprint Matching, International Conference on Pattern Recognition ICPR 2006, 20 - 24 August 2006, Hong Kong, IEEE-CS publisher, pp. 1034-1037, 2006

[15] Wayman, J, Jain, A, Maltoni, D. and Maio, D. “Biometric Systems: Technology, Design and Performance Evaluation,” Book, Springer-Verlag, 2006.

[16] Wang, C. and Gavrilova, M. Delaunay Triangulation Algorithm for Fingerprint Matching IEEE-CS proceedings, ISVD 2006, pp. 208-216, Banff, AB, Canada, July 2006.

[17] Xiao, Y. and Yan, H. Facial Feature Location with Delaunay Triangulation/Voronoi Diagram Calculation, Conferences in Research and Practice in Information Technology, 11. Feng, D. D., Jin, J., Eades, P. and Yan, H., Eds., ACS. 103-108, 2002.

## About the author

Vladimir Gudkov, professor at Chelyabinsk State University, Department of Applied Mathematics. His contact e-mail is [diana@sonda.ru](mailto:diana@sonda.ru).

Marina Gavrilova, PhD, University of Calgary, Canada. Her contact e-mail is [marina@cpsc.ucalgary.ca](mailto:marina@cpsc.ucalgary.ca).



# Edge Based Image Comparison and Noise Removal

Vladislav Shubnikov, Sergey Belyaev

Applied Math Department of St.Petersburg State Polytechnical University

## Abstract

PSNR (Peak Signal to Noise Ratio) metric is typically used to estimate difference between two given images. Recent time more reliable metric (SSIM – Structural Similarity) was introduced to detect image changes/noise/etc. This paper introduces more advanced approach for the image difference measurement based on weighted sum of the image structure estimation and edges detection. Proposed metric (ESSIM – Edgeted Structural Similarity) is more close to the human image difference perception; it provides strong emphasize on pixels near edges. The paper illustrates two image modification cases where PSNR and SSIM cannot find difference between images, but proposed metric can do it. Also this paper introduces a novel image noise removal approach based on the weighted mixture of bilateral method and advanced edge detection approach.

**Keyword:** Image Similarity; Image Processing; Edge Detection; Noise Removal; Image Quality Metrics;

## 1. PREVIOUS WORKS

Image noise removal techniques are widely used for many human activities: professional and amateur photography, aerospace photography, medical images processing, images classification, face detection, etc. To estimate quality of a method for noise removal, you need to do the following pipeline. On the first step to get from somewhere a source image with more or less good quality. Then RGB color information is converted into YCrCb or YUV or other color space, where color intensity information is concentrated in the one channel (not in 3 channels like in RGB color space). On the second step you should introduce some artificial noise into the image. On the third step the method under testing is applied to the “dirty” image, so denoised image is produced. On the fourth, last step, the denoised image is compared with the source image: the less difference, the better noise removal quality. It is very important to use good metrics for image similarity / difference estimation. In the [2] there is a good review of popular image quality metrics, started from simplest PSNR and continued to more complicated techniques. There are also very interesting approaches proposed, but not all of them are implemented. Implementation of one idea - using edge-based measurement - is described in this paper. The paper [4] gives a prove that SSIM measurement is more natural in comparison with PSNR.

Nowadays there are a lot of image removal techniques, and modern approaches try to receive excellent denoising quality for reasonable time. Most of simple convolution methods with static convolution matrix provide very fast image denoising, but unfortunately, add significant blurring on the edges. To keep edges structure is very important for human perception. Bilateral filtration [5] is more or less good solution for this task. We propose some improvements into this well known method to reach better denoising results.

## 2. IMAGE DIFFERENCE CALCULATION APPROACH

Using SSIM measurement (described in [4]) as a basis, we will

modify calculation a little bit and introduce the new coefficient, significantly affecting the result value. According to [2, 4] and many other sources, for given two images  $P_x$  and  $P_y$  SSIM calculation sequence is:

$$\mu_x = \frac{1}{W * H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} P_x(i, j) \quad \mu_y = \frac{1}{W * H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} P_y(i, j)$$

where  $W, H$  – image dimensions (should be the same for both compared images). Here  $\mu_x, \mu_y$  - so called mean values for images  $P_x$  and  $P_y$ . After that we can calculate value, which characterizes contrast for each image:

$$C_x = \frac{1}{W * H - 1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} [P_x(i, j) - \mu_x]^2$$

$$C_y = \frac{1}{W * H - 1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} [P_y(i, j) - \mu_y]^2$$

We can also calculate correlation between images, using the following formula:

$$R = \frac{1}{W * H - 1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} |P_x(i, j) - \mu_x| * |P_y(i, j) - \mu_y|$$

or each image we have 3 feature-related values, and can calculate three coefficients, characterizing luminance, contrast and structure (covariance) differences between images:

$$K_l = \frac{(2\mu_x\mu_y + C_1)}{(\mu_x\mu_x + \mu_y\mu_y + C_1)} \quad K_c = \frac{(2C_x C_y + C_2)}{(C_x C_x + C_y C_y + C_2)}$$

$$K_s = \frac{(R + C_3)}{(C_x C_y + C_3)}$$

where  $C_1, C_2, C_3$  – constants to avoid division by zero.

Resultant SSIM value will be multiplication of previous coefficients  $K_l, K_c$  and  $K_s$ . Below, in the section 4, we will show significant drawbacks of SSIM criterion.

Now let us imagine that for any input image we can build the image which describes edges, detected in the input. Values close to 0, describe smooth image areas; values close to 1, describe strong edges. Talking “edges” we assume significant intensity difference in neighbourhood pixels. There are a lot of possibilities how we can calculate edge map. Adding weights to the contrast and correlation calculation, we can introduce edge-dependant contrast and correlation:

$$C_x = \frac{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} W_x(i, j) [P_x(i, j) - \mu_x]^2}{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} W_x(i, j)}$$

$$C_y = \frac{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} W_x(i, j) [P_y(i, j) - \mu_y]^2}{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} W_x(i, j)}$$

where  $W_x$  – weight matrix with values in [0..1], describing edge feature for pixel  $P_x(i,j)$ . We will use only one weight matrix, created from the first image, to create dependence on edges. In this approach, the first image assumed as original and “clean” image, the second one is the result of denoising procedure. Edge-weighted correlation will be calculated as:

$$R_w = \frac{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} W_x(i, j) [P_x(i, j) - \mu_x] * [P_y(i, j) - \mu_y]}{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} W_x(i, j)}$$

Also, we introduce contrast correlation (D) and weighted contrast correlation ( $D_w$ ):

$$D = \frac{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} [(P_x(i, j) - \mu_x) - (P_y(i, j) - \mu_y)]^2}{W * H - 1}$$

$$D_w = \frac{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} W_x(i, j) [(P_x(i, j) - \mu_x) - (P_y(i, j) - \mu_y)]^2}{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} W_x(i, j)} \quad F$$

or the new introduced image characteristics we will calculate a special coefficient, showing relationship between D and  $D_w$ :

$$K_w = \frac{(D + C_4)}{(D + D_w + C_4)}$$

For the images which have noticeable difference in edge areas,  $K_w$  drops down to zero. Finally, we can calculate resultant ESSIM value:

$$ESSIM = K_l * K_c * K_s * K_w$$

ESSIM value has the same meaning, as SSIM: 0 means absolutely different images, 1 means the same images. The more ESSIM close to 1, the more similar images are.

### 3. EDGES CALCULATION

In the previous section we have introduced special matrix  $W_x$ , which describes edges existence for each original image pixel. For the sake of simplicity we can calculate  $W_x$  values, based on [7] with sequence of the following operations. First, we will smooth input image in order to roughly remove noise artefacts. For this purpose any

smoothing convolution can be applied. Then we calculate two gradient fields:  $G_x$  and  $G_y$ :

$$G_x(x, y) = \sum_{j=-1}^{+1} \sum_{i=-1}^{+1} P_x(x+i, y+j) * KG_x(1+i, 1+j)$$

$$G_y(x, y) = \sum_{j=-1}^{+1} \sum_{i=-1}^{+1} P_y(x+i, y+j) * KG_y(1+i, 1+j)$$

where  $KG_x$  and  $KG_y$  are simple kernel matrices used in convolution:

$$KG_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad KG_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$G_x$  and  $G_y$  values describe intensity gradients in both directions: horizontal and vertical.  $KG_x$  and  $KG_y$  are well known under the name of Sobel operator kernel. Final “edge” feature value for any image pixel (x,y) can be calculated as:

$$W_x(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2}$$

Fig. 1 and 2 show detected edges, calculated by this formula for different test images (original images are on the left):

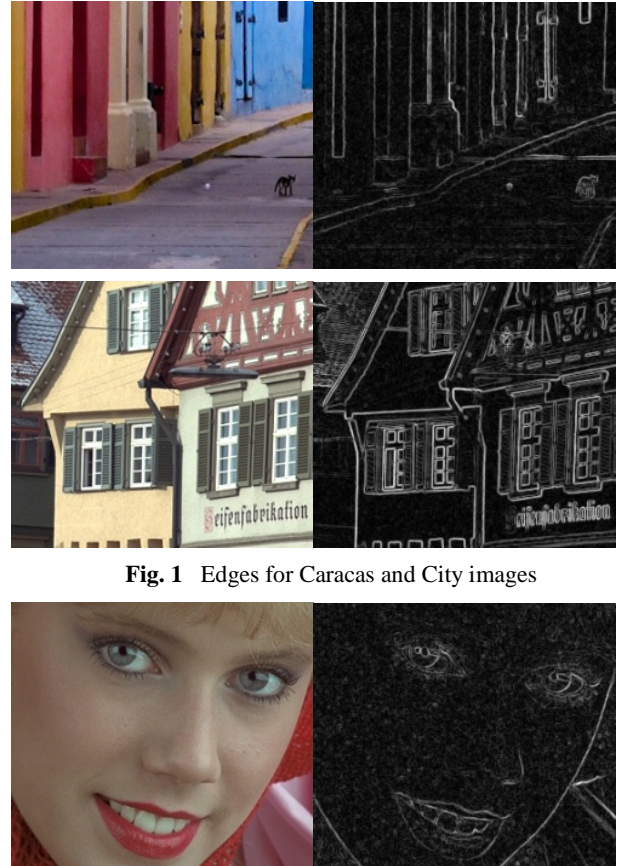


Fig. 1 Edges for Caracas and City images

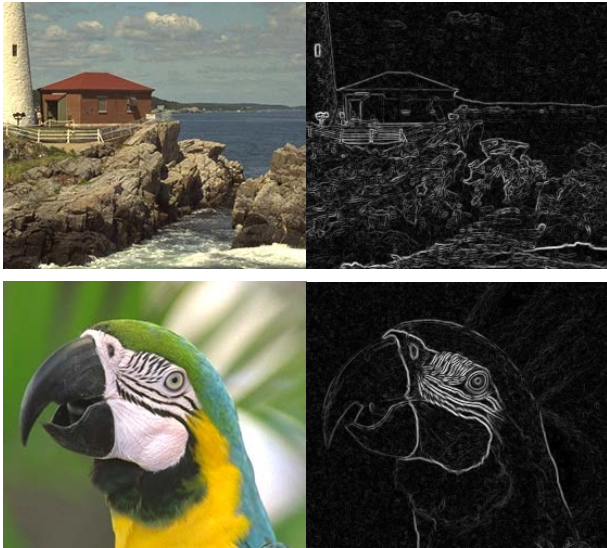


Fig. 2 Edges for Girl, Light House and Macaw images

#### 4. TEST RESULTS

Let us first test sensitivity of the proposed ESSIM metric as compared with well known metrics for a synthetic image. For example, for test image Fig. 3.

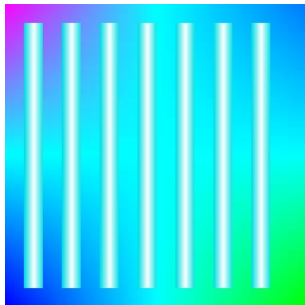


Fig. 3 Test sample image

We will insert artificial white Gaussian noise into this image in two different manners: the first approach will touch input image areas without visible edges; the second approach will change pixels around image edges, as shown on Fig. 4.

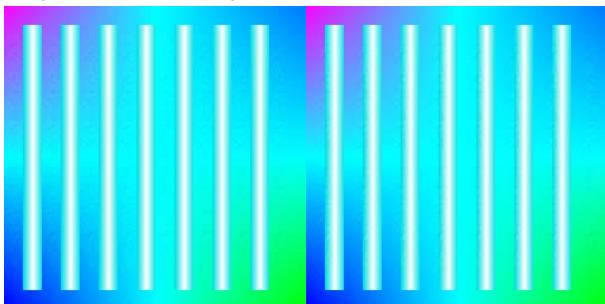


Figure 4: Very similar light modifications, applied to the different areas

In both cases corrupted area has the same size (in pixels), noisy pseudo-random fluctuations are the same too, and both modifications have equal noise range. Important difference between images is the areas, where pixels are modified. It is possible (but not very easy) to

notice that right image looks more “corrupted”. This impression can be received after edges observation: on the left image they are more “perfect” and human eye can “decide” that left image is of higher quality in comparison with right one. This effect is very close to the well-known Mach bands optical illusion, illustrating that human eye will see the same color in different ways, depending on the edge changes. Very good review of optical illusions can be found in [1] and [3]. Comparison between last two distorted images and original image gives not significant difference both for PSNR and SSIM measurements, but proposed ESSIM measure will detect noticeable difference, as shown in Table 1.

Table 1. Comparison of three image difference criteria for the synthetic image

Image modification	PSNR	SSIM	ESSIM
Noise in non-edge areas	29.53	0.9992	0.9682
Noise in edge areas	29.98	0.9990	0.5088

PSNR estimation detects the right image as more similar to the original than the left one. SSIM measurement shows that both modifications are the same and only ESSIM displays significant difference between two distortions according human perception: right one is more noticeable.

A set of natural test images is shown on Figs 5-7. They are results of various modifications of Girl image (Fig. 2). The differences between these images and source one calculated by three criteria



Fig. 5 Higher contrast image (left); negative image (right)



Fig. 6 Light Gaussian blur; lossy compression



Fig. 7 Light blurring; RGB components modification (R+10, G-5, B-5)

are shown in Table 2.

Table 2. Comparison of three image difference criteria for natural images

Image modification	PSNR	SSIM	ESSIM
Contrast	27.44	0.9767	0.3947
Negative	10.87	0.9454	0.6778
Added Gaussian noise	29.98	0.9892	0.5066
Lossy compression	28.66	0.9886	0.3530
Add light blur	26.43	0.9870	0.2489
RGB channels modification	100.0	0.9997	0.6703

In all cases ESSIM appears to be more sensitive to distortions than two well-known criteria. Of course, image blurring, contrast and loss of details are the strongest distortions affecting edges. We see corresponding large ESSIM differences for these modifications. Last image modification is most interesting: each pixel red color component was increased by 10 (for the color component range [0.255]), and green, blue components were decreased by 5. Both PSNR and SSIM detect no changes in image (but distortion is noticeable by eye), but ESSIM detects significant changes.

Yet one more interesting image comparison is received after RGB components modification (R=-20; G+=10; B+=10), applied to synthetic generated image Fig. 8.

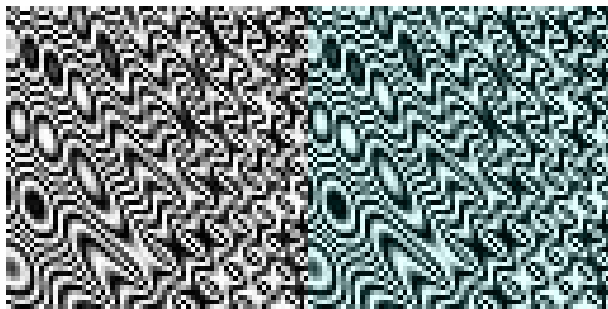


Fig. 8 Synthetic test image (left) and its light rgb components modification

Here SSIM value is equal to 0.9995 (which means that images are almost the same), but ESSIM is equal to 0.5730.

## 5. IMAGE NOISE REMOVAL METHOD

Bilateral filtering was introduced in [5] and uses more advanced approach as compared with simple Gaussian weighted non-linear filtration. Tricky data driven weight calculation is a famous way to produce new noise removal algorithms. Based on classical bilateral approach, a special combination of two different bilateral filters is introduced in [6]. Big disadvantage of this method is a hidden weight coefficient calculation based on the difference between medians for neighbouring pair of image pixels. Here we introduce a new approach for weights calculation. Let  $W$  and  $H$  be image dimensions. Let  $N$  – radius of the square shaping neighbourhood around each image pixel  $(x,y)$ . Neighbourhood area is required to take into account some pixels around  $(x,y)$ . The size of the neighbourhood area is equal to:

$$S = (2N+1)(2N+1)$$

Based on the non-linear filtering, we need to compute the new intensity of the pixel  $(x,y)$ :

$$P'(x, y) = \frac{\sum_{j=-N}^N \sum_{i=-N}^N w(i, j) P(x+i, y+j)}{\sum_{j=-N}^{N+1} \sum_{i=-N}^{N+1} w(i, j)}$$

Increasing value of  $N$  we make the calculation more “integrated”. Negative impact of increasing  $N$  is increased calculation time. Classical bilateral filter approach uses the following formula to calculate weights in pixel  $(x,y)$  neighbourhood:

$$w(i, j) = e^{-\frac{(i^2+j^2)}{2\sigma_s^2}} e^{-\frac{|P(x+i,y+j)-P(x,y)|^2}{2\sigma_r^2}}$$

Here  $\sigma_s$  and  $\sigma_r$  are so called spatial and radiometric constants, operator  $|P(x+i,y+j)-P(x,y)|$  is a squared difference between intensities of central pixel  $(x,y)$  and current pixel in its neighbourhood. Formula for  $w(i,j)$  calculation shows a simple principle for weight calculation: result weight is depending on two weights components multiplication – spatial and radiometric components. Pixel  $(i,j)$  in the neighbourhood of  $(x,y)$  will be weighted less with increasing distance between  $(x,y)$  and  $(i,j)$ . Pixel  $(i,j)$  will be weighted less if difference between intensity value of this pixel and central pixel will increase, radiometric coefficient will be smaller. Talking simpler, if we have very similar intensities in the pixel  $(x,y)$  neighbourhood, we will calculate average intensity for this neighbourhood. If intensity of the central pixel  $(x,y)$  differs much from the intensities in its neighbourhood, the weights of neighbourhood pixels will drop down to 0, so in this case value of the central pixel intensity will not be modified by calculation. In our approach we will go one step further: try to compare “edginess” of the central pixel  $(x,y)$  and pixel in its neighbourhood  $(i,j)$ . If the current pixel lies on the same edge area as the central one (the same if both are not on the edge) we will take into account its value and assign “large” weight to pixel  $(i,j)$ . If the current pixel has different “edginess” as compared with the central one, we will assign low weight in this case. So, for each pixel  $(i,j)$  in the neighbourhood of pixel  $(x,y)$  we will calculate spatial, radiometric and edginess coefficients:

$$C_s = i^2 + j^2; C_r = |P(x, y) - P(x+i, y+j)|^2$$

$$C_e = |W(x, y) - W(x+i, y+j)|$$

where  $W(x,y)$  – edges matrix, created by Sobel method (was described above in section III). Resultant weight can be calculated as:

$$w(i, j) = e^{-\frac{(C_s * C_r * C_e)}{\sigma}}$$

Here,  $\sigma$  is Gaussian constant, affected on “blurring” in resultant image. Important difference between proposed method and classic bilateral filtration is inside multiplication of components (not addition). This will cause stronger impact of a small change in a component on the resultant weight. For the practical implementation it is important to change two parameters of the proposed method: the size of the neighbourhood area (affecting whole integration) and  $\sigma$  value, affecting smoothness (larger values give more blurred result).

## 6. HIGH ISO NOISED IMAGES PROCESSING

For the synthetic noise added to the good quality images, the proposed edge bilateral method is not so impressive: it is not so strong noticeable visual difference between results of proposed

method and classic bilateral filtration. For natural test images the difference is more evident: the proposed method provides less blurring, as it is shown on Figs. 9 and 10.



**Fig. 9** Caracas noised image after bilateral filtration (left) and after proposed edge bilateral filtration (right)



**Fig. 10** Girl noised image after bilateral filtration (left) and after proposed edge bilateral filtration (right)

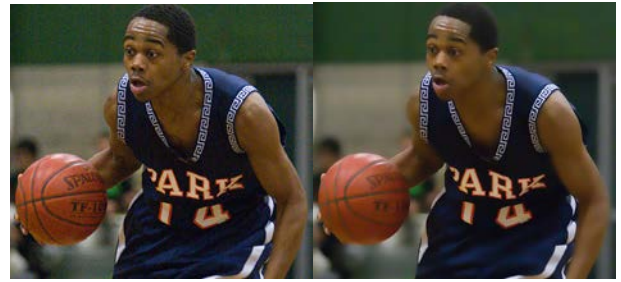
Using proposed approach we have applied developed filter to the set of real life high noised digital photos in order to remove noise. Each of Figs 11-13 show source noised picture (no artificial noise added, left) and result of applying proposed denoising filter (right).  $N$  is the radius of neighbourhood and  $\sigma$  is the parameter of weight calculation (see formulas in section 5).



**Fig. 11** Face high ISO digital image after proposed edge bilateral filtration ( $N=6$ ,  $\sigma=900$ )



**Fig. 12** Pier high ISO image after edge bilateral filtration ( $N=6$ ,  $\sigma=100, 800$ )



**Fig. 13** Nba high ISO image after edge bilateral filtration ( $N=6$ ,  $\sigma=800$ )

## 7. RESULTS

Table 3 shows higher edge-preservation quality of the proposed method.

Table 3. Comparison of three image difference criteria for two denoising methods

Test image name	Denoising method	PSNR	SSIM	ESSIM
Caracas	Bilateral	<b>26.44</b>	<b>0.9792</b>	0.3643
	Edge Bilateral	25.73	0.9735	<b>0.3896</b>
City	Bilateral	<b>26.21</b>	<b>0.9900</b>	0.4167
	Edge Bilateral	25.22	0.9874	<b>0.4508</b>
Macaw	Bilateral	27.54	<b>0.9932</b>	0.3344
	Edge Bilateral	<b>27.73</b>	0.9931	<b>0.3569</b>
Girl	Bilateral	27.18	0.9746	0.3625
	Edge Bilateral	<b>28.46</b>	<b>0.9754</b>	<b>0.3788</b>
Light House	Bilateral	<b>27.04</b>	<b>0.9859</b>	0.4078
	Edge Bilateral	26.58	0.9837	<b>0.4281</b>

Better results in each measurement method are shown with bold font. It is easy to see, that ESSIM metrics always correlate to the best denoising method.

## 8. CONCLUSION

Proposed new image difference metric, based on principles more close to human image vision. Illustrated several cases, where simple metrics can't detect image differences, but proposed metric can do it. Proposed image difference metrics (ESSIM) can be used for non-real time applications, like professional image removal tools/plugins, but it is hard to use it for real-time applications due to high calculation cost. Proposed improvement for bilateral filtration which can save image features on edges and prevent edge blurring.

## 9. IMAGE SOURCES

Real-life (noisy) images:

Face: Amateur photo, Canon digital camera, courtesy of Ivan Krylov

Pier: Flickr image database

Nba: Digital Photography Forum (<http://photography-on-the.net>)

**Special test images:**

Girl, Light House, City, Macaw: Kodak image database (<http://r0k.us/graphics/kodak/>). Caracas: Flickr image database

## 10. REFERENCES

- [1] E.H. Adelson, *Lightness Perception and Lightness Illusions*, In the New Cognitive Neurosciences, 2<sup>nd</sup> ed., MIT press, pp 339-351, 2000.
- [2] Y. A.Y. Al-Najar, D.C.Soong, *Comparison of Image Quality Assessment: PSNR, HVS, SSIM, UIQI*, International Journal of Scientific & Engineering Research, vol.3, Iss.8, 2008.
- [3] J.Andraos, *Named Optical Illusions*, Department of Chemistry, New York University, 2003-2011.
- [4] Peter Ndajah, Hisakazu Kikuchi, Masahiro Yukawa, Hidenori Watanabe, Shogo Muramatsu, *SSIM Image Quality Metrics for Denoised Image*, 2010.
- [5] C.Tomasi, R.Manduchi *Bilateral filtering for gray and color images*, in Proc. Int. Conf. Computer Vision, 1998, pp. 839–846.
- [6] G.Vijaya, V.Vasudevan, *A Novel Noise Reduction Method using Double Bilateral Filtering*, European Journal of Scientific Research, Vol.46 No.3 (2010), pp.331-338.
- [7] P.Zhou, W.Ye, Y.Xia, Q.Wang *An Improved Canny Algorithm for Edge Detection*, Journal of Computational Systems 7:5 (2011).

## About the author

Vladislav Shubnikov is assistant professor in Applied Math Department, St.Petersburg State Polytechnical University, Russia

[vlad.shubnikov@gmail.com](mailto:vlad.shubnikov@gmail.com)

Sergey Belyaev is professor in Applied Math Department, St.Petersburg State Polytechnical University, Russia

[sergey.belyaev@d-inter.ru](mailto:sergey.belyaev@d-inter.ru)

# Adaptive Feature Extraction for Gender Classification of Human Faces

Vladimir Khryashchev, Lev Shmaglit, Andrey Priorov, Andrey Shemyakov  
P.G. Demidov Yaroslavl State University, Yaroslavl, Russia  
dcslab@uniyar.ac.ru

## Abstract

An automatic gender recognition algorithm based on machine learning methods is proposed. It consists of two stages: adaptive feature extraction and support vector machine classification. Both training technique of the proposed algorithm and experimental results acquired on a large image dataset are presented.

**Keywords:** *image recognition, face detection, gender classification, machine learning, adaptive feature generation, support vector machine.*

## 1. INTRODUCTION

This research is devoted to the development of an automatic system capable to distinguish people's gender by analyzing their faces on digital images. Such systems can find application in different fields, such as robotics, human-computer interaction, demographic data collection, video surveillance, online audience measurement for digital signage networks and many others. In addition, gender recognition can be applied as a preprocessing step for face recognition. Machine learning techniques used in gender classification have universal character which allows to apply the solutions and knowledge in gender recognition to any other image understanding or object classification task.

There are comprehensive surveys written on face detection [5], face recognition [13] and facial expression analysis [1]. Gender recognition has been studied less. The comparative analysis of lately proposed gender classification algorithms has been presented in paper [7]. The researches have proposed algorithms based on artificial neural networks [11], on a combination of Gabor wavelets and principal component analysis [6], on independent component analysis and linear discriminant analysis (LDA) [4]. In paper [10] the authors utilize genetic algorithm for feature selection and support vector machine (SVM) for classification. In paper [9] an algorithm based on local binary patterns in combination with AdaBoost classifier was proposed. Experiments with radial basis function (RBF) networks and inductive decision trees are described in paper [12].

A new gender recognition algorithm, proposed in this paper, is based on non-linear SVM classifier with RBF kernel. To extract information from image fragment and to move to a lower dimension feature space we propose an adaptive feature generation algorithm which is trained by means of optimization procedure according to LDA principle. In order to construct a fully automatic face analysis system, gender recognition is used in connection with AdaBoost face detection classifier which selects candidates for analysis [3]. Detected fragments are preprocessed to align their luminance characteristics and to transform them to uniform scale.

The rest of the paper is organized as follows. The scheme of the proposed algorithm is described in section 2. Section 3 considers training methodology and experimental setup. In section 4, the

results of the proposed algorithm comparison with state-of-the-art gender classification methods is presented. Section 5 concludes the paper.

## 2. THE PROPOSED ALGORITHM

Classifier is based on Adaptive Features and SVM (AF-SVM). Its operation includes several stages, as shown in fig. 1.

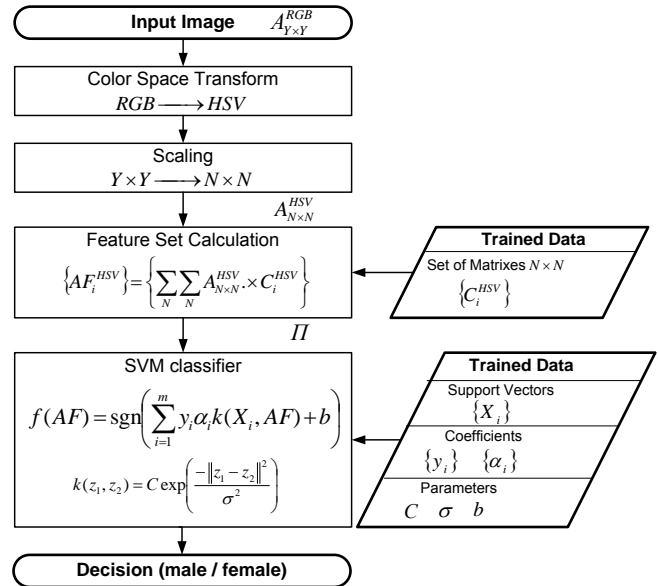
AF-SVM algorithm consists of the following steps: color space transform, image scaling, adaptive feature set calculation and SVM classification with preliminary kernel transformation. Input image  $A_{Y \times Y}^{RGB}$  is converted from RGB to HSV color space and is scaled to fixed image resolution  $N \times N$ . After that we calculate a set of features  $\{AF_i^{HSV}\}$ , where each feature represents the sum of all rows and columns of element-by-element matrix product of an input image and a coefficient matrix  $C_i^{HSV}$  with resolution  $N \times N$ , which is generated by the training procedure:

$$AF_i^{HSV} = \sum_N \sum_N A_{N \times N}^{HSV} \times C_i^{HSV}.$$

The obtained feature vector is transformed using a Gaussian radial basis function kernel:

$$k(z_1, z_2) = C \exp\left(\frac{-\|z_1 - z_2\|^2}{\sigma^2}\right).$$

Kernel function parameters  $C$  and  $\sigma$  are defined during training. The resulted feature vector serves as an input to linear SVM



**Fig. 1:** The scheme of the proposed gender classification algorithm.

classifier which decision rule is:

$$f(AF) = \text{sgn} \left( \sum_{i=1}^m y_i \alpha_i k(X_i, AF) + b \right).$$

The set of support vectors  $\{X_i\}$ , the sets of coefficients  $\{y_i\}$ ,  $\{\alpha_i\}$  and the bias  $b$  are obtained at the stage of classifier training.

### 3. TRAINING AND TESTING SETUP

Both gender recognition algorithm training and testing require huge enough color image database. The most commonly used image database for the tasks of human faces recognition is the FERET database [8], but it contains insufficient number of faces of different individuals, that's why we collected our own image database, gathered from different sources (Table 1).

Faces on the images from the proposed database were detected automatically by AdaBoost face detection algorithm. After that false detections were manually removed, and the resulted dataset consisting 10 500 image fragments (5 250 for each class) was obtained. This dataset was split into three independent image sets: training, validation and testing. Training set was utilized for feature generation and SVM classifier construction. Validation set was required in order to avoid the effect of overtraining during the selection of optimal parameters for the kernel function. Performance evaluation of the trained classifier was carried out with the use of the testing set.

The training procedure of the proposed AF-SVM classifier can be split into two independent parts: feature generation, SVM construction and optimization. Let's consider the feature generation procedure. It consists of the following basic steps:

- RGB  $\rightarrow$  HSV color space transform of the training images (all further operations are carried out for each color component independently);
- scaling training images to fixed image resolution  $N \times N$ ;
- coefficient matrix  $C_i^{HSV}$  random generation;
- feature value  $AF_i^{HSV}$  calculation for each training fragment;
- the utility function  $F$  calculation as a square of a difference between feature averages, calculated for "male" and "female" training image datasets, divided by the sum of feature variances [4]:

$$F = \frac{\left( \left\langle \{AF_i^{HSV}\}_M \right\rangle - \left\langle \{AF_i^{HSV}\}_F \right\rangle \right)^2}{\sigma \{AF_i^{HSV}\}_M + \sigma \{AF_i^{HSV}\}_F};$$

TABLE I

THE PROPOSED TRAINING AND TESTING IMAGE DATABASE PARAMETERS	
Parameter	Value
The total number of images	8 654
The number of male faces	5 250
The number of female faces	5 250
Minimum image resolution	640×480
Color space format	RGB
Face position	Frontal
People's age	From 18 to 65 years old
Race	Caucasian
Lighting conditions, background and facial expression	No restrictions

- iteratively in a cycle (until the number of iterations exceeds some preliminary fixed maximum value): random generation of coefficient matrix  $\tilde{C}_i^{HSV}$  inside the fixed neighborhood of matrix  $C_i^{HSV}$ , feature value  $A\tilde{F}_i^{HSV}$  calculation for each training fragment, calculation of the utility function  $\tilde{F}$ , transition to a new point ( $F \rightarrow \tilde{F}$ ,  $C \rightarrow \tilde{C}$ ), if  $\tilde{F} > F$ ;
- saving the matrix  $C_i^{HSV}$  after exceeding the maximum number of iterations;
- return to beginning in order to start the generation of the next (i+1) feature.

An optimization procedure, described above, allows to extract from an image only the information which is necessary for class separation. Besides, features with higher utility function value have higher separation ability. The feature generation procedure have the following adjusted parameters: training fragments resolution (N), the number of training images for each class (M), maximum number of iterations (T). The following values, as a compromise between reached separation ability and the training speed, were empirically received:

$$N = 65; \quad M = 400; \quad T = 10^5.$$

1000 features have been generated for each color component. At the second stage of training these features have been extracted from training images and were then used to learn an SVM classifier. SVM construction and optimization procedure included the following steps:

- calculation of the feature set, generated on the first stage of training, for each training fragment;
- feature normalization;
- learning an SVM classifier with different parameters of the kernel function;
- recognition rate (RR) calculation using validation image dataset;
- determination of optimal kernel function parameters (maximizing RR);
- learning a final SVM classifier with the found optimal kernel function parameters.

The goal of SVM optimization procedure is to find a solution with the best generalization ability, and thus with the minimum classification error. The adjusted parameters are: the number of features in a feature vector (N2), the number of training images for each class (M2), the kernel function parameters  $\sigma$  and  $C$ .

Grid search was applied to determine optimal kernel parameters: SVM classifier was constructed varying  $C=10^{k1}$  and  $\sigma=10^{k2}$ , where  $k1$  and  $k2$  – all combinations of integers from the range [-15 ... 15]; during the search recognition rate was measured using validation image dataset. Maximum recognition rate (about 80%) was obtained for  $C=10^6$  and  $\sigma=10^8$ .

Besides, we investigated the dependence of classifier performance from the number of features extracted from each color component – N2, and from the number of training images for each class – M2.

The analysis shows that each feature has essential separation ability, and at  $N2 = 30$  recognition rate reaches 79.5%. At the same time the growth of RR is observed both with the growth of



N2 and M2 due to the accumulation of information about considered classes inside the classifier. Thus, to obtain a compromise between quality and speed the following parameters were chosen:  $N2 = 30$ ;  $M2 = 400$ .

#### 4. EXPERIMENTAL RESULTS

In this section we present the results of the proposed AF-SVM algorithm comparison with state-of-the-art classifiers: SVM and KDDA (Kernel Direct Discriminant Analysis) [2].

Classifier AF-SVM was trained according to a technique, given above. SVM and KDDA classifiers have far less adjustable parameters as they are working directly with image pixel values instead of feature vectors. To construct these classifiers the same training base, as for AF-SVM classifier, was used. The following conditions also were identical for all three considered classifiers: the number of training images for each class, training fragments resolution and image preprocessing procedure. Optimization of SVM and KDDA kernel function parameters was held using the same technique and the same validation image dataset as used in case of AF-SVM classifier. Thus, equal conditions for independent comparison of considered classification algorithms, using testing image dataset, were provided.

For the representation of classification results we utilized the Receiver Operator Characteristic (ROC-curve). As there are two classes, one of them is considered to be a positive decision and the other – a negative. ROC-curve is created by plotting the fraction of true positives out of the positives (TPR = true positive rate) vs. the fraction of false positives out of the negatives (FPR = false positive rate), at various discrimination threshold settings. The advantage of ROC-curve representation lies in its invariance to the relation between the first and the second error type's costs.

The results of AF-SVM, SVM and KDDA testing are presented in fig. 2 and in table 2. The computations were held on a personal computer with the following configuration: operating system – Microsoft Windows 7; CPU type – Intel Core i7 (2 GHz) 4 cores; memory size – 6 Gb.

The analysis of testing results show that AF-SVM is the most effective algorithm considering both recognition rate and

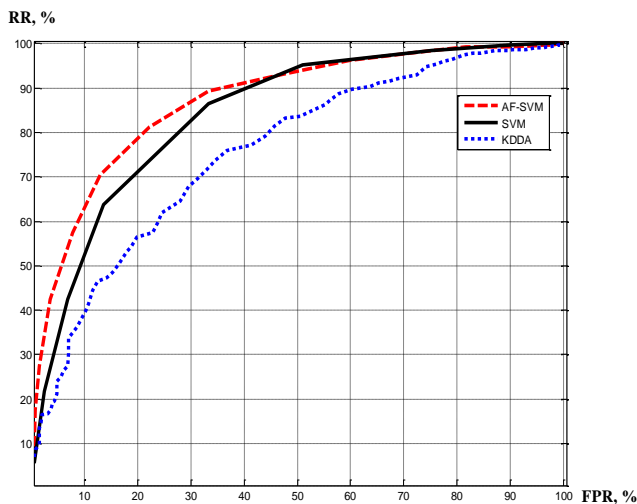


Fig. 2: ROC-curves of tested gender recognition algorithms.

TABLE II  
COMPARATIVE ANALYSIS OF TESTED ALGORITHMS PERFORMANCE

Algorithm \ Parameter	SVM		KDDA		AF-SVM	
	True	False	True	False	True	False
Recognition rate						
Classified as "male", %	80	20	75.8	24.2	80	20
Classified as "female", %	75.5	24.5	65.5	34.5	79.3	20.7
Total classification rate, %	<b>77.7</b>	22.3	<b>69.7</b>	30.3	<b>79.6</b>	20.4
Operation speed, faces/sec	44		45		65	

operational complexity. AF-SVM has the highest RR among all tested classifiers – 79.6% and is faster than SVM and KDDA approximately by 50%.

Such advantage is explained by the fact that AF-SVM algorithm utilize a small number of adaptive features, each of which carries a lot of information and is capable to separate classes, while SVM and KDDA classifiers work directly with a huge matrix of image pixel values.

Let's consider the possibility of classifier performance improvement by the increase of the total number of training images per class from 400 to 5000. Experiments showed that SVM and KDDA recognition rates can't be significantly improved in that case. Besides, their computational complexity increases dramatically with the growth of the training dataset. This is explained by the fact that while the number of pixels, which SVM and KDDA classifiers utilize to find an optimal solution in a high dimensional space, increases it becomes harder and even impossible to find an acceptable solution for the reasonable time.

In the case of AF-SVM classifier the problem of the decrease of SVM classifier efficiency with the growth of the training database can be solved by the use of the small number of adaptive features, holding information about a lot of training images at once. For this purpose we suggest that each feature should be trained using a random subset (containing 400 training images per class) from the whole training database (containing 5000 images per class). Thus each generated feature will hold the maximum possible amount of information, required to divide the classes, and a set of features will include the information from each of the 10 000 training images.

On the stage of feature generation 300 features were trained according to the technique described above. After that an SVM classifier, utilizing these features, was trained similarly as before. Besides, we preserved the number of training images for SVM construction equal to 400, and thus the operation speed of the final classifier remained the same as in previous experiments – 65 faces processed per second.

The results of AF-SVM algorithm trained using expand dataset ( $M = 5000$ ) and the initial AF-SVM classifier ( $M = 400$ ) comparison are presented in table 3 and in fig. 3.

The results show that AF-SVM algorithm together with the proposed training setup allow to significantly improve the

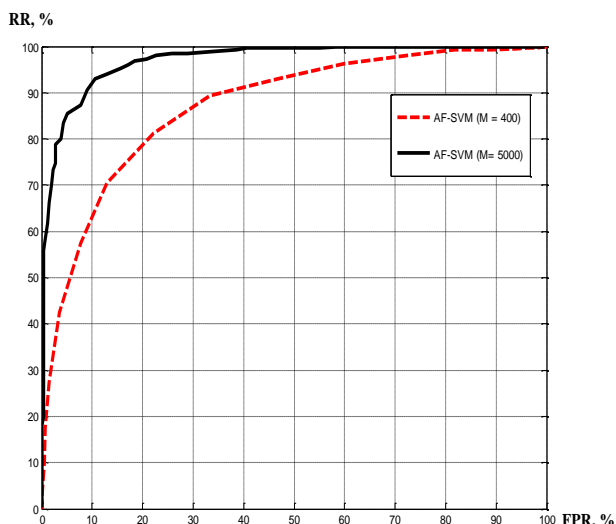


Fig. 3: ROC-curves for AF-SVM algorithm trained on datasets of different size.

TABLE III  
RECOGNITION RATE OF AF-SVM ALGORITHM TRAINED ON DATASETS OF DIFFERENT SIZE

Algorithm Parameter	AF-SVM (M=5000)		AF-SVM (M=400)	
	True	False	True	False
Classified as "male", %	90.6	9.4	80	20
Classified as "female", %	91	9	79.3	20.7
Total classification rate, %	<b>90.8</b>	9.2	<b>79.6</b>	20.4

classifier performance in case of increasing the training database size to 5000 images per class. Recognition rate of nearly 91% is achieved. It should be also noted that the adaptive nature of the feature generation procedure allows using the proposed AF-SVM classifier for the recognition of any other object on an image (in addition to faces).

### 5. CONCLUSION

A new classifier based on adaptive features and support vector machines, solving the problem of automatic gender recognition via face area analysis, is proposed. It shows recognition rate of 79.6%, which is 1.9% and 9.9% higher than that of SVM and KDDA correspondingly. The possibility of AF-SVM recognition rate improvement up to 91% in case of increasing the training database size to 5000 images per class is shown. The proposed algorithm allows to process 65 faces per second which is enough to use it in real time video sequence analysis applications.

### 6. REFERENCES

[1] Fasel B., Luettin J., "Automatic facial expression analysis: A survey," *Pattern Recognition Letters*, vol. 36, No 1, pp. 259-275, 2003.

[2] Gao H., Davis J., "Why direct LDA is not equivalent to LDA," *Pattern Recognition Letters*, vol. 39, No 5, pp. 1002-1006, 2006.

[3] Gutta S., Wechsler H., Phillips P.J., "Gender and ethnic classification of face images," *Proc. Int. Conf. on Automatic Face and Gesture Recognition*, pp. 194-199, 1998.

[4] Jain A., Huang J., "Integrating independent components and linear discriminant analysis for gender classification," *Proc. Int. Conf. on Automatic Face and Gesture Recognition*, pp. 159-163, 2004.

[5] Kriegman D., Yang M.H., Ahuja N., "Detecting faces in images: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, No 1, pp. 34-58, 2002.

[6] Lyons M., Budynek J., Plante A., Akamatsu S., "Classifying facial attributes using a 2-d Gabor wavelet representation and discriminant analysis," *Proc. Int. Conf. on Automatic Face and Gesture Recognition*, pp. 202-207, 2000.

[7] Makinen E., Raisamo R., "An experimental comparison of gender classification methods," *Pattern Recognition Letters*, vol. 29, No 10, pp. 1544-1556, 2008.

[8] Phillips P.J. et al, "The FERET evaluation methodology for face recognition algorithms," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, No 10, pp. 1090-1104, 2000.

[9] Sun N. et al, "Gender classification based on boosting local binary pattern," *Proc. Int. Symposium on Neural Networks*, vol. 2, pp. 194-201, 2006.

[10] Sun Z., Bebis G., Yuan X., Louis S.J., "Genetic feature subset selection for gender classification: A comparison study," *Proc. IEEE Workshop on Applications of Computer Vision*, pp. 165-170, 2002.

[11] Tamura S., Kawai H., Mitsumoto H., "Male/female identification from 8 to 6 very low resolution face images by neural network," *Pattern Recognition Letters*, vol. 29, No 2, pp. 331-335, 1996.

[12] Viola P., Jones M., "Rapid object detection using a boosted cascade of simple features," *Proc. Int. Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 511-518, 2001.

[13] Zhao W., Chellappa R., Phillips P., Rosenfeld A., "Face recognition: A literature survey," *ACM Computing Surveys (CSUR)*, vol. 35, No 4, pp. 399-458, 2003.

### About the author

Vladimir Khryashchev is with the Image Processing Laboratory, P.G. Demidov Yaroslavl State University. His contact email is [vhr@yandex.ru](mailto:vhr@yandex.ru).

Lev Shmaglit is with the Image Processing Laboratory, P.G. Demidov Yaroslavl State University. His contact email is [lev\\_shmaglit@yahoo.com](mailto:lev_shmaglit@yahoo.com).

Andrey Priorov is with the Image Processing Laboratory, P.G. Demidov Yaroslavl State University. His contact email is [andcat@yandex.ru](mailto:andcat@yandex.ru).

Andrey Shemyakov is with the Image Processing Laboratory, P.G. Demidov Yaroslavl State University. His contact email is [andrey.shemiakov@gmail.com](mailto:andrey.shemiakov@gmail.com).

# Multidimensional filtering in application to progressive video rendering

Gruzdev A., Frolov V., Vostryakov K., Ignatenko A.

Lomonosov Moscow State University, Computational Mathematics and Cybernetics Department, Moscow, Russia  
aleks-gruzdev92@yandex.ru, vfrolov@graphics.cs.msu.ru, kvostryakov@nvidia.com, ignatenko@graphics.cs.msu.ru

## Abstract

In this paper we propose a new approach to progressive video rendering. We apply a multidimensional filtering to samples with additional information about a scene along with color. Our filter takes noisy output of path tracing with a low amount of paths per pixel and yields the resulting movie quicker than tracing a large amount of rays. Our approach adds only a linear member  $O(N)$  to a path tracing computing complexity (where  $N$  is amount of pixels). With the help of progressive scheme the resulting video quality is improving after the every iteration. Using our approach we can get a video of a good quality even computing 10 paths per pixel.

**Keywords:** *path tracing, multidimensional filtering, progressive rendering, video rendering, video filtering.*

## 1. INTRODUCTION

Photorealistic rendering is one of the main tasks of computer graphics. A properly high quality and physical correctness are provided by the Monte Carlo ray tracing algorithm [Ritschel et al 2011] (and its modifications: backward ray tracing, path tracing), but it has a high computing complexity that complicates application of ray tracing for real-time rendering. Often it takes minutes to render every frame of a movie even for a simple scene. When rendering process has finished and a user estimates the result he may want to change it by tweaking scene parameters, so the long rendering process should be performed again.

A small amount of rays can be used for faster frames rendering, but it leads to highly noised results, so a filtration is necessary in this case. But many fast denoising methods can't safe small details and sharp edges. On another hand, many high quality algorithms have a high complexity and can't work in real-time.

Aiming at achieving a high speed we apply a fast filtrating algorithm to noisy ray tracing results. It removes the most part of noise saving details and edges. The rest of noise is expected to be removed with the help of progressive rendering scheme after a few iterations.

## 2. RELATED WORKS

### 2.1. Bilateral filtering

**Bilateral filtering** [Tomasi and Manduchi 1998] smoothes images while preserving edges, by means of a non-linear combination of nearby image values. The method is non-iterative, local, and simple. It combines gray levels or colors based on both their geometric closeness and their photometric similarity, and prefers near values to distant values in both domain and range. In contrast with filters that operate on the three bands of a color image separately, a bilateral filter can smooth colors and preserve edges in a way that is tuned to human perception. Also, in contrast with standard filtering, bilateral filtering produces no phantom colors along edges in color images, and reduces phantom colors

where they appear in the original image. The main disadvantage of the bilateral filtering is its computing complexity which doesn't allow to apply the classic algorithm for real-time filtering.

Bilateral filtering is computationally expensive due to the adaptive kernel recomputation at every pixel. [Pham and van Vliet 2005] present a **separable implementation of the bilateral filter**. Separable implementation of a multi-dimensional bilateral filter offers equivalent adaptive filtering capability at a fraction of execution time compared to the traditional filter.

In [Yang et al 2009] a new **real-time bilateral filtering** algorithm with computational complexity invariant to filter kernel size is proposed. Also, the algorithm lends itself to a parallel implementation. The method gives the same output accuracy and can be about 10x faster on average than the state-of-the-art.

### 2.2. Non-local denoising

Essentially different method of a **non-local image denoising** is proposed by [Buades et al 2005]. This algorithm computes for every pixel a weighted average of all pixels in the image. It compares spatial neighborhoods of pixels and gives large weights to the similar ones and small weights to others, so gathering information from the whole image, though that leads to a nonlinear complexity of the algorithm. In comparison to bilateral filtering [Tomasi and Manduchi 1998] the non-local approach shown better results of denoising, giving the best results on periodic images. In [Seo and Milanfar 2008] the non-local denoising is implemented for 3D filtration for video through enlargement neighborhoods along time axis.

### 2.3. Time-coherence video filtering

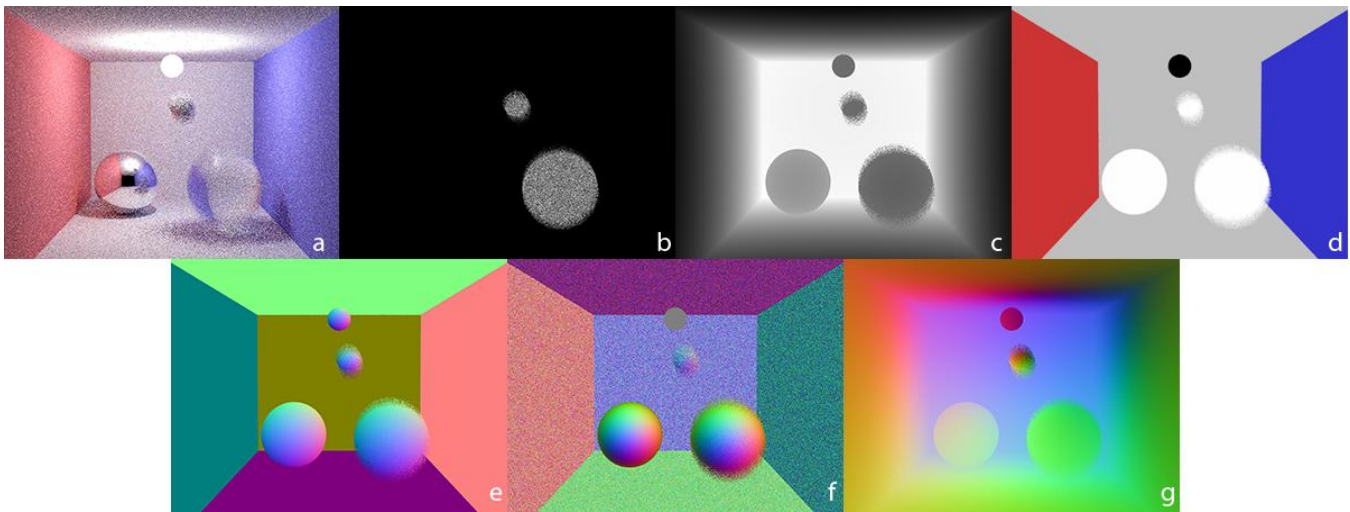
A modification of the non-local image denoising is presented by [Liu and Freeman 2010]. An integrating of optical flow into the method is a key to ensure temporal coherence in video filtering, and an approximate K-nearest neighbor matching reduces the high complexity of the classical algorithm.

[Bartovcak and Vrankic 2012] present an **adaptive pixel-wise algorithm** based on temporal averaging. Processing blocks of pixels requires lots of resources, and this approach observes a video as a group of 1D signals – one time depended signal per each pixel. The proposed method is simple and intuitive, has a lower computing complexity than some other algorithms. Giving comparable to other methods results, nevertheless it has a weakness of edges processing.

[Tawara et al 2004] propose to extend traditional photon density estimation methods for global illumination computation by using spatio-temporal bilateral filtering to reduce stochastic noise, while preventing excessive blurring in reconstructed lighting. This method is suitable for practical animation systems, where the rendering speed is the key factor even at the expense of lower accuracy in the lighting computation.

### 2.4. Multidimensional filtering

[Gastal and Oliveira 2012] present a new approach to efficiently performing high-quality **high-dimensional filtering**. The method



**Figure 1.** Components of the sample vector: a) color; b) time; c) depth; d) material color; e) normal direction; f) reflected ray direction; g) shadow ray direction

accelerates filtering by evaluating the filter's response on a reduced set of sampling points and using these values to interpolate the filter's response at all input pixels. The resulting filter is quite flexible, being capable of producing responses that approximate either standard bilateral filter [Tomasi and Manduchi 1998]. The presented filter can be implemented for a large number of dimensions, so it can be applied for video filtering or filtering with additional information. For a proper choice of the sampling points the total cost of the filtering operation is linear both in a number of pixels and in a number of dimensions.

[Sen and Darabi 2012] propose **random parameter filtering** for noisy results of Monte Carlo ray tracing with a low amount of samples per pixel. The method considers a sample as a high-dimensional vector of scene parameters, that allows using more information about a pixel than its color, and computes functional relationships between sample values and random parameter inputs. Then the approach uses all information to compute weights of every pixel when applying a cross-bilateral filter [Petschnigg et al 2004], which removes only the noise caused by the random parameters but preserves important scene detail.

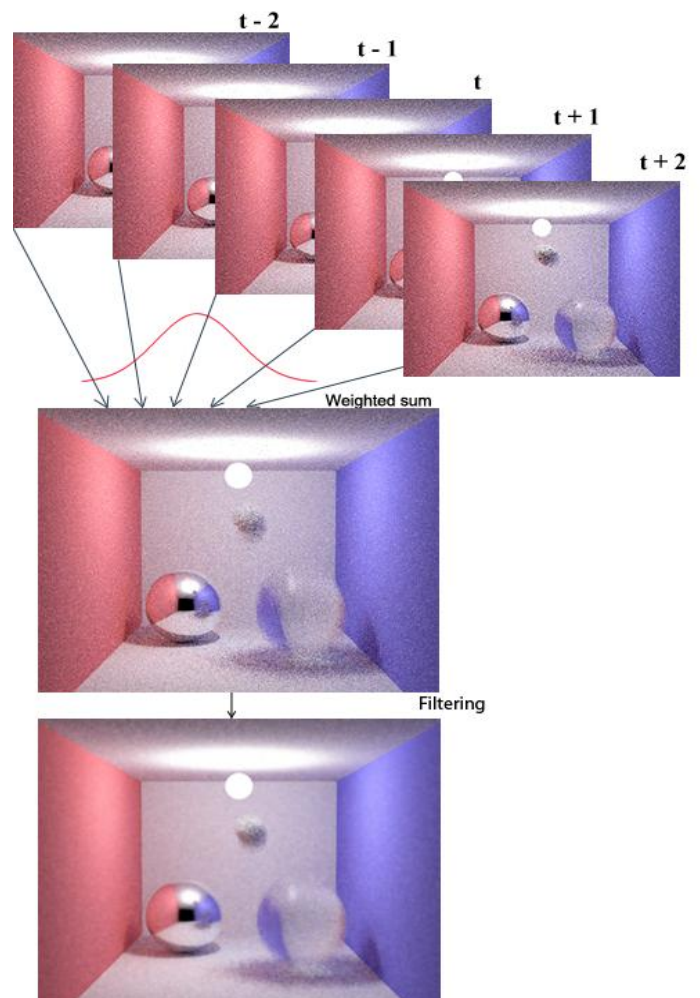
## 2.5. Filtering for progressive rendering

[Schwenk et al. 2012] presents an approach for filtering in progressive Monte Carlo rendering. This method performs a strong denoising with saving sharp edges and is able to display the first resulted image after ray tracing a few paths per pixel. Using light path classification high-variance and low-variance noise is separated into different buffers; the bilateral filtering is applied only to high-variance noise. High complexity of bilateral filtering is compensated by not each frame denoising and accumulating new samples during filtration process.

## 2.6. Summary

The reviewed methods can be used for filtering video, though all of them have disadvantages for our task. The non-local approaches ([Buades et al 2005], [Tomasi and Manduchi 1998], [Seo and Milanfar 2008], [Liu and Freeman 2010]) have very high computing complexity, so they don't fit progressive video rendering well. The pixel-wise algorithm [Bartovcak and Vrankic 2012] has a weakness saving noise at edges, while the bilateral filtering process edges better. The bilateral filters ([Tomasi and

Manduchi 1998], [Yang et al 2009], [Pham and van Vliet 2005]) show good results in image and video denoising, and can be applied for filtering results of 3D rendering ([Tawara et al 2004]), though they can't be applied for high-dimensional data without



**Figure 2.** Steps of the algorithm. Computing a weighted sum and filtering.

modifications. The filtering for progressive Monte Carlo rendering [Schwenk et al. 2012] is not suitable for our task due to bilateral filtering disadvantages and an absence of ability to accumulate new samples during filtration process. Both [Gastal and Oliveira 2012] and [Sen and Darabi 2012] approaches can work with additional information and be applied for a large number of dimensions, though the first one is faster and implement a more general approach. So we use the method of high-dimensional filtering. In contrast to [Gastal and Oliveira 2012] we apply it to filtering of videos saving time coherence but for single images.

### 3. SUGGESTED APPROACH

In our implementation frames are rendered by Monte Carlo path tracer with a low amount of paths per pixel (1-8 paths). The algorithm takes at the input a sequence of frames with additional information about a scene: time  $t$  of a frame for filtering, parameter  $\sigma$  of blurring strength, a size of a window. We define the window as a sequence of rendered frames in a definite time range, including time  $t$  as well. (Fig. 2)

Information from all frames in the window is used for filtering the current frame. We use a Gaussian weighted sum while an accumulating samples from the frames. Also we can produce a new frame if the time  $t$  is set between existing frames. In future work we are going to avoid using a separate filter pass for time and use filtering capability of the multidimensional filter [Gastal and Oliveira 2012].

Each sample is considering as a high-dimensional vector:  $S = (R, G, B, T, Z, Mr, Mg, Mb, Nx, Ny, Nz, Dx, Dy, Dz, Sx, Sy, Sz)$ , where  $R, G, B$  – computed color;  $T$  – time;  $Z$  – ray depth value;  $Mr, Mg, Mb$  – material color;  $Nx, Ny, Nz$  – direction of a normal to the first ray hit point;  $Dx, Dy, Dz$  – direction of the reflected ray after the first bounce;  $Sx, Sy, Sz$  – direction of the shadow ray. (Fig. 1) All additional data can be got from the classical path tracer and don't require any considerable modification of the path tracing algorithm.

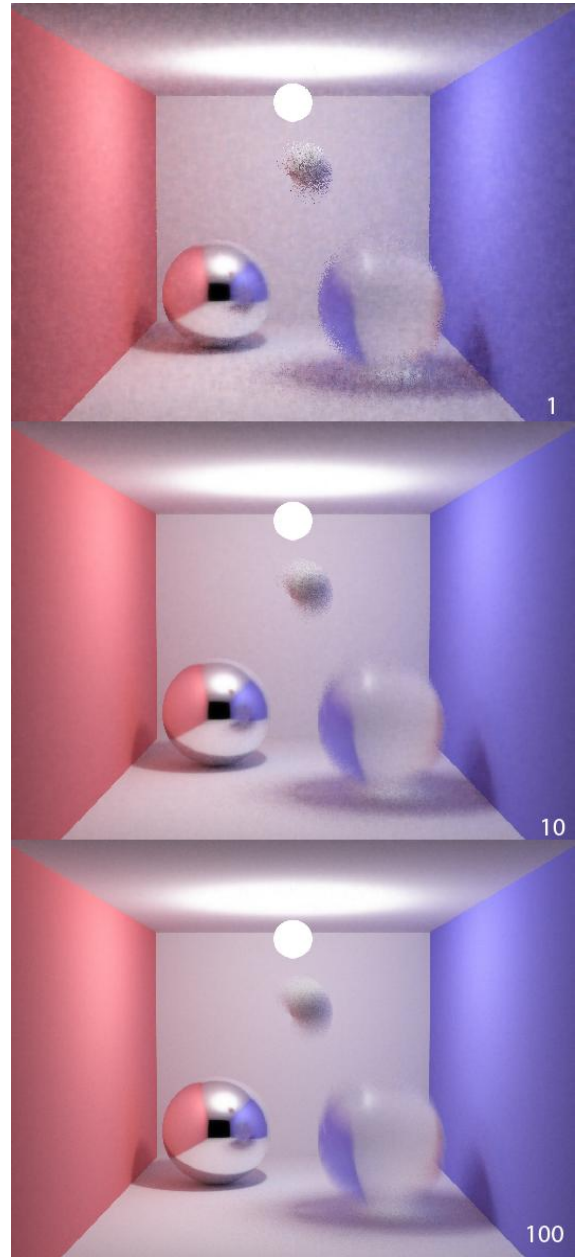
As a stopping criterion of the [Gastal and Oliveira 2012] algorithm, a manifolds tree depth limit is chosen. The depth 2 is considered as a good balance of denoising quality and computing speed. The fixed depth and dimensions amount make the filtration complexity  $O(N)$ .

The algorithm works iteratively, processing the whole video sequence and outputting the resulted video on the each iteration. The following main steps of processing one video frame can be marked out:

- 1) Accumulating information to the current frame from neighbor frames by computing samples as a sum of all frames with the Gaussian weights. We use the Gauss distribution with mathematical expectation equal  $t$  and dispersion equal 1 (the value can be varied like an algorithm parameter).
- 2) Performing high-dimensional filtration of the current frame, using the method of [Gastal and Oliveira 2012]. We reduce the blurring strength  $\sigma$  after the each iteration [Hachisuka et al 2012]:

$$\frac{\sigma_{i+1}^2}{\sigma_i^2} = \frac{i+a}{i+1},$$

where  $i$  – iteration number,  $a$  – a parameter from (0; 1). This way we achieve consistency of the Monte-Carlo estimation and



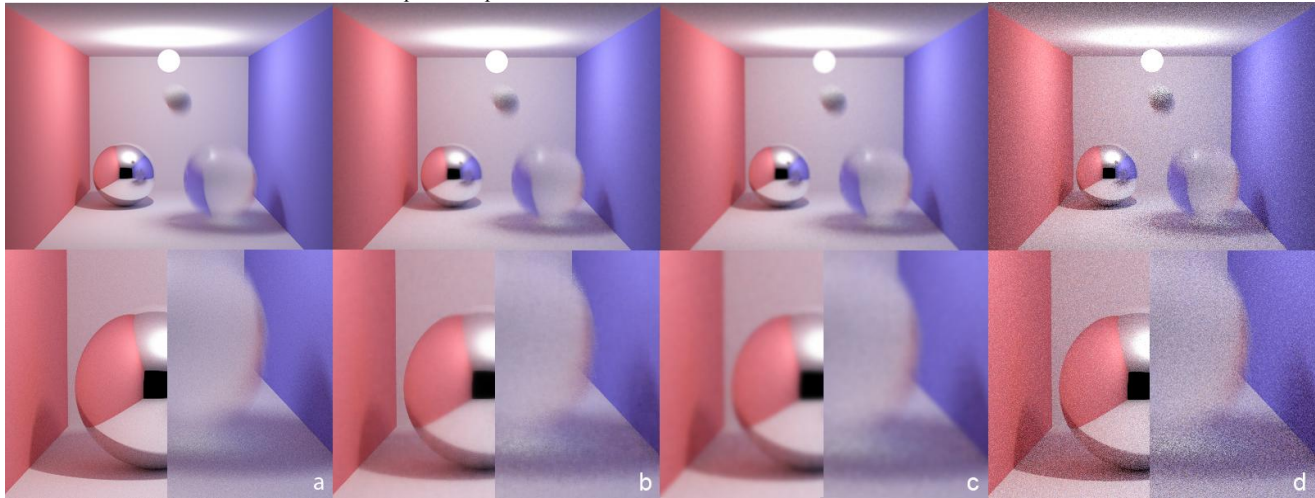
**Figure 3.** Output of the algorithm after 1, 10 and 100 iterations

filtering blur will be vanished when the number of iterations gets infinite.

- 3) The resulted samples are added to a current frame buffer with cumulative values of all previous iterations. The buffer contents can be displayed right after this step.

### 4. RESULTS

Our implementation performs a progressive video rendering through path tracing with 1 sample per pixel and multidimensional filtering per iteration. The method reaches a good output quality after 10 iterations (Fig. 3). Also the motion blur effect, that usually radically influences performance, can be



a) Etalon rendering. 100 spp.

b) Our approach. 10 spp.

c) Progressive rendering with Gaussian blurring with 10 spp.

d) Non-filtered path tracing rendering with 10 spp.

**Figure 4.** Comparison of progressive filtering methods.

achieved with the help of this algorithm with only small additional performance cost.

Taking into account the additional information gives an advantage over other non-multidimensional algorithms. Figure 4 shows that our method yields a result comparable to etalon path tracing after 10 iterations, while Gaussian filter keeps edges blurry. Path tracing of the equal amount of paths outputs a noisy image.

The algorithm was implemented on MATLAB with the help of the filtration scripts of [Gastal and Oliveira 2012]. Frames samples were got from smallpt path tracer [Beason 2010].

Taking into consideration the linear complexity of the [Gastal and Oliveira 2012] filtering and a high speed of tracing 1 path per pixel we expect this algorithm to work in the real-time on the modern GPU. This is our next goal.

## 5. ACKNOWLEDGMENTS

The work is supported by the following grant: RBRF- 12-01-31027 MOL\_A

## 6. REFERENCES

- [1] [Bartovcak and Vrankic 2012] Bartovcak, D. and Vrankic, M.: *Video denoising based on adaptive temporal averaging*. In *Engineering Review* vol. 32, issue 2, 64-69. 2012
- [2] [Beason 2010] Beason, K.: *Path tracer smallpt*. <http://www.kevinbeason.com/smallpt/>
- [3] [Buades et al 2005] Buades, A., Coll, B. and Morel, J.M.: *A non-local algorithm for image denoising*. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (2005)
- [4] [Gastal and Oliveira 2012] *Adaptive manifolds for real-time high-dimensional filtering*. In *ACM SIGGRAPH 2012* vol 31, issue. 4, July 2012
- [5] [Hachisuka et al 2012] Hachisuka, T., Jarosz, W., Bouchard, G., Christensen, P., Frisvad, J.R., Jakob, W., Jensen, H.W., Kaschalk, M., Knaus, C., Selle, A., Spencer, B.: *State of the art in photon density estimation*. In *ACM SIGGRAPH 2012 Courses* (New York, NY, USA, 2012), SIGGRAPH '12, ACM, pp. 6:1-6:469. 4, 6.
- [6] [Liu and Freeman 2010] Liu, C. and Freeman, W.T.: *A high-quality video denoising algorithm based on reliable motion estimation*. In *European Conference on Computer Vision (ECCV)*. 2010.
- [7] [Petschnigg et al 2004] Petschnigg, G., Agrawala, M., Hoppe, H., Szeliski, R., Cohen, M., and Toyama, K.: *Digital photography with flash and no-flash image pairs*. In *ACM Transactions on Graphics*, vol 23, no. 3, pp 664-672, Proceedings of the ACM SIGGRAPH Conference, 2004.
- [8] [Pham and van Vliet 2005] Pham, T.Q. and van Vliet, L.J.: *Separable bilateral filtering for fast video preprocessing*. In: *International Conference on Multimedia and Expo*. (2005)
- [9] [Ritschel et al 2011] Ritschel, T., Dashedbacher, C., Grosch, T., Kautz, J.: *The state of the art in interactive global illumination*. In *Computer Graphics Forum*, vol 31, issue 1, p 160-188, February 2012.
- [10] [Schwenk et al. 2012] Schwenk, K., Kuijper, A., Behr, J., Fellner, D.W.: *Practical noise reduction for progressive stochastic ray tracing with perceptual control*. In *IEEE Computer Graphics and Applications*, vol 32, issue 6, pp 46-55. Nov.-Dec. 2012.
- [11] [Sen and Darabi 2012] Sen, P. and Darabi, S.: *On filtering the noise from random parameters in Monte Carlo rendering*. In *ACM Transactions on Graphics*, vol. 31, issue 3, no. 18, May 2012.
- [12] [Seo and Milanfar 2008] Seo, H.J. and Milanfar, P.: *Video denoising using higher order optimal space-time adaptation*. In *IEEE 1-4244-1484-9/08*. (2008)
- [13] [Tawara et al 2004] Tawara, T., Myszkowski, K., Dmitriev, K., Havran, V., Damez, C., Seidel, H.-P.: *Exploiting temporal coherence in global illumination*. *SCCG 2004*, ISBN:1-58113-967-5
- [14] [Tomasi and Manduchi 1998] Tomasi, C. and Manduchi, R.: *Bilateral filtering for gray and color images*. In: *ICCV*. (1998) 839-846.
- [15] [Yang et al 2009] Yang, Q., Tan K-H., Ahuja, N.: *Real-time O(1) bilateral filtering*. 978-1-4244-3991-1/09 IEEE

## About the author

Gruzdev Alexey is a student at Moscow State University, Department of Computational Mathematics and Cybernetics. His contact email is [aleks-gruzdev92@yandex.ru](mailto:aleks-gruzdev92@yandex.ru).

Frolov Vladimir is a postgraduate at Keldysh Institute of Applied Mathematics. His contact email is [vfrolov@graphics.cs.msu.ru](mailto:vfrolov@graphics.cs.msu.ru).

Vostryakov Konstantin is an engineer at NVIDIA. His contact email is [kvostryakov@nvidia.com](mailto:kvostryakov@nvidia.com).

Ignatenko Alexey is a researcher, candidate of physical and mathematical science at Moscow State University, Department of Computational Mathematics and Cybernetics. His contact email is [ignatenko@graphics.cs.msu.ru](mailto:ignatenko@graphics.cs.msu.ru).

# Image Filtration Based on Principal Component Analysis and Nonlocal Image Processing: Algorithms and Applications

Andrey Priorov, Kirill Tumanov, Vladimir Volokhov  
Department of Physics

P.G. Demidov Yaroslavl State University, Yaroslavl, Russia  
andcat@yandex.ru, tumanov@susqu.edu, volokhov@piclab.ru

## Abstract

The present paper is focused on the comparison of application results gathered from several algorithms based on Principal Component Analysis (PCA) and non-local processing. Its goal is to unveil the limitations, show best practices and make recommendations of use for each of the studied filtration schemes. Studied applications included filtration of an additive white Gaussian noise (AWGN) and a mixed noise from greyscale and colour images, and the removal of blocking artefacts from greyscale images.

**Keywords:** *image filtration, principal component analysis, non-local processing, comparison, applications.*

## 1. INTRODUCTION

Chatterjee and Milanfar in 2010, have shown that the theoretical limit of image reconstruction hasn't been yet achieved [2]. There still are debates on how to increase performance of filtration techniques used today. Among the widest spread methods of cancelling an AWGN in digital images, according to [7], are the algorithms which base on: local, non-local, pointwise and multipoint processing types.

The main problems with the quality of reconstructed images are: a Gibbs effect, which becomes highly noticeable on images containing objects with high brightness contrast on their outer edges, and an edge blurring of objects on an image being processed. Both of these effects highly degrade an image perception and the affected images do not suite for high demands. That is why in addition to their primary task contemporary filtration algorithms are generally required to reduce an impact of the named effects.

Algorithms taken for the study originally were implemented for AWGN filtration. However literature on digital images noise cancelling shows that modern AWGN filtration methods used for greyscale images may be successfully used in a series of other digital image processing tasks. Examples of such tasks are: colour image filtration, filtration of "raw" images, deletion of blurring from objects' edges, sharpening of objects' edges, etc.

Usage of the AWGN model may be explained with the help of statistics theory, namely – central limit theorem. It has an important practical value and may be interpreted to describe the work of devices containing numerous independent additive noise sources, each of which has its own random distribution, generally unknown. Resulting sum of these noise distributions is best described as a Gaussian distribution. On practice AWGN model well suits to simulate a thermal noise which is inevitably observed in digital components such as charge-coupled devices (CCDs) or CMOS matrixes.

## 2. USED FILTRATION SCHEMES DESCRIPTION

For the present study we took two similar methods which both used PCA and non-local processing approaches and a basic fundamental PCA method used in both of them.

### 2.1 Two-stage PCA filtration scheme

The most basic method used was the modification of two-stage PCA filtration scheme (Adaptive PCA + empiric Wiener filter (APCA+Wiener for short)).

The first processing stage forms a first "raw" evaluation  $\hat{\mathbf{x}}^I$  of an unnoised image  $\mathbf{x}$ . After that, on the second processing stage, a "fine" evaluation  $\hat{\mathbf{x}}^{II}$  of an unnoised image  $\mathbf{x}$  is formed based on the "raw" evaluation  $\hat{\mathbf{x}}^I$ , received after the previous stage.

We decided to test this filtration scheme along with two more advanced ones in order to see, how the latter two perform in comparison with the APCA+Wiener, which is one of their major components.

### 2.2 Sequential filtration scheme

First, as it was noted, this scheme includes an abovementioned APCA+Wiener filtration scheme as a base which forms an input for non-local denoising algorithm. The latter algorithm calculates the non-local means discussed previously [1]. As a result we receive a final non-local evaluation of the processed pixel  $\hat{\mathbf{x}}^{II}(i, j)$  using the following formula:

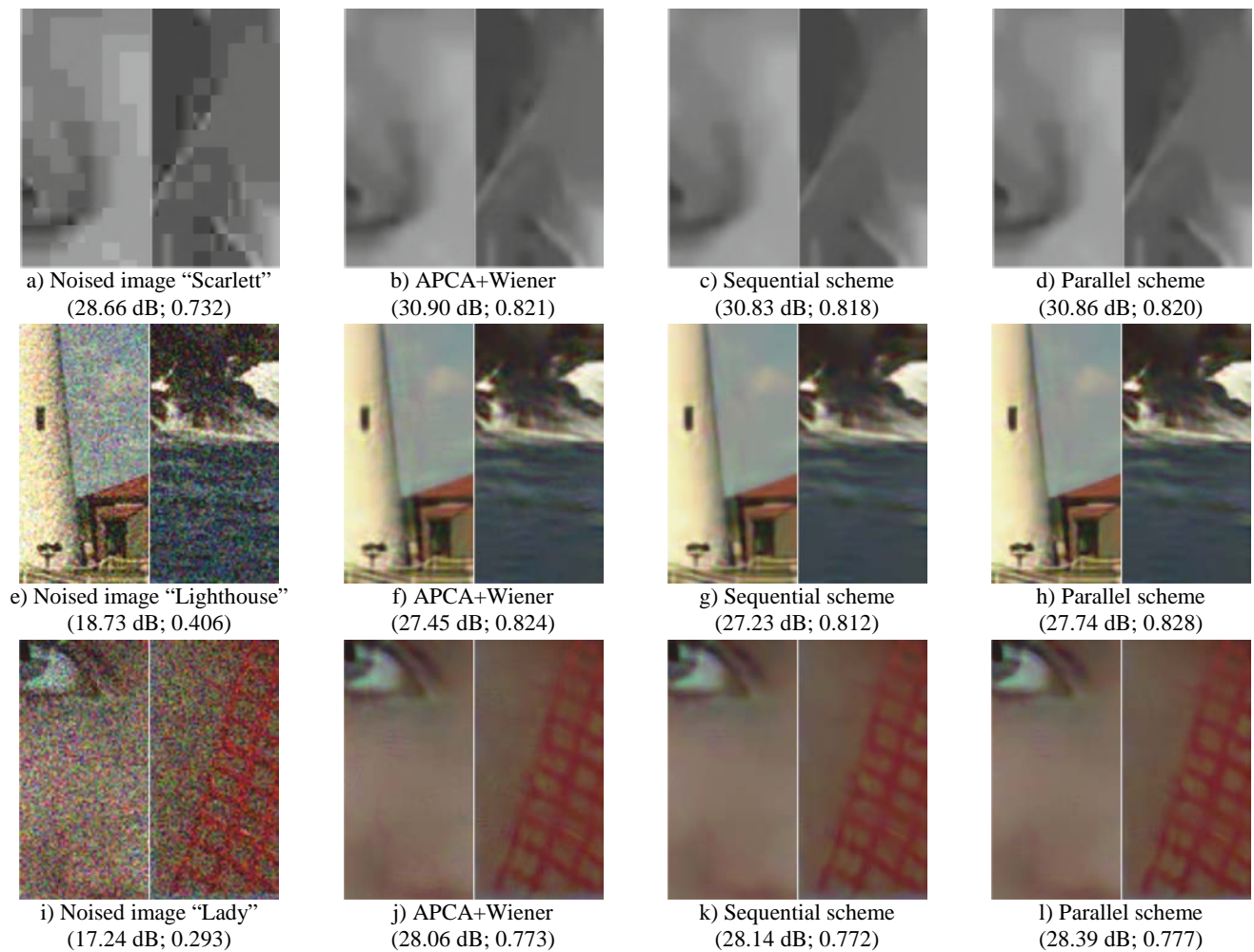
$$\hat{\mathbf{x}}^{III}(i, j) = \sum_{k,l} g_{h^m}(i, j, k, l) \hat{\mathbf{x}}^{II}(k, l), \quad (1)$$

$$\text{where } g_{h^m}(i, j, k, l) = \frac{w_{h^m}(i, j, k, l)}{\sum_{k,l} w_{h^m}(i, j, k, l)}. \quad (2)$$

### 2.3 Parallel filtration scheme

The last method used in this work is a parallel filtration scheme based on the same algorithms which were used in the previous method.

Notable is that the "Two-stage PCA based filtration" block performs completely the same tasks that it does in a sequential scheme. On the other hand, contrary to the previous method, block "Non-local algorithm of image denoising" processes a noised image  $\mathbf{y}$ , not a second evaluation  $\hat{\mathbf{x}}^{II}$  of an unnoised image  $\mathbf{x}$ . Wherein weight of a pixel  $y(k, l)$  similar to a processed pixel  $y(i, j)$  in a final evaluation  $\hat{\mathbf{x}}^{III}$  of an unnoised image  $\mathbf{x}$ , received as an output of the block, is calculated using the formula:



**Figure 1:** Examples of image reconstruction "Scarlett" ( $Q = 5$ ,  $\sigma = 25$ ), "Lighthouse" ( $\sigma = 30$ ), and "Lady" ( $\sigma_1 = 15$ ,  $\sigma_2 = 0.2$ ) processed by APCA+Wiener, sequential and parallel filtration schemes. In brackets PSNR, dB and MSSIM

$$w_{h^{\text{III}}}(i, j, k, l) = \exp \left\{ - \frac{\sum_{m, n \in N} g_a(m, n) \cdot [\hat{x}^{\text{II}}(i+m, j+n) - \hat{x}^{\text{II}}(k+m, l+n)]^2}{(h^{\text{III}})^2} \right\}. \quad (3)$$

### 3. USAGE OF THE FILTRATION SCHEMES IN MODERN IMAGE PROCESSING TASKS

In this section we will show how the described AWGN filtration methods in addition to their primary use may be used for: (a) denoising AWGN-affected colour images; (b) filtration of mixed noises from greyscale images; (c) suppression of blocking artefacts in compressed JPEG images; (d) filtration of mixed noises from colour images.

#### 3.1 Removal of blocking artefacts

The task was formulated as a situation where an image compression using JPEG algorithm is used as a noise model [5]. In this case a noise component  $\mathbf{n}$  may be treated as a result of distortion connected with blocking artefacts on a digital image. Then a solution to this task may be found as dispersion  $\sigma^2$  of a

noise component  $\mathbf{n}$ . A possible way of finding  $\sigma^2$ , using an *a priori* knowledge about a quantization matrix of JPEG standard coefficients, is shown in [4]. In this study search of  $\sigma^2$  was performed manually.

For our experiments on blocking artefacts removal we used the same source of greyscale images [9]. We tested our algorithms on  $256 \times 256$  pixels and  $512 \times 512$  pixels images.

JPEG compression quality parameter  $Q$  was used to set the degree of compression, and  $\sigma^2$  varied to demonstrate a dependence of the image reconstruction quality from the filtration smoothing parameter.

Notable is that the average increase rate for each algorithm was relatively low both on PSNR and MSSIM scales. Images compressed with  $Q = 15$  after the processing with each of the algorithms were more damaged than reconstructed. Therefore it



can be concluded that neither of the studied algorithms may be applied to the JPEG compressed images with  $Q \geq 15$ . Although they remove blocking artefacts from the input image each of them gives a decrease in MSSIM value of a reconstructed image. This decrease is expressed in smoothing too much detail from test images and in most cases is considered unacceptable.

Special attention through all our further test analysis was devoted to the best performance results for each combination of variables and an algorithms' comparison based on this data.

Applying sequential filtration scheme to images with resolution higher than  $256 \times 256$  pixels proved to be inadvisable because it showed lower average increase rates and it gave the least number of best reconstruction results both for PSNR and MSSIM. In addition it was observed that in higher resolution images the MSSIM decrease becomes more exponential and PSNR decreases linearly with the  $Q$  growth. However APCA+Wiener filtration scheme which showed the minimum number of best results for  $256 \times 256$  pixels images performed much better on  $512 \times 512$  pixels images and its average quality increase rates on average were better than the ones of sequential filtration scheme. This makes us consider the APCA+Wiener filtration method applicable for this task. On the other hand, parallel scheme strengthened its positions among the compared algorithms.

Results of JPEG compressed greyscale digital images filtration with the discussed filtration schemes are visualised on Figure 1 on example of "Scarlett"  $512 \times 512$  pixels image. Hereinafter only fragments of images are shown for easier comparing.

It can be concluded that all the named filtration methods may be successfully applied to the task of removal of blocking artefacts with the notion to the listed limitations, however the reconstructed images quality shows to be relatively low and thus a further research in this area is needed.

### 3.2 AWGN-affected colour images filtration

The task is of especially current interest from the standpoint of modern applications. That is why numerous solutions have been formulated to perform it. The one we used in the present work is a direct channelwise processing of an RGB image. For simplicity we did no transfer from RGB images to images with separated colour and brightness information [4]. AWGN was added to each channel independently with the same characteristics. Relevancy of use of the described noise model may be confirmed with the presence of image capture systems which consist of three separate CCDs or CMOS matrixes. This method was used for simplicity and for further research it may be extended by using specific noise models and applying them to each image layer in a variation of interest.

For this test we used  $768 \times 512$  pixels colour images from the CIPR's Kodak image database [8]. We used AWGN with  $\sigma$  values in a range from 15 to 25.

Notable is the fact that through our entire test series sequential scheme never showed a best performance neither in PSNR nor in MSSIM. This enforces our proposal of use the parallel filtration scheme with its approach of using two independent evaluations of an unnoised image  $\mathbf{x}$ . It should also be mentioned that APCA+Wiener filtration scheme showed very competitive results in terms of MSSIM. This scheme even outperformed the parallel scheme for AWGN with  $\sigma = 30$ . That is why this scheme

may be of use when a "good" instead of "excellent" colour images filtration results are needed.

All the compared filtration methods provide a high-quality processing of main objects' edges. This fact shows its results in this test series – absolute values of PSNR and MSSIM decrease, but carefully processed edges slow this decrease for MSSIM.

Applying sequential filtration scheme to the task of colour images filtration proved to be infeasible as well as for the removal of blocking artefacts. At the same time parallel scheme showed almost absolute best performance in this task, especially according to PSNR quality assessment of the reconstructed images.

Results of AWGN-affected colour digital images filtration with the named filtration schemes are visualised on Figure 1 on example of "Lighthouse"  $768 \times 512$  pixels image.

It can be concluded that APCA+Wiener and parallel filtration methods may be successfully applied to the task of AWGN-affected colour images filtration. Quality of the reconstructed images for these methods is rather high, although on high-resolution colour images the smoothing effect, which arises after filtration procedures, becomes more visible, due to the superposition of different image layer filtration defects. The smart way of layers integration may be of good help in solving the issue, and its implementation requires an additional study.

### 3.3 Mixed noise images filtration

The discussed AWGN model may be complicated by a usage of mixed noise model. An example of such model was proposed by Hirakawa and Parks in 2006 [6] to characterize noise of CMOS matrixes. The model may be described as follows:

$$\mathbf{y} = \mathbf{x} + (\sigma_1 + \sigma_2 \mathbf{x})\mathbf{n}, \quad (4)$$

Where  $\sigma_1$  and  $\sigma_2$  – are the constants which determine a noisiness degree, and  $\mathbf{n}$  – is an AWGN with zero mean and  $\sigma = 1$ . If  $\sigma_2 = 0$  this noise model transforms into the described earlier AWGN model.

Because of the irregular character of noise dispersion in the mixed noise model, which is explained by the dependency of noise from the initial signal, a direct application of the described schemes is impossible. For this reason we used a generalized homomorphic filtration method [3], proposed by Ding and Venetsanopoulos in 1987. The idea of this method is in using a logarithm-type transform to interpret noised data  $\mathbf{y}$  as a sum of an initial unnoised signal and AWGN, process them with described filtration schemes and then reconstruct the data with the inverse transform.

For this test we used all the mentioned above images -  $256 \times 256$  and  $512 \times 512$  pixels greyscale and  $768 \times 512$  pixels colour images from [8,9]. We used a mixed noise with  $\sigma_1$  values in a range from 15 to 25 and  $\sigma_2$  values in a range from 0.1 to 0.3.

The parallel filtration scheme showed best results of image reconstruction on a PSNR scale in a prevailing number of tests. However, MSSIM quality assessment results were almost equally distributed between all three filtration schemes. This may be explained by the fact that the MSSIM values are formed based on evaluating the image, which colour layers were processed independently, so that each scheme at the end formed a synergetic reconstructed image.

Although sequential filtration scheme showed nearly as many best results as parallel scheme on MSSIM scale for  $256 \times 256$  pixels greyscale images, application of the sequential filtration scheme to this task is infeasible for the higher resolution images and colour images.

PSNR and MSSIM increase for correlating pairs of results is almost linear. All the compared filtration methods provide a high-quality processing of main objects' edges and filtration quality in general.

Results of mixed noise affected colour digital images filtration with the discussed filtration schemes are visualised on Figure 1 on example of "Lady"  $768 \times 512$  pixels image.

Application of all three algorithms to images affected by this noise model on high levels of  $\sigma_1$  and  $\sigma_2$  resulted in visible colour changes of minor image details and objects. For example, on a "Caps" colour image several little clouds previously of a white colour were reconstructed as red-like, because of the high number of red noise pixels on an input noised image. We consider this type of reconstruction defects significant as they are easily noticeable, and we understand that for a successful use of the discussed filtration schemes to the mixed noise filtration on colour images some additions to the algorithms need to be made. However the overall quality of reconstructed images which were noised with  $\sigma_2 = \{0.1, 0.2\}$  is high and the defects described above are unnoticeable. That is why it can be concluded that APCA+Wiener and parallel filtration methods may be successfully applied to the task of mixed noise affected greyscale and colour images filtration with limitation in using the high  $\sigma_2$  values for colour images.

#### 4. COMPARISON OF THE USED FILTRATION METHODS

Here we give a brief discussion on the filtration schemes performance in the described digital image processing applications.

##### 4.1 Modification of the two-stage PCA filtration scheme

The most advantageous feature of this method is its low computational cost and construction simplicity.

Primary disadvantages of using this filtration scheme from the standpoint of reconstructed images quality are: (1) substantial amount of ringing artefacts on image objects' edges, this effect is especially visible on high-contrast image parts (see Figure 1); (2) high blurring of image objects' edges, compared to other modern filtration methods [4].

##### 4.2 Sequential filtration scheme

Advantages of this method are in its relative construction and implementation simplicity and the decrease of the amount of ringing artefacts on image objects' edges (see Figure 1).

Primary disadvantages of using this filtration scheme are in the presence of high blurring of image objects' edges and high computational cost of the filtration algorithm.

#### 4.3 Parallel filtration scheme

Advantages of this method are: (1) high quality of the reconstructed images both on PSNR and MSSIM scales; (2) minimal amount of ringing artefacts on image objects' edges, and low blurring of image objects' edges (see Figure 1).

Primary disadvantage of using this filtration scheme is in the high computational cost of the filtration algorithm.

#### 5. CONCLUSION

Our study has shown how different digital image filtration algorithms based on the PCA and non-local processing may be applied to modern digital image processing tasks. Experimental results obtained prove the idea of successful application of these filtration methods to the removal of blocking artefacts, AWGN- and mixed noise affected image filtration. In the present work we listed the limitations of use for each method and proposed approaches of their overcome. Results of other contemporary state-of-the-art filtration algorithms are not provided due to the absence of relevant experimental data. A thorough comparison may be a topic of our further investigation.

#### 6. AKNOLEDGMENTS

This work was supported in part by the Russian Foundation for Basic Research under Grant № 12-08-01215-a "Development of methods for quality assessment of video".

#### 7. REFERENCES

- [1] Buades A. Image and film denoising by non-local means. *PhD thesis*, Universitat de les Illes Balears. 2005.
- [2] Chatterjee P., Milanfar P. Is denoising dead? // *IEEE Trans. Image Processing*. 2010. V. 19, №4, pp. 895–911.
- [3] Ding R., Venetsanopoulos A.N. Generalized homomorphic and adaptive order statistic filters for the removal of impulsive and signal-dependent noise // *IEEE Trans. Circuits Syst.* 1987. V. CAS-34, №8, pp. 948–955.
- [4] Foi A., Katkovnik V., Egiazarian K. Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images // *IEEE Trans. Image Processing*. 2007. V. 16, №5, pp. 1395–1411.
- [5] Gonsales R., Woods R. *Digital image processing // Prentice Hall*. 2008.
- [6] Hirakawa K., Parks T.W. Image denoising using total least squares // *IEEE Trans. Image Processing*. 2006. V. 15, №9, pp. 2730–2742.
- [7] Katkovnik V., Foi A., Egiazarian K., Astola J. From local kernel to nonlocal multiple-model image denoising // *Int. J. Computer Vision*. 2010. V. 86, №8, pp. 1–32.
- [8] The RPI-CIPR Kodak image database, <http://www.cipr.rpi.edu/resource/stills/kodak.html>. 2013
- [9] University of Granada Computer Vision Group test images database, <http://decsai.ugr.es/cvg/dbimagenes>. 2013

#### About the author

The authors are with P.G. Demidov Yaroslavl State University, Yaroslavl, Russia. (email: andcat@yandex.ru; tumanov@susqu.edu; volokhov@piclab.ru).

# Wavelet Lifting on Application Specific Vector Processor

David Barina

Pavel Zemcik

Faculty of Information Technology  
Brno University of Technology  
{ibarina, zemcik}@fit.vutbr.cz

## Abstract

With the start of the widespread use of discrete wavelet transform the need for its efficient implementation is becoming increasingly more important. This work presents a general approach of discrete wavelet transform scheme vectorisation evaluated on an FPGA-based Application-Specific Vector Processor (ASVP). This unit can be classified as SIMD computer in Flynn's taxonomy. The presented approach is compared with two other non-vectorised approaches. Using the frequently exploited CDF 9/7 wavelet, the achieved speedup is about  $2.6\times$  compared to naive implementation.

**Keywords:** discrete wavelet transform, lifting scheme, SIMD, parallelization, vectorisation

## 1 Introduction

The discrete wavelet transform (DWT) is mathematical tool which is able to decompose discrete signal into lowpass and highpass frequency components. Such a decomposition can be performed at several scales. DWT is often used as the basis of sophisticated compression algorithms. This is the case of JPEG 2000 and Dirac compression standards in which CDF 9/7 wavelet [5] is employed for lossy compression. Responses of this wavelet can be computed by a convolution with two FIR filters, one with 7 and the other with 9 coefficients. For the DWT computation, the well known Mallat's [9] filtering scheme can be used. Alternatively, one can use usually faster scheme called lifting which was presented by I. Daubechies and W. Sweldens in [6]. Lifting data flow graph consists of regular grid computational scheme suitable for SIMD vectorisation. Both of the algorithms can be performed over some approximation of real numbers. This paper focuses on single-precision floating-point (SP FP) format.

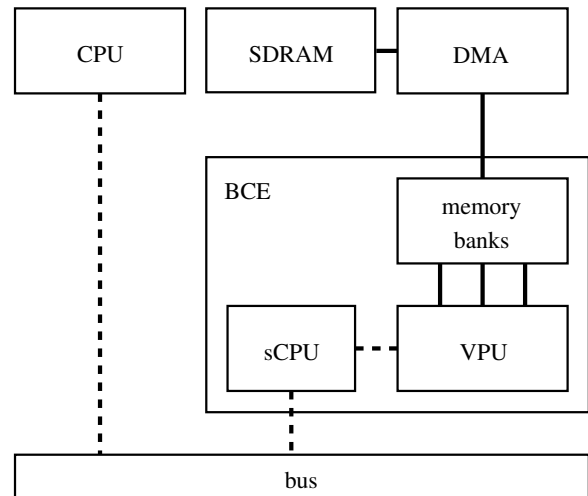
In the field of FPGA-based accelerators, the platform called Application-Specific Vector Processor (ASVP) was recently proposed [11], [10]. This platform uses several simple units referred as Basic Computing Elements (BCEs). BCEs are able to accelerate simple operations (like addition or multiplication) on long single-precision floating-point vectors. Thus, these units can be classified as SIMD computers in Flynn's taxonomy. Lifting of CDF 9/7 transform can be directly adapted on them.

In this work, we discuss vectorisation (parallelization) of discrete wavelet transform on ASVP platform. The rest of the paper is organized as follows. More traditional approaches to DWT computation together with ASVP platform are reviewed in Section 2. Section 3 gives the basics behind the lifting scheme. Section 4 describes opportunities for lifting scheme parallelizations and presents our proposed approach. Introduced parallelization methods are compared in section 5. Finally, Section 6 concludes the paper.

## 2 Related work

In 2000, the problem of minimum memory implementations of lifting scheme was addressed in [4] by Ch. Chrysafis and A. Ortega. This approach is very general and it is not focused on parallel processing. The work was also later extended to [3] where same authors addressed a problem of minimum memory implementation of 2-D transform. Also, variation of this approach was presented six year later in [7] which is specifically focused on CDF 9/7 wavelet transform.

In [8] R. Kutil *et al.* presented SIMD vectorisation of several frequently used wavelet filters. This vectorisation is applicable only on those filters discussed in their paper. Specifically, vectorisation of CDF 9/7 wavelet computed using lifting scheme is vectorised here by a group of four successive pairs of coefficients. For CDF 9/7 wavelet discussed in this paper, their measurements gave a speedup of 2.65 for forward and 1.7 for backward transform on Intel Pentium 4 with SSE extension.



**Figure 1:** Organization of ASVP platform. Solid lines indicates data paths. In our case, function of BCE is controlled by host CPU (MicroBlaze). The sCPU means for simple CPU what is the PicoBlaze processor here. Moreover, the BCE element consists of four memory banks, each 1024 32-bit words long, and Vector Processing Unit (VPU) which performs actual operations. The BCE accesses RAM through DMA engine.

The platform used in this paper is Application-Specific Vector Processor (ASVP, originally EdkDSP) recently presented in [11], [10], [1] and [2]. This heterogeneous multi-core platform employs up to several units called Basic Computing Element (BCE) which can accelerate floating-point vector operations. For organization of ASVP see Figure 1. These elements use a combination of a simple Pi-

coBlaze CPU (sCPU in Figure 1) with a configurable pipelined datapath. The computation performed by BCE can be changed through replacing the PicoBlaze firmware. Moreover, the ASVP platform contains host CPU (MicroBlaze in this case) that is executing the main program. Thus, the computation is distributed between host CPU and one or more BCE units. This change of the BCE firmware can be made from MicroBlaze CPU in runtime. The BCE contains four memory banks each of 1024 words long (one word denotes 32-bit SP FP). Before BCE can start its program, the input data must be transferred from main DDR memory into BCE's memory banks. Similarly, the output data should be transferred back when BCE computation is done. These data are transferred by DMA controller. The operations performed by BCE are element-wise move, addition, multiplication, etc.

### 3 Lifting scheme

According to the number of arithmetic operations, the lifting scheme [6] is today's most efficient scheme for computing discrete wavelet transforms. Any discrete wavelet transform with finite filters can be factored into a finite sequence of  $N$  pairs of predict and update convolution operators  $P_n$  and  $U_n$ . Each predict operator  $P_n$  corresponds to a filter  $p_i^{(n)}$  and each update operator  $U_n$  to a filter  $u_i^{(n)}$ .

$$P_n(z) = \sum_{i=-l_n}^{g_n} p_i^{(n)} z^{-i} \quad (1)$$

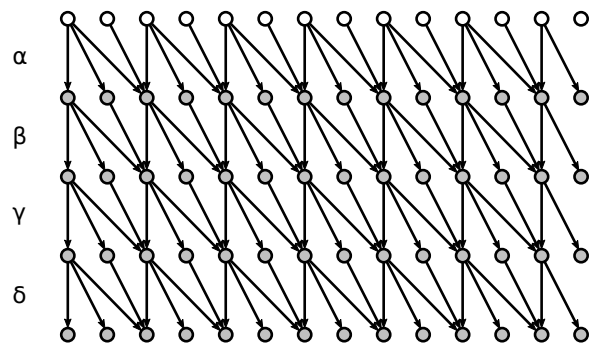
$$U_n(z) = \sum_{i=-m_n}^{f_n} u_i^{(n)} z^{-i} \quad (2)$$

This factorisation is not unique. For symmetric filters, this non-uniqueness can be exploited to maintain symmetry of lifting steps.

Consider the decomposition of the signal of length of  $L$  samples. Without loss of generality one can assume only signals with even length  $L$ . Possible remaining coefficient can be treated separately in the prolog or epilog phases together with border extension. Thus, the transform contains  $S = L/2$  pairs of resulting wavelet coefficients  $(s, d)$ . The  $s$  coefficients represent a smoothed signal. On the contrary, the  $d$  coefficients form a difference or detail signal.

In their paper [6], Daubechies and Sweldens demonstrated an example of CDF 9/7 transform factorisation which resulted into four lifting steps ( $N = 2$ ) plus scaling of coefficients. In this example, the individual lifting steps use 2-tap symmetric filters for the prediction as well as the update. In all figures shown in this paper, the coefficients of these four 2-tap symmetric filter are denoted  $\alpha, \beta, \gamma$  and  $\delta$  respectively.

When coefficient scaling is omitted, the calculation of a pair of the DWT coefficients at the position  $l$  ( $s_l$  and  $d_l$ ) is performed through four lifting steps. Intermediate results ( $s_l^{(n)}$  and  $d_l^{(n)}$ ) can be appropriately shared between neighbouring pairs of coefficients ( $s_l$  and  $d_l$ ). Finally, the calculation of the complete CDF 9/7 DWT is depicted in Figure 2. This is an in-place implementation, which means the DWT can be calculated without allocating auxiliary memory. Resulting coefficients ( $s_l$  and  $d_l$ ) are interleaved in place of the input signal.



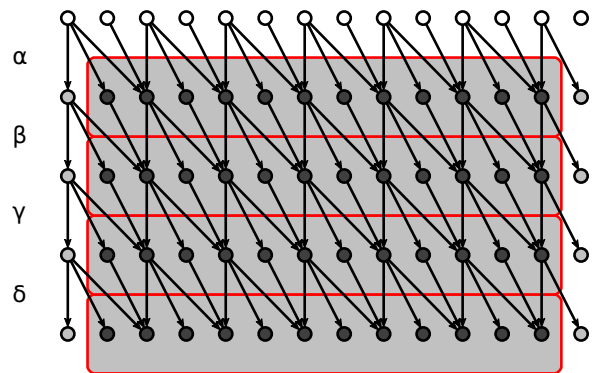
**Figure 2:** Complete data flow graph of CDF 9/7 wavelet transform. The input signal is on top, output at the bottom. The graph borders must be treated in a special way using prolog and epilog phases.

### 4 Vectorisation

The calculation scheme described in the previous section can be realized in a number of different ways. In this work, two of such ways are described. The main difference between them is in the order of lifting steps evaluation. Alternatively, the data flow graph in Figure 2 can be split into areas that are evaluated sequentially according to their data dependencies.

#### 4.1 Horizontal vectorisation

The naive approach of data flow graph evaluation directly follows the lifting steps ( $n$ ). Thus, all intermediate  $s^{(1)}$  and  $d^{(1)}$  coefficients are evaluated in the first step. Then, all  $s^{(2)}$  and  $d^{(2)}$  are evaluated in second step, etc. Unfortunately, this algorithm requires several reads and writes of the intermediate results  $s_l^{(n)}$  and  $d_l^{(n)}$ . For long signals, these intermediate results will be several times evicted from the CPU cache in favor of other intermediate results. Consequently, many cache misses during such a computation will occur.



**Figure 3:** The horizontal vectorisation of the CDF 9/7 data flow graph. The scaling of coefficients was omitted. The computation within the highlighted areas can be processed in parallel.

In this paper, this method is called the horizontal vectorisation. This name reflects the fact that the data flow graph is split in horizontal areas as in Figure 3. In each area, the elementary calculations are

independent and can be computed in parallel. For simplicity, the scaling of coefficients and the prolog and epilogs phases were omitted in the referenced figure. An entire signal of  $2S$  samples must be loaded into the memory which is not suitable for memory limited systems.

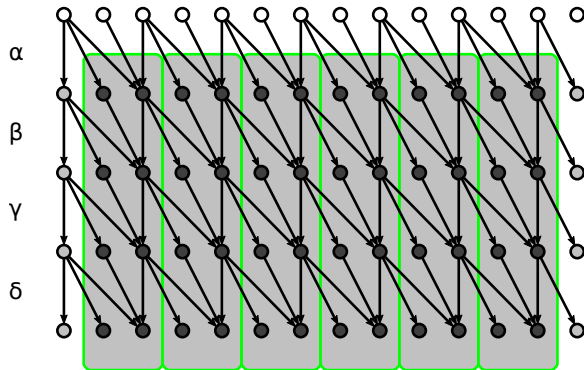
## 4.2 Vertical vectorisation

Another way of lifting data flow graph evaluation is the double-loop approach [7]. This approach is referred to as the vertical vectorisation. Earlier, it was described in [4] focusing on low memory systems but without vectorisation.

The  $P_n$  and  $U_n$  filters need not be causal. In general, non-causal systems requires storing the whole input signal into memory (as can be seen from Figure 3). This is not suitable for fast or memory limited signal processing nor for a vectorisation. Therefore, it would be appropriate to convert non-causal lifting steps ( $P_n$  and  $U_n$ ) to causal systems. The key to force these filtering steps to be causal is the introduction of appropriate delays.

$$\mathcal{P}_n(z) = z^{-l_n} P_n(z) = \sum_{i=0}^{g_n+l_n} p_{i-l_n}^{(n)} z^{-i} \quad (3)$$

$$\mathcal{U}_n(z) = z^{-m_n} U_n(z) = \sum_{i=0}^{f_n+m_n} u_{i-m_n}^{(n)} z^{-i} \quad (4)$$



**Figure 4:** Vertical vectorisation of the CDF 9/7 data flow graph. The computation within the highlighted areas cannot be processed in parallel due to data dependencies.

The transition from non-causal to causal system introduce a delay  $z^{-l_n}$  on both inputs of the prediction filtering step  $P_n$ . In the bottom input  $s$ , the delay can be distributed into both branches. This leads to a causal system  $\mathcal{P}_n$  as in (3). Analogously, a delay of  $m_n$  samples is introduced on both inputs of update step  $U_n$ . Again, this delay can be distributed into branches of upper input  $d$ . The resulting equation is shown in (4). For simplicity, the adjacent delays can be combined into single one. Finally in (5), delays of  $\eta_n$ ,  $\mu_n$  and  $\nu_n$  samples appear around each pair of filtering steps  $\mathcal{P}_n$  and  $\mathcal{U}_n$ . The resulting block diagram is shown in Figure 5.

$$\eta_n = l_n \quad (5a)$$

$$\mu_n = l_n + m_n \quad (5b)$$

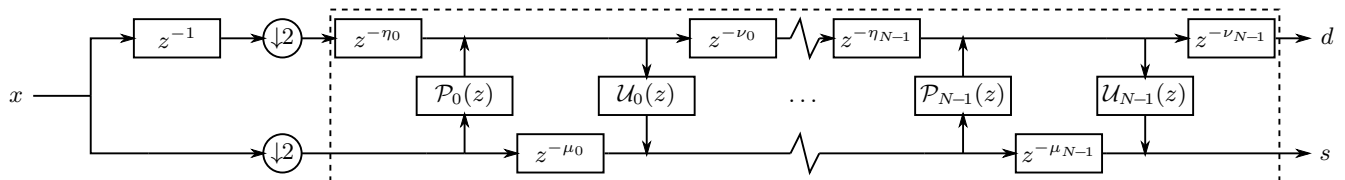
$$\nu_n = m_n \quad (5c)$$

In this method, the lifting computation is transformed into one loop instead of multiple loops over all the coefficients. Therefore, one pair of lifting coefficients  $s_l$  and  $d_l$  is computed in each iteration of such a single loop. However, the computations within each of these areas cannot be directly parallelized due to data dependencies. Even so, this procedure is advantageous because the coefficients are read and written only once. Consequently, this prevents unnecessary cache misses. In our 1-D case, the SIMD vectorisation of this method lies in processing of several adjacent areas in parallel like in [8]. The data flow graph is split in vertical areas of width of two coefficients as in Figure 4. Furthermore, this approach is particularly useful for multidimensional (e.g. 2-D) transform on PC platform where several data rows are processed in single loop at once using n-fold SIMD instructions.

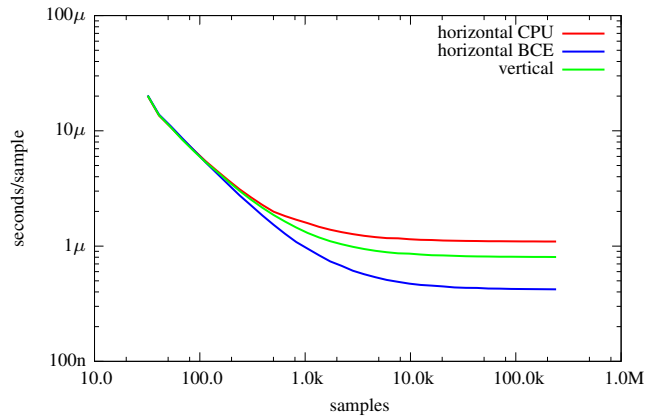
## 5 Results

The implementations of the approaches described in the previous section was compared on ASVP platform. This comparison was performed on forward DWT using CDF 9/7 wavelet. All the implementations work over a sequence of single-precision floating point numbers. According to platform performance, a length of the sequence was progressively extended from vector of 32 samples with geometrical step of 1.28 up to 240 thousands of samples. The transform was computed including a final coefficient scaling.

Our configuration contains 32-bit MicroBlaze as host CPU and two BCE acceleration units (only one used for 1-D transform). Used bitstream fits into Spartan-6 SP605 FPGA kit. We use PetaLinux as operating system on host CPU. The evaluated programs had been compiled by GCC 3.4.1 with `-O2 -mno-x1-soft-mul -mhard-float` options.



**Figure 5:** Block diagram of vertical lifting scheme vectorisation. The part bounded with dashed line correspond to the area of parallel computation.



**Figure 6:** Comparison of three described approaches on the ASVP platform. The horizontal parallelization was implemented on the CPU as well as on the BCE unit. Using the BCE unit, horizontal parallelization is clearly the fastest method with up to  $2.6\times$  speedup.

Evaluation on ASVP platform is summarized in Figure 6. The horizontal axis of this graph indicates the sequence length. The vertical axis specifies computation time per one signal sample. Both approaches from previous section as implemented on MicroBlaze CPU are plotted in this graph. Furthermore, another implementation of the horizontal parallelization accelerated using BCE unit is plotted here. Clearly, the horizontal parallelization is the fastest method when BCE is used. Without BCE, the fastest approach seems to be the vertical parallelization. The speedup of horizontal approach with BCE over baseline horizontal approach on CPU is up to  $2.6\times$ .

vectorisation	$t$ samples	$b$ coefficients	$q$ operations
horizontal	$2S$	$2S$	$S$
vertical	$2T$	$2N$	$T$

**Table 1:** Memory consumption of vectorisation methods. Each method needs  $t$  samples to start iteration and  $b$  memory words to pass intermediate results between them. In each iteration, up to  $q$  operations can be evaluated in parallel.

vectorisation	$\mu\text{s/sample}$	speedup
CPU horizontal	1.1	1.0
CPU vertical	0.8	1.4
BCE horizontal	<b>0.4</b>	<b>2.6</b>

**Table 2:** Execution times per sample measured for 240 thousands of samples. All times are related to the CDF 9/7 transform.

## 6 Conclusion

In this paper, two known methods of lifting scheme evaluation was compared on ASVP platform. The achieved speedup is up to  $2.6\times$  using Application Specific Vector Processor. The best results were obtained using horizontal parallelization performed on one BCE computing unit. This unit can accelerate operations on vectors with up to 1024 elements of length. Operations on longer vectors have

to be split into several fragments. The resulting firmware consists of 15 individual operation calls (addition, multiplication) including scaling of lifting coefficients.

Our next research will focus to an adaptation of the proposed approach to the 2-D wavelet transform. Specifically, we will use the single-loop approach proposed by R. Kutil in [7].

## Acknowledgements

This work has been supported by the EU FP7-ARTEMIS project IMPART (grant no. 316564) and the national TAČR project RODOS (code TE01020155).

## References

- [1] R. Bartosinski, M. Daněk, J. Sýkora, L. Kohout, and P. Honzík. Foreground detection and image segmentation in a flexible ASVP platform for FPGAs. In *Conference on Design and Architectures for Signal and Image Processing (DASIP)*, pages 1–2, 2012.
- [2] R. Bartosinski, M. Daněk, J. Sýkora, L. Kohout, and P. Honzík. Video surveillance application based on application specific vector processors. In *Conference on Design and Architectures for Signal and Image Processing (DASIP)*, pages 1–8, 2012.
- [3] C. Chrysafis and A. Ortega. Line-based, reduced memory, wavelet image compression. *IEEE Transactions on Image Processing*, 9(3):378–389, 2000.
- [4] C. Chrysafis and A. Ortega. Minimum memory implementations of the lifting scheme. In *Proceedings of SPIE, Wavelet Applications in Signal and Image Processing VIII*, volume 4119 of *SPIE*, pages 313–324, 2000.
- [5] A. Cohen, I. Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 45(5):485–560, 1992.
- [6] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *Journal of Fourier Analysis and Applications*, 4(3):247–269, 1998.
- [7] R. Kutil. A single-loop approach to SIMD parallelization of 2-D wavelet lifting. In *Proceedings of the 14th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, pages 413–420, 2006.
- [8] R. Kutil, P. Eder, and M. Watzl. SIMD parallelization of common wavelet filters. In *Parallel Numerics '05*, pages 141–149, 2005.
- [9] S. Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way. With contributions from Gabriel Peyré*. Academic Press, 3 edition, 2009.
- [10] J. Sýkora, R. Bartosinski, L. Kohout, M. Daněk, and P. Honzík. Reducing instruction issue overheads in Application-Specific Vector Processors. In *Proceedings of the 15th Euromicro Conference on Digital System Design (DSD)*, DSD '12, pages 600–607, 2012.
- [11] J. Sýkora, L. Kohout, R. Bartosinski, L. Kafka, M. Daněk, and P. Honzík. The architecture and the technology characterization of an FPGA-based customizable Application-Specific Vector Processor. In *IEEE 15th International Symposium on Design and Diagnostics of Electronic Circuits Systems (DDECS)*, pages 62–67, 2012.

# Regularization parameter choice for total variation based image deringing algorithm

Andrey V. Nasonov, Andrey S. Krylov  
Laboratory of Mathematical Methods of Image Processing  
Department of Computational Mathematics and Cybernetics  
Lomonosov Moscow State University, Moscow, Russia  
{nasonov, kryl}@cs.msu.ru

## Abstract

The problem of finding regularization parameter for image deringing algorithm is considered. Different images from the ringing database with modelled ringing effect were processed by total variation based ringing suppression algorithm. Reference images were used to find deringing parameter values for every image. The functional dependence between deringing parameters and ringing level was found that can be used to perform no-reference image ringing suppression.

**Keywords:** *ringing effect, deringing, total variation, regularization parameter.*

## 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. Mathematical aspects of ringing effect are discussed in [1], [2].

Ringing can appear as a result of image compression, image upsampling and other applications. An example of frequency cut-off ringing effect can be seen in Fig.1.



**Figure 1:** An image with ringing effect.

One of the main problems of image deringing is to detect the presence of ringing effect. The most of existing ringing estimation algorithms are designed for specific problems like image deringing after JPEG or JPEG-2000 compression [3], [4], [5] and image deringing after a certain type of image processing [6].

Several approaches for ringing level estimation in general case were developed but most of them have limited use. In [7], 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, but the size of this neighborhood is fixed a priori. In [8], the presence of ringing effect is detected by comparing the directions of image gradients at different scales. The work [9] 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.

The paper [10] introduced a general case no-reference ringing estimation method using the ratio between weighted total variation values of edge profiles after Gaussian blur with different parameters.

Another problem is to estimate the necessary ringing suppression level. In our work, we consider the problem of parameter estimation for an efficient deringing algorithm described in [10]. This algorithm is based on projection of image onto the set of images with bounded total variation (TV). The TV was first used in image enhancement by Rudin, Osher and Fatemi [11] for image denoising. General relations between TV and ringing effect can be found in [12].

The method to choose the deringing parameter for TV based algorithm for the problem of image deringing after resampling is proposed in [13]. For the general case, the paper [10] suggests to choose the deringing parameter a posteriori by iteratively varying the deringing parameter until the estimated ringing level of the result image is equal to the target ringing level. This approach requires performing a number of deringing algorithm applications with different deringing parameters.

In this paper, we propose a method to choose the deringing parameter for the TV based deringing algorithm a priori using one application of the deringing algorithm with fixed regularization parameter.

## 2. RINGING ESTIMATION ALGORITHM

We use the ringing estimation algorithm introduced in [10]. The algorithm consists of the following steps:

1. Finding edges the most suitable for ringing analysis.
2. Performing edge width estimation and the extraction of edge profiles from the found edges.
3. Calculating the ringing estimation value for every edge profile.
4. Output the image ringing level as the median value of ringing estimation values for edge profiles.

The ringing estimation value for a one-dimensional edge profile  $f(x)$  centered in  $x = 0$  is defined as:

$$R_E = \frac{TV(f, kd, w_{ad})}{TV(f, d, w_{ad})}, \quad (1)$$

where

$$TV(f, \sigma, w) = WTV(f * G_\sigma, w),$$

$$WTV(g, w) = \int_{-\infty}^{+\infty} |g'(x)| w(x) dx.$$

Here  $f * G_\sigma$  is a convolution of  $f(x)$  with Gaussian filter with the radius  $\sigma$ . The weight function  $w_{ad}$  is equal to:

$$w_{ad} = e^{-\frac{x^2}{2(d)^2}}.$$

The parameter  $\alpha$  controls the number of considered ringing oscillations. We use the value  $\alpha = 3$ .

The value  $d$  is the estimated edge width [10].

The parameter  $k$  depends on  $\alpha$  and the noise level. We use the value  $k = 0.25$  that was obtained experimentally.

### 3. RINGING SUPPRESSION ALGORITHM

We use a projection method of an image into the set of images with bounded total variation to perform ringing suppression [10]:

$$z_q = \arg \min_{\|z\|_V \leq q \|z_0\|_V} \|z - z_0\|_2^2, \quad (2)$$

where  $z_0$  is the input image with ringing effect,  $\|z\|_V$  is the total variation functional:

$$\|z\|_V = \sum_{i,j} |z_{i+1,j} - z_{i,j}| + \sum_{i,j} |z_{i,j+1} - z_{i,j}|,$$

and  $q$  is the ringing suppression level — the ratio between the total variation of the result image and the source image.

Subgradient method and projection gradient method are used to minimize the target functional (2) [13].

### 4. FINDING OPTIMAL DERINGING PARAMETERS FOR IMAGES FROM THE RINGING DATABASE

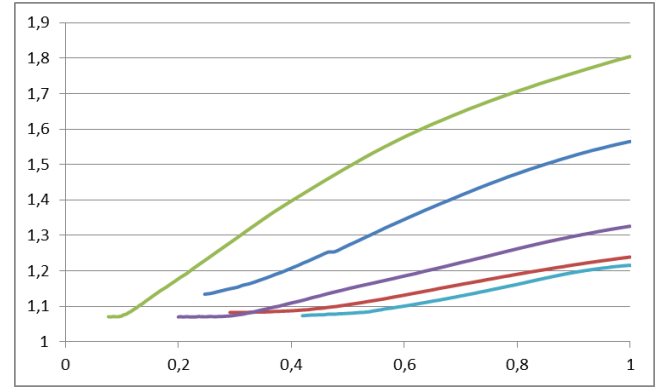
We have developed the special ringing database containing 35 reference images with ringing effect modeled by different methods [14]. The reference images are real images that contain nature and architecture objects. Ringing effect is modeled by interpolation, sharpening and low-pass filtering and compression algorithms.

For every image in the database we perform a series of applications of deringing algorithm (2) with  $q$  in the range of  $[0, 1]$  and calculate ringing level values  $R_E$ .

Experiments has shown that the functional dependence

$$f(q) = R_E(z_q), \quad (3)$$

is close to linear function in  $[0.5, 1]$  range. Examples of  $f(q)$  functions for different input images are shown in Fig. 2.



**Figure 2:** Examples of  $f(q)$  functions (3) for different input images.

We estimate the function  $f(q)$  with a linear function

$$g(q) = 2(f(0.5)(1 - q) + f(1)(q - 0.5))$$

and calculate the maximum error

$$\max |f(q) - g(q)|, \quad q \in [0.5, 1].$$

The average of maximum errors was found as 0.02 while the standard deviation was 0.015 on the set of 900 images from the database [14] that do need deringing.

It makes possible to use the estimation  $g(q)$  instead of  $f(q)$  to choose the deringing parameter. The image deringing algorithm is applied twice: first with  $q = 0.5$ , then with interpolated  $q$ . To achieve more accurate result, the secant method can be used.

### 5. RESULTS

The proposed method of finding deringing parameter is demonstrated in Fig. **Figure 33** for target  $R_E = 1.2$ .

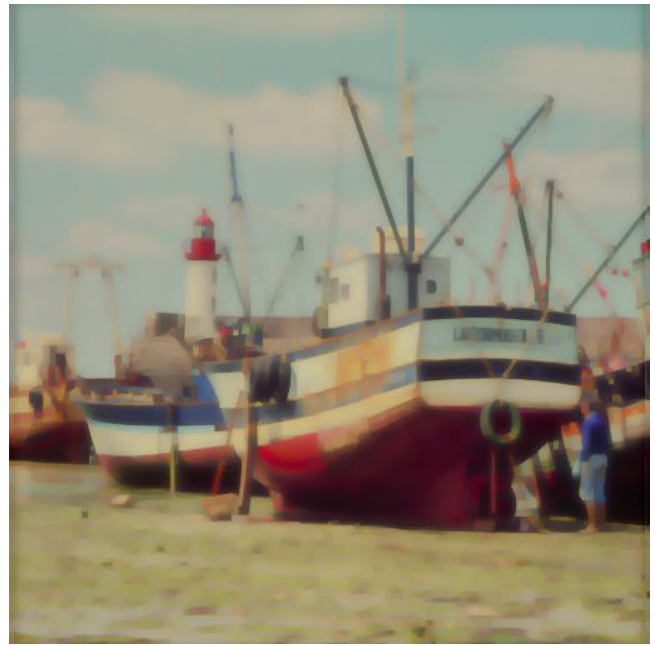
The ringing value for the input image is 1.388, the ringing value after deringing with  $q = 0.5$  is 1.124. To achieve ringing level equal to 1.2, we choose  $q = 0.654$ . The ringing level of the obtained image is 1.204, the MSE and SSIM values are significantly better than for the input image.





a) The reference image from the database.

$$R_E = 1.286$$



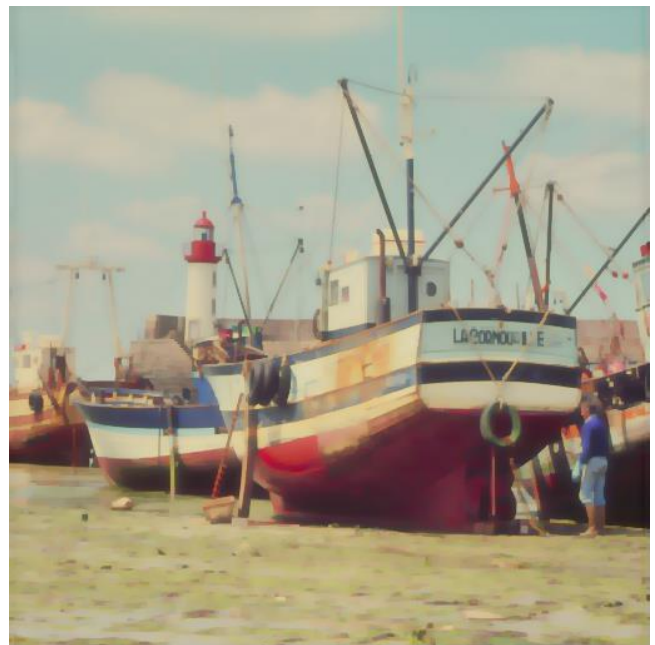
c) Application of the deringing algorithm with  $q = 0.5$ .

$$R_E = 1.124, MSE = 567, SSIM = 0.954$$



b) The image with modeled ringing effect.

$$R_E = 1.388, MSE = 196, SSIM = 0.955$$



d) Application of the deringing algorithm with  $q = 0.654$ .

$$R_E = 1.204, MSE = 65, SSIM = 0.984.$$

**Figure 3:** Application of the proposed method for image deringing.

4 images: reference, with ringing, after preliminary ringing reduction, final ringing reduction with target ringing level.

## 6. CONCLUSION

The experimental analysis of finding the ringing suppression parameter for no-reference ringing effect reduction has been performed. Nearly linear dependence of ringing level on the deringing parameter has been shown. A method to choose the deringing parameter according to the target ringing level has been proposed.

The work was supported by RFBR grant 13-07-00438\_a.

## 7. REFERENCES

- [1] David Gottlieb, Chi-Wang Shu, "On the Gibbs Phenomenon and Its Resolution" // *SIAM review*, Vol. 39, No. 4, 1997, pp. 644–668.
- [2] E. Tadmor "Filters, mollifiers and the computation of the Gibbs phenomenon" // *Acta Numerica*, Vol. 16, 2007, pp. 305–378.
- [3] Amal Punchihewa, Alexander Keerl, "Test pattern based evaluation of ringing and blur in JPEG and JPEG2000 compressed images" // *2010 4th International Conference on Signal Processing and Communication Systems (ICSPCS)*, 2010, pp. 1–7.
- [4] Quoc Bao Do, M. Luong, A. Beghdadi, "A new perceptually adaptive method for deblocking and deringing" // *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*, 2010, pp. 533–538.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] Hantao Liu, Nick Klomp, and Ingrid Heynderickx, "A No-Reference Metric for Perceived Ringing Artifacts in Images" // *IEEE Transactions on Circuits and Systems For Video Technology*, Vol 20, No 20, 2010, pp. 529–539.
- [10] A. V. Nasonov, A. S. Krylov. "Adaptive Image Deringing" // *In: 19-th International Conference on Computer Graphics GraphiCon'2009*. Moscow, Russia, 2009, pp. 151–154.
- [11] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms" // *Physica D*, vol. 60, pp. 259–268, 1992.
- [12] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1999.
- [13] A. S. Krylov and A. V. Nasonov, "Adaptive total variation deringing method for image interpolation" // *Proceedings of ICIP'08*, pp. 2608–2611, 2008.
- [14] <http://imaging.cs.msu.ru/research/ringing/database>
- [15] 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.

## About the author

Andrey Krylov is a professor, head of the Laboratory of Mathematical Methods of Image Processing, Department of Computational Mathematics and Cybernetics, Moscow Lomonosov State University. His contact email is [kryl@cs.msu.ru](mailto:kryl@cs.msu.ru)

Andrey Nasonov is a senior scientist at the Laboratory of Mathematical Methods of Image Processing, Department of Computational Mathematics and Cybernetics, Moscow Lomonosov State University. His contact email is [nasonov@cs.msu.ru](mailto:nasonov@cs.msu.ru)

# Curve-skeleton extraction using silhouettes' medial axes

Andrey Zimovnov, Leonid Mestetskiy  
Department of Computational Mathematics and Cybernetics  
Moscow State University, Moscow, Russia  
zimovnov@gmail.com, mestlm@mail.ru

## Abstract

A new method of curve-skeleton extraction is proposed in this paper. Unlike most methods this approach retrieves information from a polygonal mesh via rasterization and extraction of silhouettes' continuous medial axes. Described method shows a great improvement in computational time comparing to existing methods. Method shows good extraction results for models with complex geometry and topology. Resulting curve-skeletons conform with most requirements to universal curve-skeletons.

**Keywords:** curve-skeleton, medial axis, visual hull, distance transform, mean-shift

## 1 Introduction

Curve-skeleton is a graph that depicts simplified versions of object's geometry and topology. An inscribed sphere is associated with every node of curve-skeleton. The envelope of all curve-skeleton spheres approximates the shape of the object. Curve-skeletons find many applications in problems where object's shape analysis is needed, such as object recognition, shape classification, object skeletal animation, object segmentation, finding visually similar objects in databases and others [Cornea et al. 2005].

Most existing methods analyze polygons or voxels in space [Cornea et al. 2005], which leads to slow extraction process. In paper [Mestetskiy and Tsiskaridze 2009] authors utilize the observation that on a projection of object without occlusions silhouette's medial axis is the projection of curve-skeleton's bones. Although object's occlusions lead to spurious curves in resulting skeleton, experiments [Livesu et al. 2012] show that even with occlusions projections store enough information for curve-skeleton extraction.

In this paper a new approach to curve-skeleton extraction is proposed. Our method uses continuous medial axes extraction from object's silhouettes [Mestetskiy and Semenov 2008] to organize an efficient iterative contraction process similar to [Au et al. 2008].

## 2 Proposed method

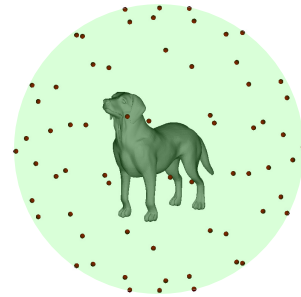
In this paper we assume that object is represented by a polygonal mesh. Proposed method is an iterative process. One iteration can be divided into the following steps:

- Curve-skeleton approximation with point cloud reconstructed from silhouettes' medial axes.
- Object contraction based on inscribed spheres radii reduction associated with every point of the approximating cloud.

### 2.1 Curve-skeleton approximation with point cloud

Proposed method is based on the observation that on a projection of object without occlusions silhouette's medial axis is the projection of curve-skeleton's bones. Assuming there're no occlusions on a projection, curve-skeleton can be reconstructed by medial axis back-projecting. To eliminate the influence of object's occlusions

many projections are used. The main idea is to back-project object's parts that are visible without occlusions and filter out spurious occluded parts. Orthogonal projections are captured from uniformly distributed cameras on a sphere around the object (Fig. 1). Uniformly distributed points on a sphere are generated using hexahedron subdivision. Filtered parts acquired from different projections are combined into one graph and aligned.



**Figure 1:** Uniformly distributed cameras on a sphere.

To sum up, the approximation process can be described with the following steps:

- Back-projection of silhouettes' medial axes
- Nodes and bones filtering in space
- Alignment of skeleton nodes and bones from different projections.

#### 2.1.1 Back-projection of silhouettes' medial axes



**Figure 2:** Object's projection, silhouette and medial axis.

For each silhouette medial axis is a continuous graph [Mestetskiy and Semenov 2008] with nodes and bones (Fig. 2). In order to back-project the node  $v_i$  of medial axis a ray is casted from the node towards camera view and two intersections with object are saved:  $x_1^{(i)}$  — the first intersection with front-facing polygon and  $x_2^{(i)}$  — the first intersection with back-facing polygon (Fig. 3). Calculation is done on GPU via depth buffer and culling, so there's no need to find intersections analytically.

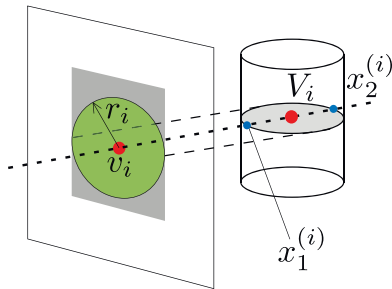
Now the depth of curve-skeleton node  $V_i$  can be estimated as

$\frac{1}{2}[x_1^{(i)} + x_2^{(i)}]$ . Such depth calculation helps to back-project the nearest to the camera parts of the object even with heavy occlusions.

The maximum inscribed ball radius  $R_i$  with the center in  $V_i$  can be estimated as follows:

$$R_i = \min \left[ r_i, \frac{|x_1^{(i)} - x_2^{(i)}|}{2} \right],$$

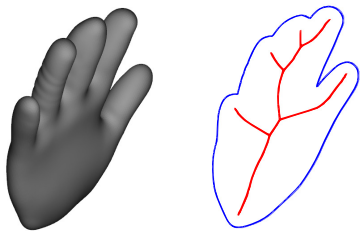
where  $r_i$  is the radius of maximum inscribed circle with the center in  $v_i$ . Such estimation is robust due to implicit measuring of radius  $R_i$  in two orthogonal directions.



**Figure 3:** Back-projection of the node  $v_i$  of medial axis in order to get curve-skeleton node  $V_i$ .

### 2.1.2 Nodes and bones filtering in space

Occlusions in silhouettes produce a huge number of spurious nodes and bones (Fig. 4).



**Figure 4:** Projection with occlusions and its medial axis. Medial axis doesn't represent object geometry well due to occlusions.

**Nodes filtering** The basic idea of nodes filtering comes from curve-skeleton definition: this graph should reproduce object's shape with the set of maximum inscribed balls in its nodes. Thereby a node is considered spurious if a corresponding inscribed ball is not inside the object. For efficient nodes filtering Visual Hull is used: the object is approximated with an intersection of prisms with object's silhouettes in their bases. A strict definition of Visual Hull (VH) is as follows:

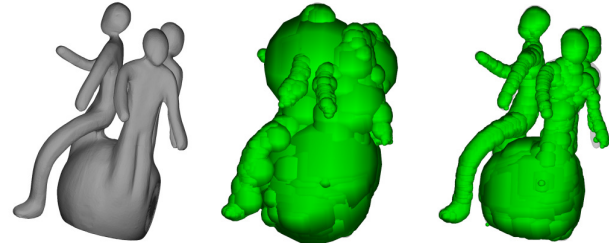
$$VH = \{x \in \mathbb{R}^3 : \forall i \ Pr_i(x) \in S_i\},$$

where  $\{S_i\}$  — set of object's silhouettes,  $Pr_i(x)$  — orthogonal projector of point  $x \in \mathbb{R}^3$  on the plane of silhouette  $S_i$ .

To check whether a sphere is inside the Visual Hull, Distance Transforms (DT) of silhouettes are computed [Felzenszwalb and Huttenlocher 2004]. Thus to check a sphere on a single silhouette we need to compare its radius with the minimum distance from its projected

center to the silhouette border acquired with DT. If the sphere radius is less or equal to DT values throughout all silhouettes from  $\{S_i\}$ , sphere is inside VH and the corresponding node is considered relevant. Otherwise, the node is deleted from resulting graph.

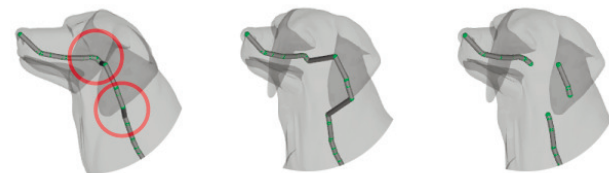
The resulting approximation of object with inscribed balls is much better after filtering (Fig. 5).



**Figure 5:** An object, inscribed balls from step 2.1.1 and filtered inscribed balls.

**Bones filtering** Occlusions lead to spurious bones after back-projecting. These bones usually stand out of the model due to occlusions of object's parts. Suggested method deletes bones that have small angle ( $\leq \phi_0$ ) with corresponding camera view direction. This approach might delete relevant curves, but due to usage of the set of projections this curve will likely be captured from another projection.

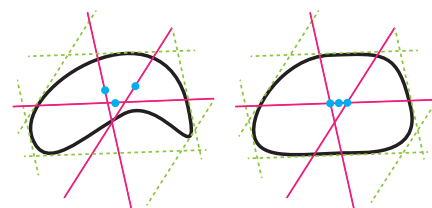
The example of filtering result is shown in Fig. 6.



**Figure 6:** Skeleton back-projection, back-projection from different view, filtered bones.

### 2.1.3 Alignment of skeleton nodes and bones from different projections

Even without occlusions same part of the object on different projections can be back-projected into centered, but quite different curves. This happens when the object's cut is not a circle (Fig. 7). We suggest to organize an alignment process to join nodes and bones from different projections and to remove such noise.



**Figure 7:** Inconsistent back-projecting throughout different projections due to the concavity of the object's cut or its elliptic shape.

At this stage nodes and bones are centered in the model and look like a very noisy curve-skeleton. The basic idea behind alignment is noise canceling by finding resulting nodes with the highest density of neighbors. For this purpose Mean-Shift algorithm is used, that iteratively moves every node from starting position to the average position of its neighbors. When the node stops to significantly change its position, the node is replaced with the final average. The algorithm doesn't take into account the skeleton bones, that are automatically shifted with adjacent nodes.

The following algorithm describes this approach:

**Require:**  $X$  — input nodes,  $\varepsilon$  — minimum shift threshold,  $\alpha$  — minimum density of neighbors threshold,  $K(x)$  — kernel function;

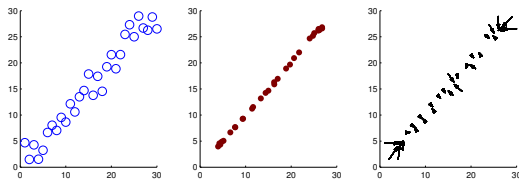
**Ensure:**  $Y$  — shifted nodes;

```

1:  $Y := \emptyset$ ;
2: for  $x_i \in X$  do
3:    $m := x_i$ ; {start Mean-Shift from  $x_i$ }
4:   repeat
5:      $m_{old} := m$ ;
6:      $m := \frac{\sum_{x_j \in X} K(x_j - m_{old})x_j}{\sum_{x_j \in X} K(x_j - m_{old})}$ ;
7:   until  $\|m - m_{old}\| \geq \varepsilon$ 
8:   if  $\sum_{x_j \in X} K(x_j - m) > \alpha$  then
9:      $Y := Y \cup \{m\}$ ;
10:  end if
11: end for

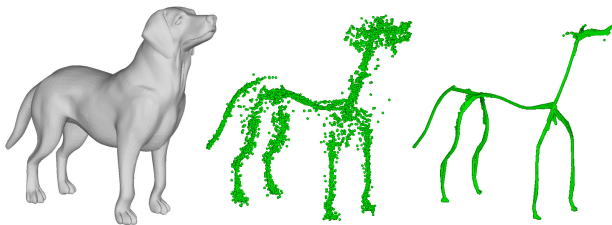
```

The alignment algorithm is illustrated in case of points on plane in Fig. 8. There's an obvious drawback of such approach: the ending points of the line travel too far. We suggest to use a heuristic that forbids ending points of medial axis (which is a graph) to travel too far.



**Figure 8:** The alignment algorithm for points on plane. From left to right: set of points, aligned points, travel paths.

The result of such alignment is shown in Fig. 9. The result is much cleaner and looks like a good curve-skeleton approximation.



**Figure 9:** An object, back-projecting and filtering result, aligned nodes and bones.

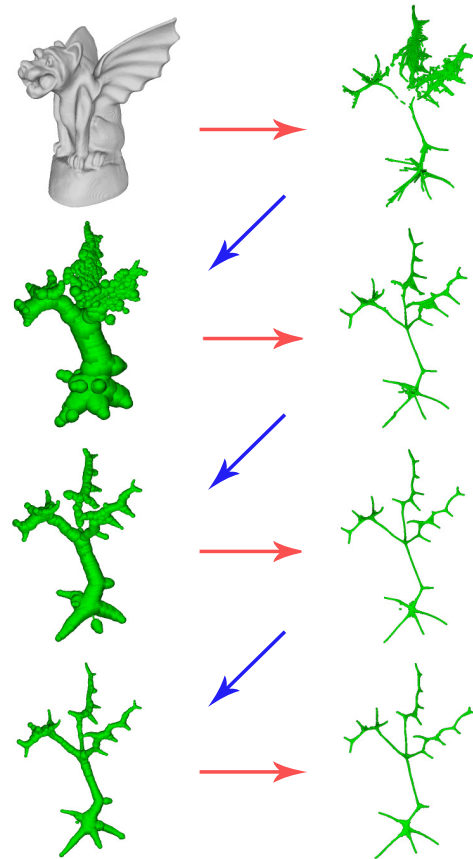
## 2.2 Iterative contraction

We suggest to implement object contraction through spheres of curve-skeleton approximation radii reduction. Point cloud from section 2.1 approximates curve-skeleton and thus is centered and describes shape quite well, hence spheres radii reduction leads to visually correct object contraction (Fig. 10).



**Figure 10:** Object contraction through spheres radii reduction.

Contraction makes the object thinner, which helps to get rid of occlusions, thus making point cloud approximation better through iterations. Fig. 11 illustrates 4 iterations of object contraction. We can see that point cloud approximation of curve-skeleton is sufficient for object contraction.



**Figure 11:** 4 iterations of object contraction. Red arrows depict curve-skeleton approximation with point cloud, blue arrows depict object contraction based on approximation.

### 3 Experiments

Experiments were conducted on the computer with Core i5-3570K 3.4 GHz CPU, 8 GB RAM, Intel HD Graphics 4000 GPU. For continuous skeleton extraction library [Mestetskiy and Semenov 2008] was used.

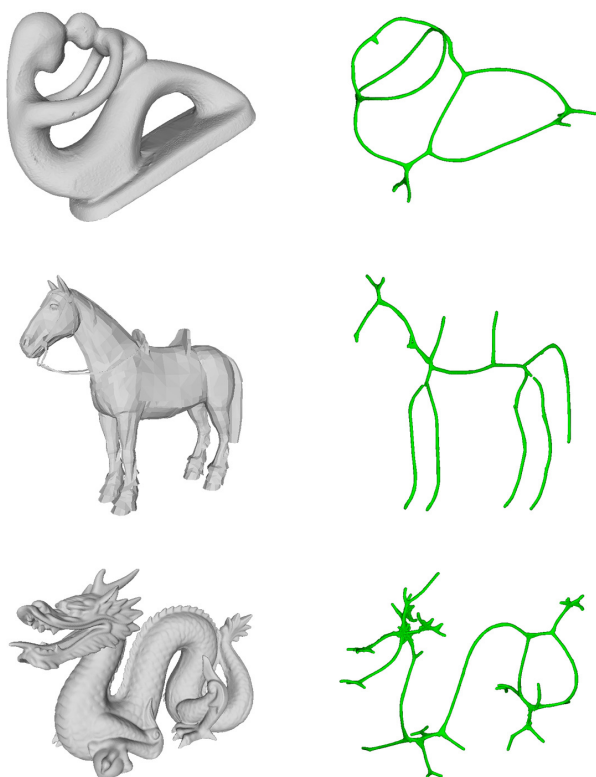
Method used 66 projections with resolution  $300 \times 300$ . Bones filter was used with  $\phi_0 = 45^\circ$ . Alignment of nodes and bones was used with  $\varepsilon$  equal to the height of one pixel in model coordinates,  $\alpha$  equal to 5%–10% percentile, kernel function  $K(x) = \exp(-C\|x\|^2)$  with  $C = 100$ .

Table 1 shows timings in milliseconds for different stages of algorithm for different models. Due to the fact that method doesn't deal with polygons in space, model complexity (number of polygons) has little effect on extraction time.

**Table 1:** Timings in milliseconds for different stages of algorithm.

Model	Faces in model	Medial axes	Back-project	Filters	Mean-Shift	Total
Fertility	50,000	129	405	153	484	1,171
Memento	52,550	152	388	145	957	1,642
Horse	7,951	144	328	185	1,193	1,850
Elk	48,026	160	381	195	602	1,338

Fig. 12 shows curve-skeletons for different models. The resulting curve-skeletons depict models' geometry and topology well.



**Figure 12:** Examples of curve-skeletons extracted with described method.

### 4 Method analysis

Resulting curve-skeletons conform with most requirements to universal curve-skeletons based on research [Cornea et al. 2005]. This is achieved using filters, mean-shift and inherited properties of medial axes like centeredness and isometric invariance.

Compared to [Au et al. 2008] method shows 6 times increase in extraction speed (in average 3 sec. vs 19 sec. for different models), though it's hard to compare the quality of resulting skeletons due to poor formalization of the problem [Dey and Sun 2006].

Method looks ideal for parallel implementation as most stages can be done on GPU, which promises a huge decrease in computational time and possibly new real-time applications for curve-skeletons.

Method fails to extract curve-skeletons for heavily occluded objects with parts not visible without occlusions. Fortunately, most applications of curve-skeletons assume that we have a character-like object with significant visual branches.

### 5 Conclusion

A new method of curve-skeleton extraction is proposed in this paper. Unlike most methods this approach retrieves information from a polygonal mesh via rasterization and extraction of silhouettes' continuous medial axes. The usage of continuous medial axes reduces computational time and gives an ability to analyze medial axis as a graph. Described method shows good extraction results for models with complex geometry and topology.

### Acknowledgements

The work was partially supported by RFBR, research project No. 12-07-31107-mol\_a.

### References

- AU, O. K.-C., TAI, C.-L., CHU, H.-K., COHEN-OR, D., AND LEE, T.-Y. 2008. Skeleton extraction by mesh contraction. In *ACM SIGGRAPH 2008 papers*, ACM, New York, NY, USA, SIGGRAPH '08, 44:1–44:10.
- CORNEA, N. D., SILVER, D., AND MIN, P. 2005. Curve-skeleton applications. In *IEEE Visualization*, IEEE Computer Society, 13.
- DEY, T. K., AND SUN, J. 2006. Defining and computing curve-skeletons with medial geodesic function. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SGP '06, 143–152.
- FELZENSZWALB, P. F., AND HUTTENLOCHER, D. P. 2004. Distance transforms of sampled functions. Tech. rep., Cornell Computing and Information Science.
- LIVESU, M., GUGGERI, F., AND SCATENI, R. 2012. Reconstructing the curve-skeletons of 3d shapes using the visual hull. *IEEE Trans. Vis. Comput. Graph.* 18, 11, 1891–1901.
- MESTETSKIY, L., AND SEMENOV, A. 2008. Binary image skeleton - continuous approach. In *VISAPP (1)*, INSTICC - Institute for Systems and Technologies of Information, Control and Communication, A. Ranchordas and H. Araújo, Eds., 251–258.
- MESTETSKIY, L., AND TSISKARIDZE, A. 2009. Spatial reconstruction of locally symmetric objects based on stereo mate images. In *VISAPP (1)*, INSTICC Press, A. Ranchordas and H. Araújo, Eds., 443–448.

# Quantum Hough Transform

Karthik Mahesh Varadarajan  
Technical University of Vienna  
[kv@acin.tuwien.ac.at](mailto:kv@acin.tuwien.ac.at)

## Abstract

With the rapid growth of quantum computing, several quantum algorithms are being designed for a variety of applications. However, the application of quantum algorithms to image processing and computer vision has been rather restricted. The Quantum Fourier Transform (QFT) is one such algorithm that is used in applications such as discrete logarithms and period finding that posit employment in image processing, but cannot be used for traditional applications such as spectrum estimation and filtering. In this paper, we present a novel quantum version of the popular Hough Transform and show the advantages of the Quantum Hough Transform in terms of the computational and space requirements over the traditional formulation. Specifically, we transform accumulation in Hough space to implicit quantum superposition resulting in gains in space and speed requirements.

**Keywords:** *Quantum algorithms, Hough Transform, Quantum superposition, Quantum computing.*

## 1. INTRODUCTION

Quantum computers offer the possibility of enhanced computing speeds for several types of computational problems that are inherently hard to solve on traditional computers. With recent developments in the field of theoretical quantum computing, efforts have also been made to identify algorithms designed for conventional computing that are particularly well-suited for implementation on quantum computers. Some popular conventional algorithms that have found quantum computing implementations include prime factorization (Shor's algorithm [1]), index search (Grover's algorithm [2]), periodicity detection in modulo arithmetic (Simon's algorithm [3]) and Quantum Fourier Transform [4]. These algorithms have been largely restricted to the domain of traditional artificial intelligence with few applications in the domain of image processing and computer vision. Quantum Fourier Transform (QFT) for example is used in the for Period Finding, Discrete Logarithms, Order Finding, Quantum Counting, Hidden Sub-Group problems, Encryption Decoding, but is not applicable to traditional fields of application of Discrete Fourier Transform (DFT) such as Spectrum Analysis, Filtering, Compression etc. This is because, quantum algorithms do not work directly in the space of the input vectors but on quantum states and the resulting outputs from the application of quantum algorithms are also quantum states with the only observable or measurable quantity being the state of the system and not its amplitude. For example, in the case of QFT, the measurable quantity is the quantum state in the encoded quantum fourier basis, while the amplitude of the frequency component can only be measured as the squared probability of occurrence of this state. Thus, it can be seen that it is rather difficult to build quantum versions of conventional algorithms and even in cases this is possible, the output is in a state that cannot be directly used in the traditional application of interest. However, it has been observed that quantum algorithms (in cases where the output can be directly used in the application of interest) provide orders of

speed-up that are not possible with conventional algorithms. The most popular quantum algorithms operate in polynomial time whereas the corresponding conventional algorithms operate in exponential time, thereby resulting in an exponential order of speed-up. Much of the speed-up in the case of quantum algorithms is attributed to prior knowledge about the structure of the problem being used [10]. The accumulation of histories in the case of a classical problem to as much as 50% of the information required to solve a problem serves as a significant factor in the quantum speed-up.

Two specific sub-fields of Quantum Computing are also relevant from the standpoint of computer vision. The first of these is Quantum Signal Processing (QSP) and the second is composed of quantum systems for optimization of object recognition problems. Besides these two sub-fields, there have been a few algorithms that draw inspiration from quantum state modeling towards solving conventional image processing and computer vision problems. Notable among these include the use of quantum interference analogies in mutation for genetic programming for image registration [5], use of qubit like structures called Conjugate Information Variables that have non-isotropic sensitivity regions employable for classification of rising, falling and flat regions of functions (or images) and hence be used for edge-sensitive adaptive filtering [6], quantum representations for 4-color channel images (RGB-A) [7], quantum representations for joint color image encoding and image structure description for recognition [8] and the use of quantized states (though not quantum states) for Total Variation (TV) in a Mumford-Shah like energy formulation for image segmentation and denoising [9].

The field of QSP [11] is aimed at developing new or modifying existing algorithms by applying some of the principles of quantum mechanics. Hence these are not true quantum algorithms that can be implemented on a quantum computer, but rather use principles that are borrowed from quantum literature. Hence, the algorithms developed using QSP are not limited by the constraints of quantum mechanics. Some of the original QSP applications include frame theory, quantization and sampling methods, compressed sensing matched filters, subspace coding, detection, parameter estimation, covariance shaping and multi-user wireless communication systems. These QSP algorithms use a Quantum Input Mapping followed by the application of either Rank-One Measurements (ROM) or Subspace Measurements, again followed by an Output Mapping in order to achieve various required tasks. QSPs have been used in image processing, recently, for a number of tasks such as image half-toning, edge detection, cryptography - by mapping binary images into the  $|0\rangle$  and  $|1\rangle$  states of the quantum system and estimating the probabilities of these two states as  $|p_0|^2$  and  $|p_1|^2$  using an exponential function of the variables measured from the image intensity values [12]. Such a formulation has been used for structural element description for noise filtering in [13]. This has also been used in relation to neighborhood pixel intensities for defining features, corners and subsequently edges in [14].

The second sub-field includes quantum algorithms for optimization of object recognition problems. Primary among these

algorithms for image feature matching include those formulated as Quadratic Unconstrained Binary Optimization (QUBO) and solved using D-Wave Superconducting Adiabatic Quantum Computer (ADC) with the theory developed in [15,16,17] and hardware implementations demonstrated in [18,19]. Another “true” quantum algorithm that is focused on object recognition and de-noising that works on quantum states at all stages of the processing is [20].

## 2. ALGORITHM

In this paper, we further the scope of application of quantum algorithms to image processing and computer vision by creating a quantum version of the Hough Transform [22], the conventional version of which has been used for numerous applications such as edge detection/selection and object template matching. It should be noted here that the algorithm proposed in this paper – Quantum Hough Transform or QHT is a “true quantum algorithm” that can be implemented on quantum computers as opposed to QSP which subsumes quantum inspired algorithms. Furthermore, using quantum complexity analysis, we demonstrate that time and space requirements of QHT are several orders of magnitude smaller than that required for traditional Hough transform (HT) implementations, thereby demonstrating the need and benefits of QHT over traditional HT.

### 2.1 Hough Transform (HT)

A popular algorithm used in image processing for detection of features such as edges, curves, circles, contours, planes, cylinders and even entire shapes is the Hough Transform. The Generalized Hough Transform (GHT) [23] in particular is extremely useful in template matching of arbitrary objects described using models. In the case of GHT, the problem is transformed to one of determination of spatial coordinates of the shape template being matched in the image of interest. An edge based implementation of GHT was first proposed by measuring the distance from points in the shape. While HT can be used for a variety of purposes, with the classical HT being used for analytic functions, the simplest variant of the HT which is used for the detection of edges in images remains the most popular and widely used application of HT. All variants of HT use a voting procedure in parameter space called the accumulator space from which object candidate are localized by extraction of local maxima. Since an accumulator has to be maintained across the entire parameter space of the function, HT typically takes a great amount buffer space for implementation. For example, for the task of finding edges in an image, wherein the HT uses the parametrization  $(r, \theta)$ , the space complexity of the HT is given by  $O(n_r \cdot n_\theta)$ , where  $n_r$  is the number of bins along the  $r$  parameter space and  $n_\theta$  is the number of bins along the  $\theta$  parameter space. Given that the range of  $r$  spans the space from 0 to  $\sqrt{l^2 + w^2}$ , where  $l$  and  $w$  are the image dimensions and  $\theta$  spans the entire range of angles from 0 to 360, it can be seen that even for a simple 2 parameter accumulator space, the computational requirements can be quite high depending on the necessary resolution. While variants of Hough Transform exist that attempt at limiting the computational complexity by reducing the size of the accumulator search space or by using additional information from the image space such as image gradients to coarsely determine the HT parameters, we focus our attention in this paper on the simplest form of HT operating in the full accumulator space since most such optimization modifications are also applicable to the HT estimation method presented in this paper.

### 2.2 Quantum Representation of Hough Transform Space

The first transformation required to build the quantum algorithm is the transformation of the input pixel co-ordinate space  $p(x, y)$  into the one dimensional complex vector space  $\mathbb{C}^1$ , also called the  $z$ -plane. Hence,

$$p(x, y) = (p_x, p_y) \Rightarrow p_x + p_y i$$

In order to enable the QHT to maintain the same ranges as that of HT, we use a discretized and bounded version of the complex space restricted in range to that of the original image space. Hence,

$$p_x \in \{1 \dots l\}, \quad p_y \in \{1 \dots w\}$$

Using the parametrization  $(r, \theta)$ , where  $r$  represents the shortest distance from the origin to the line and  $\theta$  represents the angle that the vector from the origin to the line (at its closest point) makes with the first coordinate axis, for the Hough lines to be determined, the original formulation for the Hough lines

$$y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{r}{\sin\theta}\right)$$

can be rewritten as

$$r(\theta) = x \cos\theta + y \sin\theta$$

For an arbitrary point  $(x_0, y_0)$  on the line, the value of  $r(\theta)$  is given by

$$r(\theta) = x_0 \cos\theta + y_0 \sin\theta$$

Hence, the value of  $r(\theta)$  from arbitrary pixels in the image containing the line can be expected to be roughly constant for a range of  $\theta$  values -  $\theta_k$ , typically from 0 degrees to 180 degrees, where  $k \in \{k_{min} = 0, \dots, k_{max}\}$ . Hence, the required values of  $r$  and  $\theta$  corresponding to the parameterization of a single Hough line can be determined as

$$\max_k r(\theta)$$

The analogous equations in the complex plane representation are given by

$$\begin{aligned} r(\theta) &= x_0 \cos\theta + y_0 \sin\theta \\ &= x_0 \cos\theta - i^2 y_0 \sin\theta \\ &= x_0 \cdot \cos\theta - (i \cdot y_0) \cdot (i \cdot \sin\theta) \\ &= p_{x0} \cdot \cos\theta - p_{y0} \cdot (i \cdot \sin\theta) \\ &= p_{x0} \cdot \cos(-\theta) + p_{y0} \cdot (i \cdot \sin(-\theta)) \\ &= p_0 \cdot e^{-i\theta} \end{aligned}$$

Representing the above equation using a dot product and introducing the matrix notation, along with the range subscript  $k$ , we have

$$r(\theta_k) = [p_{x0} \ p_{y0}] \begin{bmatrix} \cos(-\theta_k) \\ \sin(-\theta_k) \end{bmatrix}$$

where  $\begin{bmatrix} \cos(-\theta_k) \\ \sin(-\theta_k) \end{bmatrix}$  corresponds to  $e^{-i\theta_k}$ .

Since  $\theta_k$  typically ranges from 0 degrees to 180 degrees (or 0 to  $\pi$ ) in incremental steps based on  $k_{max}$ , we have

$$\theta_k = \frac{2\pi k}{2k_{max}}$$

Hence,



$$\begin{aligned} r(\theta_k) &= p_0 \cdot e^{-i\frac{2\pi k}{2k_{max}}} \\ &= p_0 \cdot \omega^{-\frac{k}{2}} \end{aligned}$$

where,

$$\omega = e^{\frac{i2\pi}{k_{max}}}$$

is the  $k_{max}$ <sup>th</sup> root of unity.

Given a binary edge image of size  $l \cdot w$  pixels with values  $I_{x_0, y_0}$  at the pixel  $p_0(x_0, y_0)$ , the pixel indices corresponding to an image intensity value of 1 can be linearized into the following complex vector

$$P = \begin{bmatrix} p_0 \\ p_1 \\ \dots \\ p_N \end{bmatrix}$$

where  $I_{p_m} = 1, m \in \{1, \dots, N\}$ .

A conventional HT requires an accumulator space spanning the entire range of  $\theta$  and  $r$  in order to compute the Hough Transform. However, with the QHT, we simulate the accumulation process for finding the peak in the  $(r, \theta)$  space through quantum superposition. In order to describe the problem in a format suitable for quantum superposition, we break down the accumulator into two sub-matrix representations.

The first sub-matrix representation (QHT1) involves the estimation of accumulated  $r$  for various values of  $\theta$ .

$$\begin{bmatrix} p_0 \\ p_1 \\ \dots \\ p_N \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & \dots & 1 \\ \omega^{-\frac{1}{2}} & \omega^{-\frac{1}{2}} & \dots & \omega^{-\frac{1}{2}} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \omega^{-\frac{k_{max}}{2}} & \omega^{-\frac{k_{max}}{2}} & \dots & \omega^{-\frac{k_{max}}{2}} \end{bmatrix} = \begin{bmatrix} r_0 \\ r_1 \\ \dots \\ r_{k_{max}} \end{bmatrix}$$

Simultaneously, the following outer-product (QHT2) or projection matrix is also computed using the partial product terms from the above matrix computation.

$$\begin{bmatrix} p_0 \\ p_1 \\ \dots \\ p_N \end{bmatrix} \cdot \begin{bmatrix} 1 \\ \omega^{-\frac{1}{2}} \\ \dots \\ \omega^{-\frac{k_{max}}{2}} \end{bmatrix}^T = \begin{bmatrix} r_{00} & r_{10} & \dots & r_{N0} \\ r_{01} & r_{11} & \dots & r_{N1} \\ \dots & \dots & \dots & \dots \\ r_{0k_{max}} & r_{1k_{max}} & \dots & r_{Nk_{max}} \end{bmatrix}$$

Now, representing the input complex pixel space vector  $p_m$  in the equivalent quantum superposition state, we have

$$\sum_{i=0}^N p_i |i\rangle$$

where  $|i\rangle = |0\rangle \dots |N\rangle$  are orthonormal basis vectors. Using the above representation, the first sub-matrix representation yields

$$|i\rangle \rightarrow \sum_{k=0}^{k_{max}} r_k |k\rangle$$

which is a quantum superposition of the basis vectors in  $|k\rangle$ . It should however be noted that the output of QHT1 cannot be measured directly in the space of  $r_k$ . In other words, the magnitude of  $r_k$  cannot be calculated due to the nature of quantum states. The only output that can be measured is the state  $|k\rangle$  itself (index) and the probability of measurement of this state  $|k\rangle$  is given by  $|r_k|^2$ . For the case of the HT, this is not a disadvantage, but rather a useful phenomenon. While, the use of QFT is restricted in the case of measurement of spectra due to the inability to measure the magnitude of frequency components as a

result of this phenomenon, this constraint is not a drawback in the case of QHT, since the goal of QHT is the estimation of  $|k\rangle$  or  $k$  and  $r$  that maximize  $r_k$ . Since the values of  $k$  that maximize  $r_k$  are observed during quantum measurement with a very high probability, this enables the determination of the Hough peak. Furthermore, by performing several measurements, multiple (possible) candidate Hough peaks can be estimated. This also enables the estimation of multiple line segments in an image through the estimation of several Hough peaks. Thus, quantum superposition inherently acts as an accumulator, preserving the quantum states, without the need for an explicit one that demands high storage and computation requirements.

The second transformation QHT2 transforms the input quantum state as follows

$$|i\rangle \rightarrow \begin{bmatrix} \sum_{j=0}^N r_{j0} |j0\rangle \\ \sum_{j=0}^N r_{j1} |j1\rangle \\ \dots \\ \sum_{j=0}^N r_{jk_{max}} |jk_{max}\rangle \end{bmatrix}$$

Once the candidate values of  $k$ , corresponding to the Hough peaks are determined from QHT1, the appropriate quantum state vectors can be selected using the measured  $|k\rangle$  states. The selected superposition state(s) vector,

$$\sum_{j=0}^N r_{jk_{sel}} |jk_{sel}\rangle$$

is then used as input to the Quantum Counting algorithm [21]. Parameters for discretizing the  $r$  space can also be directly incorporated into the counting algorithm. This algorithm counts the instances of each selected  $r_{jk_{sel}}$  up to a threshold and also returns the indices of  $j$ , thus not just enabling the detection of  $r$  corresponding to the Hough peaks, but also the pixels in the image that correspond to the Hough line. Thus, detection of  $r, \theta$  as well pixel indices corresponding to the Hough Transform are computed using the Quantum formulation.

### 2.3 Complexity Analysis

It should be noted that the Quantum Counting algorithm uses only  $\Theta\left(\frac{1}{\epsilon} \sqrt{\frac{N}{t}}\right)$ , where  $t$  is the number of selected  $r_j$  in the list.

Furthermore, it should also be noted that in the computation of QHT1 and QHT2, each basis state index can be represented in the binary form (here,  $N$  is assumed to be a power of 2 -  $n$  for the sake of simplicity, though an approximation can be used for any value of  $N$ )

$$\begin{aligned} |i\rangle &= |i_1, i_2, \dots, i_n\rangle \\ &= |i_1\rangle \otimes |i_2\rangle \otimes \dots \otimes |i_n\rangle \end{aligned}$$

where,

$$i = i_1 2^{n-1} + i_2 2^{n-2} + \dots + i_n 2^0$$

Since the product terms in QHT1 are powers of  $\omega$ , the computation of intermediate terms can be chained similar to that with QFT. Operations on  $n$ -qubits can be factored into tensor products of  $n$  single qubit operations, which can be implemented using Hadamard gates ( $H$ ) and Controlled Phase Gates ( $R_\theta$ ). Figure 1 demonstrates the QHT circuit, along with the QHT1 and

QHT2 sub-matrix components and their tap points in blue and red respectively. Similar to QFT, it can be seen that the number of gates required for the implementation is given by

$$\frac{n(n+1)}{2} = O(n^2)$$

(since each factor requires one  $H$  gate and one  $R_\theta$  gate more than the previous factor), which is polynomial in the number of qubits. On the other hand, HT typically requires at least  $O(N^2)$  computations, which is an exponential order more than that required for QHT. Furthermore, as discussed earlier, QHT performs accumulation implicitly through Quantum Superposition as opposed to HT that requires a space complexity of  $O(n_r \cdot n_\theta)$ , depending on the granularity of the parametrization. The only space requirement in the Quantum Hough domain computation comes from the space used by the Quantum Counting method to estimate most probable values of  $r$ . However, since  $\theta$  has already been chosen at this stage of computation, the space requirements for the Quantum Counting algorithm is linear in  $k_{max}$  ( $O(k_{max})$ ).

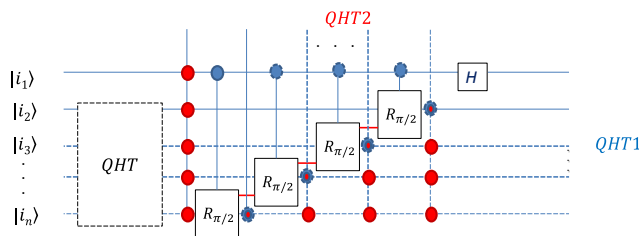


Figure 1. QHT Circuit Implementation

### 3. CONCLUSION

In this paper, we have presented a novel quantum version of the popular Hough Transform and shown the advantages of the Quantum Hough Transform in terms of computational and space requirements over the traditional formulation. A circuit has also been designed for the implementation of the algorithm. The transformation of the accumulation in Hough space into an implicit quantum superposition results in gains in space and speed requirements. It should however be noted that the determination of Hough peaks using QHT has been done through a two-step process that determines peaks in the dependent  $\theta$  space, before re-estimating peaks in the  $r$  space. Future work will involve reformulation of the problem to estimate peaks in the joint  $r, \theta$  space and comparison of performance and trade-offs between the two approaches. In addition, further work is planned towards the goal of implementing and evaluating the algorithm on a quantum computer.

### 4. REFERENCES

[1] Shor, Peter W. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer." SIAM journal on computing 26.5 (1997): 1484-1509.

[2] Grover L.K.: A fast quantum mechanical algorithm for database search, Proceedings, 28th Annual ACM Symposium on the Theory of Computing, (May 1996) p. 212.

[3] Simon, D.R. (1994), "On the power of quantum computation", Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on: 116-123

[4] L. Hales , S. Hallgren, An improved quantum Fourier transform algorithm and applications, Proceedings of the 41st Annual Symposium on Foundations of Computer Science, p.515, November 12-14, 2000.

[5] Talbi, H., Draa, A., & Batouche, M. C. (2004, April). A genetic quantum algorithm for image registration. In Information and Communication Technologies: From Theory to Applications, 2004. Proceedings. 2004 International Conference on (pp. 395-396). IEEE.

[6] Nölle, M., & Suda, M. (2011). Conjugate Variables as a Resource in Signal and Image Processing. arXiv preprint arXiv:1108.5720.

[7] Sun, B., Le, P. Q., Ilyasu, A. M., Yan, F., Garcia, J. A., Dong, F., & Hirota, K. (2011, September). A multi-channel representation for images on quantum computers using the RGBa color space. In Intelligent Signal Processing (WISP), 2011 IEEE 7th International Symposium on (pp. 1-6). IEEE.

[8] Venegas-Andraca, S. E. (2005). Discrete quantum walks and quantum image processing (Doctoral dissertation, Univ Oxford).

[9] Shen, J., & Kang, S. H. (2007). Quantum TV and applications in image processing. Inverse Problems and Imaging, 1(3), 557.

[10] Castagnoli, G. (2009). Discussing the explanation of the quantum speed up. arXiv preprint arXiv:0910.2313.

[11] Eldar, Y. C., & Oppenheim, A. V. (2002). Quantum signal processing. Signal Processing Magazine, IEEE, 19(6), 12-32.

[12] Tseng, C. C., & Hwang, T. M. (2003). Quantum digital image processing algorithms. In 16th IPPR Conference on Computer Vision, Graphics and Image Processing (CVGIP 2003).

[13] Zhou, C., Hu, Z., Wang, F., Fan, H., & Shang, L. (2010). Quantum Collapsing Median Filter. In Advanced Intelligent Computing Theories and Applications (pp. 454-461). Springer Berlin Heidelberg.

[14] Liang Chen, Xu-ming Ye, Li Yu, Bo-xia Xu, Xin-min Shi (2010). A novel feature detector based on quantum-inspired method. In International Conference on Machine Vision (ICMV).

[15] Neven, H., Rose, G., & Macready, W. G. (2008). Image recognition with an adiabatic quantum computer I. Mapping to quadratic unconstrained binary optimization. arXiv preprint arXiv:0804.4457.

[16] Neven, H., Denchev, V. S., Rose, G., & Macready, W. G. (2008). Training a binary classifier with the quantum adiabatic algorithm. arXiv preprint arXiv:0811.0416.

[17] Neven, H., Denchev, V. S., Rose, G., & Macready, W. G. (2009). Training a large scale classifier with the quantum adiabatic algorithm. arXiv preprint arXiv:0912.0779.

[18] Neven, H., Denchev, V. S., Drew-Brook, M., Zhang, J., Macready, W. G., & Rose, G. (2009). NIPS 2009 demonstration: Binary classification using hardware implementation of quantum annealing. Quantum, 1-17.

[19] Denchev, V. S., Ding, N., Vishwanathan, S. V. N., & Neven, H. (2012). Robust Classification with Adiabatic Quantum Optimization. arXiv preprint arXiv:1205.1148.

[20] Schaller, G., & Schützhold, R. (2006). Quantum algorithm for optical-template recognition with noise filtering. Physical Review A, 74(1), 012303.

[21] Brassard, Gilles, Peter Høyer, and Alain Tapp. "Quantum counting." Automata, Languages and Programming. Springer Berlin Heidelberg, 1998. 820-831.

[22] Duda, R. O. and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," Comm. ACM, Vol. 15, pp. 11-15 (January, 1972)

[23] D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes", Pattern Recognition, Vol.13, No.2, p.111-122, 1981

# Temporal multiple instance clustering for dynamic region selection in video

Yongqiang Zhang, Xudong Zhao, Daming Shi, Xianglong Tang  
 School of Computer Science and Technology  
 Harbin Institute of Technology, Harbin 150001 China  
 qt ds0@163.com, {zhaoxudong, dshi, tangxl}@hit.edu.cn

## Abstract

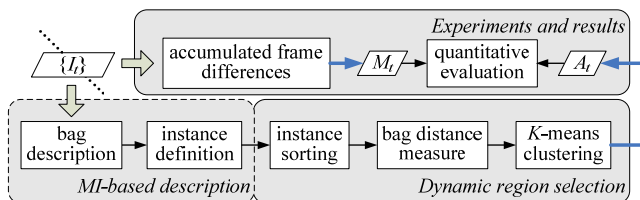
Video dynamic region corresponds to the selection of pixels according to their temporal value changes. Based on temporal multiple instance learning, we propose a dynamic region selection approach with three major contributions. First, a temporal bag and instance description differing from conventional multiple instance definition is made. Second, a bag distance measure is presented as an improvement for multi-instance clustering. Third, learning on clustering centers of bags is modified for rapid convergence. The effectiveness of our method is demonstrated using experiments on videos under different weather conditions.

**Keywords:** Multi-instance, dynamic region selection, Hausdorff distance, K-means.

## 1. INTRODUCTION

Dynamic region selection in video is utilized as a critical task for the subsequent video analysis (e.g. object tracking[1], motion coding[4], scene modeling[3] and etc.). A most common method derives from the accumulate frame differences[6], in which a manual threshold has to be appointed in advance. In fact, video region selection corresponds to the problem of Multiple Instance Learning (MIL, for short)[2]. MIL concerns the labels of the instances included in each bag to classify bags. As to video processing, regions and included image patches in a single frame are considered as bags and instances, respectively[1]. Ordinarily, the temporal information of a video is neglected.

In this paper, we propose a temporal description of bags and instances at pixel level based on MIL, and present a related approach without any parameters for dynamic region selection in video under different circumstances. First of all, bags and the corresponding instances are described. Then, the instances of each bag are sorted. An improved representation of the distance metric between bags is explored. A following algorithm based on K-means clustering is modified to accomplish dynamic region selection. Finally, experiments between the accumulated frame differences and the proposed approach are made. The organization structure is illustrated in Fig. 1.



**Figure 1:** Framework of MIL-based dynamic region selection in video.

## 2. MI-BASED DESCRIPTION

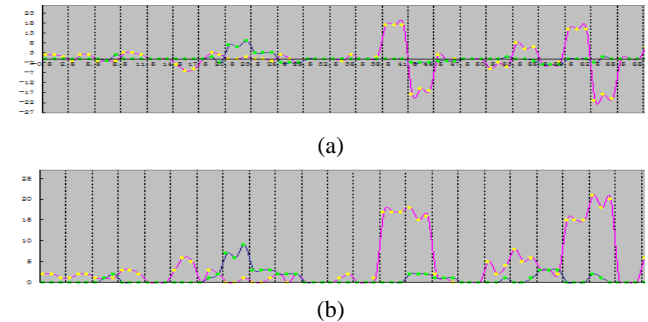
Actually, dynamic region selection consists with a generalized multi-instance problem[5], i.e., it corresponds to the classification of bags. In MIL, a positive bag possibly contains instances labeled negatively and vice versa. In outdoor video that contains different weather conditions, a location of a dynamic region might have very limited pixel changes, due to a fast lighting change existing only in a short time slot over the observation period or few snow appearances. Considering the temporal correlation of pixel values in video, we view each pixel as a bag. Therefore, dynamic region selection is equivalent to the searching of positive bags in scene. Then, a description of instance is in demand for the classification of bags. We refer to  $AD_t^c(y)$  as the absolute difference of a pixel value in RGB color space ( $c \in \{R, B, G\}$ ). That is

$$AD_t^c(y) = |I_t^c(y) - I_{t-1}^c(y)|, \quad (1)$$

where  $y$  and  $t$  represent the location and the current frame, respectively. Correspondingly, a definition of instance is expressed as follows,

$$Instance_t(y) = [AD_t^r(y), AD_t^g(y), AD_t^b(y)]. \quad (2)$$

An example of MI-based description according to two pixels (e.g., a positive pixel and a negative pixel) is shown in Fig. 2.



**Figure 2:** MI-based description of two pixels. (a)The original frame difference; (b)The absolute frame difference.

The three yellow points between two adjacent dotted lines represent an instance of the dynamic pixel corresponding to the difference in R, G and B channel, respectively. Similarly, the three green ones represent an instance of the static pixel.

## 3. DISTANCE MEASURE IN MIL

We aim at differentiating between static and dynamic bags, and meanwhile clustering bags with the same label. The fact is that each positive bag in a dynamic region probably contains static instances labeled negatively. Commonly, Hausdorff distance (HD) is selected as a standard measure between bags in MIL. Thus, the HD for classification of bags in video is expressed as follows,

$$\begin{aligned} \max h(A, B) &= \max_{a \in A} \min_{b \in B} \|a - b\| \\ \max h(B, A) &= \max_{b \in B} \min_{a \in A} \|a - b\| \end{aligned} \quad (3)$$

where  $A$  and  $B$  represent bags.  $\|\cdot\|$  denotes a norm. The HD expressed in Equation (3) is a directed distance. Therefore, we select the max HD as a distance measure. That is

$$\max H(A, B) = \max\{\max h(A, B), \max h(B, A)\}. \quad (4)$$

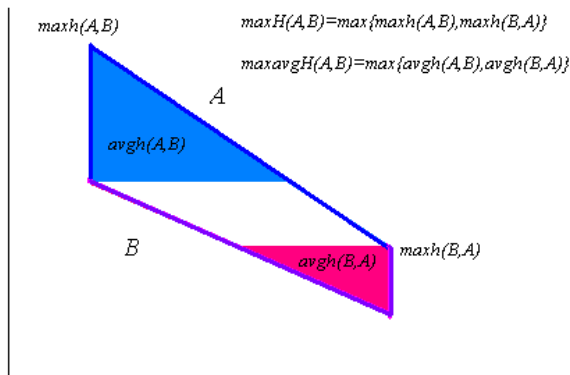
However, the selection of a dynamic region is conditioned by different circumstances in video. We follow our previous research [7] that the absolute difference of a pixel value in video under fast lighting change keeps a continuous intensity change. That is to say,  $AD_i^{\max}(y)$  remains almost the same over the right period that corresponds to fast illumination variations. Thus, it is the  $\max H$  expressed in Equation (4) that can be better used for real-time dynamic region selection under fast lighting change. On the contrary,  $\max H$  is sensitive to noise (e.g., randomly distributed snow over time) because of neglecting the contribution from other instances except the farthest couple of instances from two different bags. Therefore, we additionally define an average HD to classify different bags in video. That is

$$\begin{aligned} \text{avgh}(A, B) &= \frac{1}{|A|} \sum_{a \in A} \min_{b \in B} \|a - b\| \\ \text{avgh}(B, A) &= \frac{1}{|B|} \sum_{b \in B} \min_{a \in A} \|a - b\| \end{aligned} \quad (5)$$

where  $|\cdot|$  denotes the instance number of a bag.  $\text{avgh}$  is also a directed distance. Thus, we define a maximal average of HD as follows,

$$\max \text{avgh}(A, B) = \max\{\text{avgh}(A, B), \text{avgh}(B, A)\}. \quad (6)$$

This distance measure takes into account all the instances, so it's robust to the noise. That is, the weight of very dynamic instances is reduced. Besides, their dynamic characteristic remains still. The difference between the maximum average HD and the maximum HD is shown in Fig. 3.



**Figure 3:** The difference between the maximum average HD and the maximum HD

In this figure, we select two special bags, the instances of which have been sorted correspond to the upper line  $A$  and lower line  $B$ . The maximum HD is equal to the longer edge of the two vertical lines, and the maximum average HD is equal to the larger of the two areas.

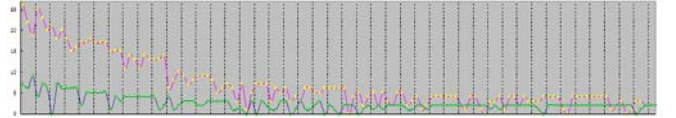
## 4. SORTING AND SELECTION OF INSTANCES

The distance of each instance in bag  $A$  to each instance in bag  $B$ , and vice versa. As the number of instances increases, the time cost for calculating the distance grows exponentially. In order to reduce the time cost, we sort instances of a bag in a descending order, and take the  $M$  first instances as a representation of the bag. Of course, we should ensure that the stability of distance measure remains after sorting. And there is another motive for sorting, i.e., we can update each center more quickly when clustering bags in the following step.

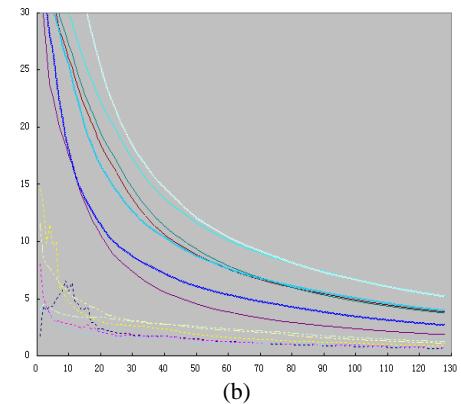
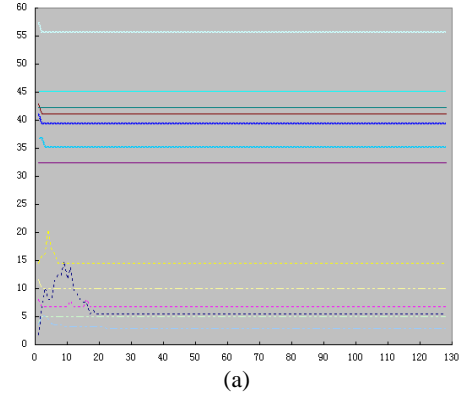
As to location  $y$  in each frame, we select its max component  $AD_i^{\max}(y)$  in R, G and B channel for instance sorting. That is,

$$\begin{aligned} AD_a^{\max}(y) &= \max\{AD_a^r(y), AD_a^g(y), AD_a^b(y)\} \\ AD_b^{\max}(y) &= \max\{AD_b^r(y), AD_b^g(y), AD_b^b(y)\} \end{aligned} \quad (7)$$

An instance  $a$  should be sorted before an instance  $b$ , if  $AD_a^{\max}(y) > AD_b^{\max}(y)$ . The sorting result is shown in Fig. 4.



**Figure 4:** Sorting of instances of two bags



**Figure 5:** The variation of HD with the growth of the number of instances. (a) The maximum HD (b)The maximum average HD

After sorting, the maximum HD is equal to the distance from the first instance of a bag to the first instance of another bag. When the number of selected instances increases, the maximum HD keeps mostly constant.

As described in Section 3, the maximum average HD is calculated according to a directed distance  $avgh$ , which is the average distance of all the instances of a bag to another. Since the dynamic appearances of the low-ranking instances is less than that of the ones ranking high, the maximum average HD decreases with the growth of the number of selected instances. Yet, the relative distance between the static and dynamic bags keeps stable. Meanwhile, the influence of isolated instances is reduced.

As shown in Fig. 5, we select two groups of typical bags (a static group and a dynamic group) and calculated the maximum HD and the maximum average HD with a growing number of selected instances. In Fig. 5, the solid lines point to outer-class distances from dynamic bags to static bags. The dashed lines point to inner-class distances in static bags or in dynamic bags. Experimental result is consistent with the analysis above. With the growth of the number of selected instances, the maximum HD remains as a straight line. The maximum average HD reduces gradually as the curve  $y=1/x$ , but the relative position remains stable. Moreover, the outer-class distance is much larger than the inner-class distance with both of the distance measures mentioned above. It means that the maximum HD or the maximum average HD is feasible to distinguish the static bags and the dynamic ones.

## 5. DYNAMIC REGION SELECTION BY K-MEANS

We follow the prevailing  $K$ -means clustering step to accomplish a classification of bags representing the static and dynamic region in video. Using HD expressed in Equation (4) and Equation (6), we improve the calculation of clustering center. That is

$$\bar{B}_j = \frac{1}{|G_j|} \sum_{B_q \in G_j} B_q \quad (j=1,2), \quad (8)$$

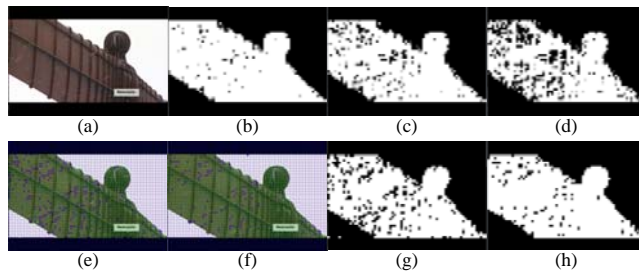
where  $B_q$  represents a bag in cluster  $G_j$ .  $|G_j|$  denotes the bag number of  $G_j$ . That is, we consider  $\bar{B}_j$  as a new clustering center.

After clustering, the pixels are divided into two groups: One group represents the dynamic pixels, the illumination variation of which is stronger. And another group corresponds to the static pixels.

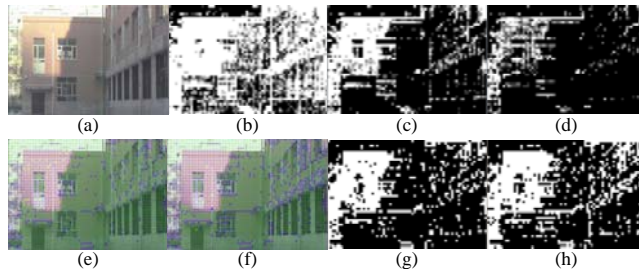
## 6. EXPERIMENTS AND RESULTS

We have tested our method on a database named as DRS<sup>1</sup> containing two video clips with obvious dynamic regions. One is a privately shot video clip with fast lighting change (namely DRS\_FLC). The other is a public movie clip with snow (namely DRS\_S). Moreover, accumulated frame differences are utilized for qualitative comparison. It is difficult to label the ground truths manually for quantitative analysis. Therefore, we set a large number of manual thresholds of accumulated frame differences, and artificially select a best dynamic region selection result as a ground truth. TP and FP represent the truly and falsely selected dynamic region. Meanwhile, TN and FN denote the true and false static region. Furthermore, we make Precision=TP/(TP+FP) and Recall=TP/(TP+FN). The experimental results are shown in Fig. 6, Fig. 7 and Table 1. A demo of the experimental results is also shown in DRS. It can be observed that  $max\ avgH$  is more competent for dynamic region selection in video with snow than  $max\ H$ . On the contrary,  $max\ H$  works better on classification of

dynamic and static bags in video with fast lighting change than  $max\ avgH$ .



**Figure 6:** Dynamic region selection on DRS\_S. (a) Video clip with snow; (b) Accumulated frame differences with Th=128; (c) Accumulated frame differences with Th=160 (selected); (d) Accumulated frame differences with Th=196; (e) Selection result using  $maxH$ ; (f) Selection result using  $maxavgH$ ; (g) Binary result using  $maxH$ ; (h) Binary result using  $maxavgH$ .



**Figure 7:** Dynamic region selection on DRS\_FLC. (a) Video clip with fast light change; (b) Accumulated frame differences with Th=128; (c) Accumulated frame differences with Th=196 (selected); (d) Accumulated frame differences with Th=256; (e) Selection result using  $maxH$ ; (f) Selection result using  $maxavgH$ ; (g) Binary result using  $maxH$ ; (h) Binary result using  $maxavgH$ .

**Table 1:** Comparison on a quantitative analysis

		TP	FP	TN	FN	Precisi on	Recall
DRS_ S	$maxH$	1519	123	1762	52	0.925	0.967
	$maxavgH$	1624	18	1729	85	0.989	0.950
DRS_ FLC	$maxH$	2280	173	881	122	0.929	0.949
	$maxavgH$	2182	271	997	26	0.890	0.988

## 7. CONCLUSION

In this paper, we propose a MIL-based dynamic region selection approach in video. According to pixel-wise time correlation and color information, we firstly describe bags and instances in video. After the sorting of instances in each bag, we improve a  $max$  Hausdorff distance measure and present a maximal average one adaptive to separate videos under different weather conditions. The clustering center is modified for the rapid convergence of MIL-based  $K$ -means clustering. Experimental results indicate the effectiveness of our method, which provides a first step support for subsequent video analysis.

<sup>1</sup> <http://pr-ai.hit.edu.cn/percy/DRS>

## 8. REFERENCES

- [1] Babenko, B., Yang, M. H., & Belongie, S. (2011). Robust object tracking with online multiple instance learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8), 1619-1632.
- [2] Dietterich, T. G., Lathrop, R. H., & Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1), 31-71.
- [3] Grech, R., Monekosso, D. N., & Remagnino, P. (2012). Building visual memories of video streams. *Electronics letters*, 48(9), 487-488.
- [4] Secker, A., & Taubman, D. (2004). Highly scalable video compression with scalable motion coding. *Image Processing, IEEE Transactions on*, 13(8), 1029-1041.
- [5] Weidmann, N., Frank, E., & Pfahringer, B. (2003). A two-level learning method for generalized multi-instance problems. In *Machine Learning: ECML 2003* (pp. 468-479). Springer Berlin Heidelberg.
- [6] Yin, H., Chai, Y., Yang, S. X., & Yang, X. (2011). Fast-moving target tracking based on mean shift and frame-difference methods. *Systems Engineering and Electronics, Journal of*, 22(4), 587-592.
- [7] Zhao, X., Liu, P., Liu, J., & Tang, X. (2011, November). A time, space and color-based classification of different weather conditions. In *Visual Communications and Image Processing (VCIP), 2011 IEEE* (pp. 1-4). IEEE.

## About the author

Yongqiang Zhang is pursuing M.Sc. degree in Computer Science from Harbin Institute of Technology. His current research interest lies in motion tracking.

Xudong Zhao received his Ph.D. degree in Artificial Intelligence and Information Processing from Harbin Institute of Technology. His current research His research interests include statistical machine learning, pattern recognition, time series analysis and image processing.

Daming Shi (M'02-SM'04) received his Ph.D. degree in mechanical control from Harbin Institute of Technology, China, and the Ph.D. degree in computer science from University of Southampton, United Kingdom. He is currently a Professor at Harbin Institute of Technology, China, and served as an Assistant Professor at Nanyang Technological University, Singapore, from 2002 to 2009. His current research interests include machine learning, medical image processing, pattern recognition and neural networks.

Xianglong Tang received his PhD degree from Harbin Institute of Technology, China in 1995. He is currently a Professor at school of computer science and technology in Harbin Institute of Technology. His research interests include OCR, biometrics, image processing and pattern recognition.

# Dynamic region selection in video based on spatio-temporal multiple instance learning

XiaoZheng Wang, XuDong Zhao, Peng Liu and XiangLong Tang  
 School of Computer Science and Technology  
 Harbin Institute of Technology, Harbin 150001 China  
 wangxiaozheng@foxmail.com, {zhaoxdong, pengliu, tangxl}@hit.edu.cn

## Abstract

The selection of dynamic region in video plays an important role in many subsequent vision-based applications, especially in scene classification with different weather conditions. In this paper, we extract five local features from pixel blocks of each frame in a video, and propose an approach to dynamic region selection based on a presented description of spatio-temporal multiple instances. The effectiveness of our method is shown using experiments on videos under different weather environments.

**Keywords:** Multi-instance, spatio-temporal feature, dynamic region, K-means.

## 1. INTRODUCTION

Dynamic region selection aims at finding the significant regions that are composed of the locations containing obvious changes in video. It plays an important role in many applications (e.g., motion coding [2], motion detection [5], scene modeling [4], scene classification [3], weather classification [6] and etc.). In the same way, Multiple Instance Learning (MIL) [1] concerns labels of instances included in each bag to classify bags. In video, bags correspond to image patches over a time slot; while, instances refer to spatial or temporal features.

In this paper, we propose a dynamic region selection approach based on spatio-temporal multiple instance learning using five local features from a pixel block. First of all, we subdivide a frame image into  $10 \times 10$  pixel blocks, and extract features to form spatial multiple instances. Then, we aggregate features frame by frame in each pixel block to produce temporal multiple instances. Together, the spatio-temporal multiple instances consist of a bag that corresponds to a pixel block. Finally, K-means clustering of bags is used to select dynamic region in video. The organization structure is illustrated in Fig. 1.

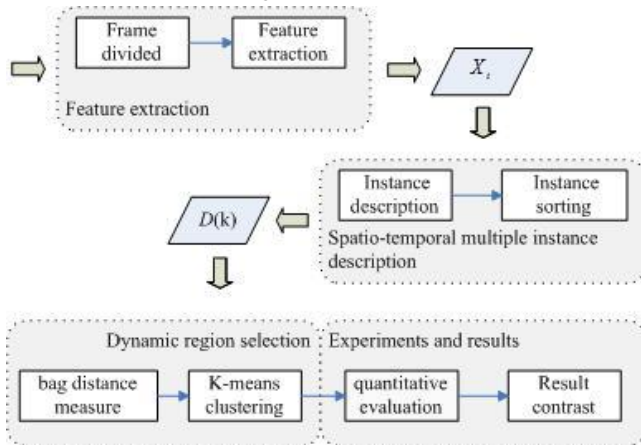


Fig.1. Framework of spatio-temporal multiple instance based dynamic region selection in video

## 2. FEATURE EXTRACTION

Features are derived from a pixel block. First, we divide an image into  $10 \times 10$  pixel blocks. Five features are extracted in each block, i.e., hue, saturation, minimum brightness, local contrast and sharpness. Gray-scale features are commonly used for image processing tasks that range from low level algorithms to highly sophisticated modules. However, we pay more attention to color information according to the low visibility deriving from bad weather. Compared to RGB color space, HSV space keeps the same way on perception of color information that human eye does. So we extract hue, saturation and brightness at pixels. The minimum value of brightness ( $V_{\min}$ ) and the local mean values of hue ( $H$ ) and saturation ( $S$ ) are taken in each pixel block. To increase the robustness of contrast estimation, we define the local contrast as follows,

$$C = \frac{V_{\max} - V_{\min}}{V_{\max} + V_{\min}}, \quad (1)$$

where  $C$ ,  $V_{\min}$  and  $V_{\max}$  represent the local contrast, the minimum and the maximum value of brightness, respectively. Besides, clearly distinguishable objects under fine weather conditions are expected to have sharp edges with large contrast differences. In addition to the contrast feature discussed above, a gradient-based method is used to determine the sharpness of the test images. It is based on an average determination of the sobel gradient magnitude, which is defined as follows,

$$T = \frac{\sum_i \sqrt{S_x^2(i) + S_y^2(i)}}{\sum_i 1}. \quad (2)$$

The sharpness  $T$  is derived from an average determination of the sobel gradient magnitude  $S_x$  and  $S_y$  with  $i$  belonging to a pixel block.

## 3. MULTIPLE INSTANCE DESCRIPTION

Multiple-instance learning (MIL) is a variation on supervised learning. Instead of receiving a set of instances which are labeled positive or negative, the learner receives a set of bags that are labeled positive or negative. Each bag contains many instances. The most common assumption is that a bag is labeled negative if all the instances in it are negative. On the other hand, a bag is labeled positive if there is at least one instance in it which is positive. From a collection of labeled bags, the learner tries to either induce a concept that will label individual instances correctly or learn how to label bags without inducing the concept. For a more accurate and detailed expression of spatial features in each block, we subdivide each  $10 \times 10$  pixel block into four  $5 \times 5$  pixel blocks in space, and extract features from every  $5 \times 5$  block. In other words, we get spatial feature vectors  $X_t$  from a  $10 \times 10$  block. Let  $X_t$  be  $X_t = \{x_t^1, x_t^2, x_t^3, x_t^4\}$ .  $t$  represents the current frame.

The feature vector from a  $5 \times 5$  pixel sub-block  $x_i^m$  ( $m=1,2,3,4$ ) is expressed as  $x_i^m = (H, S, V_{\min}, C, T)$ . Furthermore, we aggregate feature vectors frame by frame. Assuming the frame number to be  $M$ , each  $10 \times 10$  block contains  $M$  spatial multiple instances, which is recorded as  $X_1, X_2, \dots, X_i, \dots, X_M$ . In order to observe the changes in each block more clearly, we sort these  $M$  temporal multiple instances from largest to smallest by calculating the modulus of the obtained spatial vectors, which is expressed as  $|X_i| = \frac{1}{4}(|x_i^1| + |x_i^2| + |x_i^3| + |x_i^4|)$ . Thus, a new sequence of temporal instances in a bag is expressed as  $Y_1, Y_2, \dots, Y_M$ , where  $Y_i$  is  $Y_i = \{y_i^1, y_i^2, y_i^3, y_i^4\}$ . In addition, a corresponding subtraction operation to different temporal instances in each bag is made, i.e.,  $d_i^m = y_i^m - y_j^m$  ( $i \in [1, M], j \in [i+1, M]$ ). Let  $y_i^m$  be  $y_i^m \in Y_i$  and  $y_j^m$  be  $y_j^m \in Y_j$ . Besides, we sort these spatio-temporal vectors from largest to smallest by calculating the modulus  $|d_i^m|$  to get a sequence  $b_1, b_2, \dots, b_N, \dots, b_{2M(M-1)}$ . In order to reduce the complexity of the algorithm, we select the first  $N$  ( $N=M/4$ ) vectors as the spatio-temporal multiple instances. The bag of a pixel block is expressed as  $D(k) = \{b_1(k), b_2(k), \dots, b_N(k)\}$ , where  $k \in [1, n]$ .  $n$  denotes the number of  $10 \times 10$  pixel blocks in an image.

#### 4. UNSUPERVISED CLASSIFICATION

$K$ -means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume  $k$  clusters) fixed a priori. The main idea is to define  $k$  centroids, one for each cluster. In this paper, we define  $k$  as 2. These two centroids should be placed in a cunning way because different location causes different result. So, the better choice is to place them as much as possible far away from each other.  $K$ -means clustering is utilized to accomplish spatio-temporal multiple instance learning. Bags representing locations containing obvious changes in video are selected as dynamic region. Two different distance metrics between bags based on Hausdorff distance (HD) are proposed. The max HD for classification of bags in video is expressed as follows,

$$\begin{aligned} \max_h(D(k), D(r)) &= \max_{b_p(k) \in D(k)} \min_{b_q(r) \in D(r)} \|b_p(k) - b_q(r)\| \\ \max_h(D(r), D(k)) &= \max_{b_q(r) \in D(r)} \min_{b_p(k) \in D(k)} \|b_p(k) - b_q(r)\|, \end{aligned} \quad (3)$$

where  $k \in [1, n]$  and  $p, q \in [1, N]$ .  $\|\cdot\|$  denotes a norm. The HD expressed in Equation (3) is a directed distance. Therefore, the max HD distance between different bags is

$$\max H(D(k), D(r)) = \max\{\max_h(D(k), D(r)), \max_h(D(r), D(k))\}. \quad (4)$$

The average HD distance to classify different bags is

$$\begin{aligned} \text{avgh}(D(k), D(r)) &= \frac{1}{|D(k)|} \sum_{b_p(k) \in D(k)} \min_{b_q(r) \in D(r)} \|b_p(k) - b_q(r)\| \\ \text{avgh}(D(r), D(k)) &= \frac{1}{|D(r)|} \sum_{b_q(r) \in D(r)} \min_{b_p(k) \in D(k)} \|b_p(k) - b_q(r)\| \end{aligned} \quad (5)$$

$\text{avgh}$  is also a directed distance. Thus, we define a maximal average of HD as follows,

$$\max \text{avgh}(D(k), D(r)) = \max\{\text{avgh}(D(k), D(r)), \text{avgh}(D(r), D(k))\}. \quad (6)$$

Then, we follow the follow  $K$ -means clustering step as shown in Algo.1 to accomplish a classification of bags representing the static and dynamic region in video.

**Input:** We define the sorted collections of bags as  $U$ , and the number of the clusters  $K$  as 2.

**Output:** The label of each bag and the centroid of every cluster.

1. Select  $K$  bags from collections  $U$  randomly as the initial centroids  $C_1, C_2, \dots, C_K$ . A cluster collection  $G_j$  is formed using initial centroids, where  $G_j = \{C_j\}$ ,  $j = 1, 2, \dots, K$ .
2. while  $\forall j = 1, 2, \dots, K$ , we have  $C_j^* \neq C_j$ , do
3.  $C_j = C_j^*$ ;
4. For  $i \leftarrow 1$  to  $n$  do
5. If  $H(B_i, C_j) \leq H(B_i, C_k)$  ( $\exists j = 1, 2, \dots, K, \forall k = 1, 2, \dots, K$ , and  $j \neq k$ ) then
6.  $G_j \leftarrow B_i$ ; //  $B_i$  ( $i = 1, 2, \dots, n$ ) is the data bag,  $G_j$  is the cluster collection.
7.  $G_j^* = G_j \cup \{B_i\}$ ;
8.  $G_j = G_j^*$ ;
9. //Updating the centroid
10.  $\bar{B}_j = \frac{1}{|G_j|} \sum_{B_i \in G_j} B_i$ ;
11. End if
12. End for
13. End while

#### 5. EXPERIMENTS AND RESULTS

We have tested our method on two video clips with obvious dynamic regions. One is a privately shot video clip with fast lighting change (namely, DRS\_FLC). The other is a public movie clip with snow (namely, DRS\_S). Temporal multiple instance learning method based on intensity differences is utilized for qualitative comparison. Moreover, the ground truth of dynamic regions in each video is manually labeled for quantitative analysis.  $TP$  and  $FP$  represent the truly and falsely segmented dynamic region. Meanwhile,  $TN$  and  $FN$  denote the true and false static region. Furthermore, *Precision* and *Recall* are respectively defined as  $\text{Precision} = TP / (TP + FP)$  and  $\text{Recall} = TP / (TP + FN)$ . The experimental results are shown in Fig. 2, Fig. 3 and Table 1. It can be observed that max avgH is more competent for dynamic region selection in video with snow than maxH. On the contrary, maxH works better on classification of dynamic and static bags in video with fast lighting change than max avgH.

#### 6. CONCLUSION

In this paper, a dynamic region selection approach in video is proposed based on spatio-temporal multiple instance learning. Considering the influence of different weather environments, we firstly extract spatial feature vectors from sub-blocks in a pixel block. After that, temporal aggregation of spatial feature vectors is performed for the constitution of temporal multiple instances. By means of the subtraction operation, we obtain an instance bag which could present the pixel change in each block. Finally,  $K$ -means clustering of these bags is used to select dynamic areas



in video based on two different Hausdorff distance measures. Experiments indicate the effectiveness of our method.

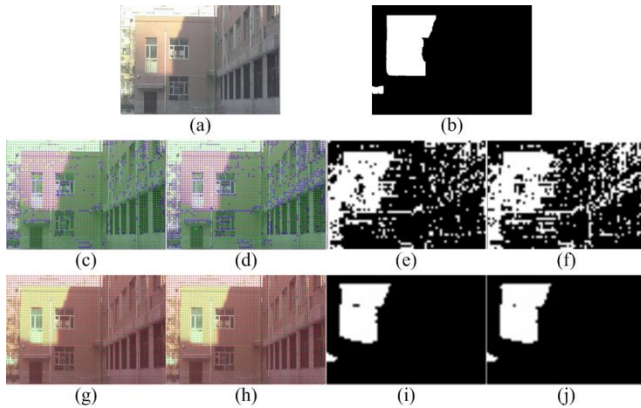


Fig. 2 Dynamic region selection on DRS\_FLC

- (a) Video clip with fast light change
- (b) Labeled ground truth
- (c) Result of MIL based on temporal intensity differences using maxH
- (d) Result of MIL based on temporal intensity differences using max avgH
- (e) Corresponding binary result using maxH
- (f) Corresponding binary result using max avgH
- (g) Result of spatio-temporal MIL using maxH
- (h) Result of spatio-temporal MIL using max avgH
- (i) Corresponding binary result using maxH
- (j) Corresponding binary result using max avgH

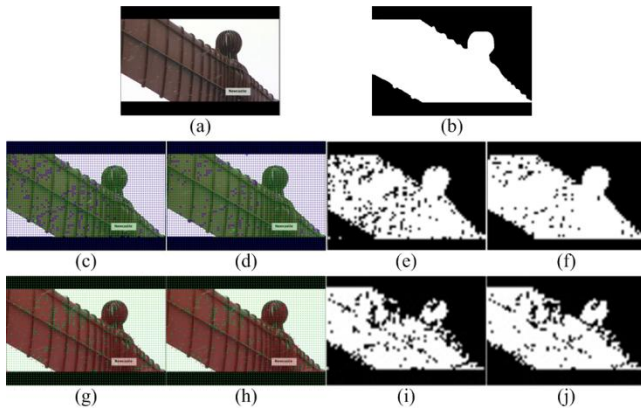


Fig. 3 Dynamic region selection on DRS\_S

- (a) Video clip with snow
- (b) Labeled ground truth
- (c) Result of MIL based on temporal intensity differences using maxH
- (d) Result of MIL based on temporal intensity differences using max avgH
- (e) Corresponding binary result using maxH
- (f) Corresponding binary result using max avgH
- (g) Result of spatio-temporal MIL using maxH
- (h) Result of spatio-temporal MIL using max avgH
- (i) Corresponding binary result using maxH
- (j) Corresponding binary result using max avgH

## 7. REFERENCES

[1] Ali, S., Shah, M.: ‘Human action recognition in videos using kinematic features and multiple instance learning’, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2010, 32, (2), pp. 288-303

[2] Ascenso, J., Brites, C., Pereira, F.: ‘Content adaptive Wyner-Ziv video coding driven by motion activity’, *IEEE International Conference on Image Processing (ICIP)*,

October 2006, Atlanta, GA, pp. 605-608 AuthorC. Reference3.

[3] Bosch, A., Zisserman, A., Muoz, X.: ‘Scene classification using a hybrid generative/discriminative approach’, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2008, 30, (4), pp. 712-727

[4] Grech, R., Monekosso, D., Remagnino, P.: ‘Building visual memories of video streams’, *Electronics Letters*, 2012, 48, (9), pp. 487-488

[5] Kim, T., Im, J., Paik, J.: ‘Video object segmentation and its salient motion detection using adaptive background generation’, *Electronics Letters*, 2009, 45, (11), pp. 542-543

[6] Roser, M., Moosmann, F.: ‘Classification of weather situations on single color images’, *IEEE Intelligent Vehicles Symposium*, June 2008, Eindhoven, Netherlands, pp. 480-485

		TP	FP	TN	FN	Precision	Recall
DRS _S	maxH	873	28	1678	877	0.9689	0.4989
	maxavgH	935	26	1680	815	0.9729	0.5343
DRS _FLC	maxH	326	6	2938	186	0.9819	0.6367
	maxavgH	335	3	2941	177	0.9911	0.6543

Table 1: Comparison on a quantitative analysis

## 8. ABOUT THE AUTHOR

Xiaozheng Wang is pursuing M.Sc. degree in Computer Science from Harbin Institute of Technology. His current research interest lies in motion tracking and machine learning.

Xudong Zhao received his Ph.D. degree in Artificial Intelligence and Information Processing from Harbin Institute of Technology. His current research interests include statistical machine learning, pattern recognition, time series analysis and image processing.

Peng Liu received his Ph.D. degree in 2007. He is currently an Associate Professor at school of computer science and technology in Harbin Institute of Technology. His research interests include digital signal processing, pattern recognition and VLSI design.

Xianglong Tang received his PhD degree from Harbin Institute of Technology, China in 1995. He is currently a Professor at school of computer science and technology in Harbin Institute of Technology. His research interests include OCR, biometrics, image processing and pattern recognition.

# An Integrated Image Clarity Method for Coal Mine Safety Monitoring

Yanqin Zhao

College of Computer and Information Engineering

Heilongjiang Institute of Science and Technology, Harbin, China

zyq\_jean@sina.com.cn

## Abstract

According to the poor image quality of the monitoring system in coal mine monitoring, an integrated image clarity method was presented which using adaptive multi-level noise reduction median filter based on the direction in the spatial domain to denoise the image first, then using multi-wavelet transform to enhance the low-light image in the transform domain. This integrated approach combines the benefits of spatial domain and transform domain noise reduction and enhancement. It can reduce the impulse noise effectively, while retaining good image detail features of the coal mine images. Experiments show that the method can greatly improve the clarity of the image of coal mine so can improve the readability of the image, and this is of great significance for the coal mine safety monitoring.

**Keywords:** Coal mine, Median filter, Multi-wavelet transform, Image denoise, Image enhancement.

## 1. INTRODUCTION

In these years, the coal mine safety monitoring system such as the coal industry television has been used widely to guide the mine safety production and command with intuitive, convenient and reliable means during the development of coal mine science. The video safety monitoring system not only monitors the mine production directly on the ground to detect the risks underground and take preventive measures, but also provides the first-hand information for the ex post analysis of the accidents.

For the harsh environment of the coal mine, such as the poor lighting, low illumination and serious dust, the quality of images captured by the monitoring system has been degraded. The correlation between pixels structure and content has been destroyed and this makes the further image analysis and finding the danger in the mine much more difficult. So it is very necessary to preprocess the images captured by the system for denoising or enhancement and to increase the clarity of the monitoring images. This article designs an integrated algorithm based on directional adaptive median filter and wavelet transform according to the illumination under complex noise environment in the coal mine.

## 2. COAL MINE IMAGE NOISES

Image noise usually means the visible error information produced by CCD/CMOS or digital signal system. Visible noise in digital images is often affected by temperature effects and the ISO sensitivity: the higher the two values are, the worse the effect is. For the mine harsh conditions, the CCD images obtained through D/A converter, transmission lines will produce noise pollution. There are variable types of noise collected in the monitoring images, but the most common and most serious impact on image quality in general are impulse noise and Gaussian noise. So how

to reduce the effects of these two kinds of noise is the key to improve the monitoring image quality.

## 3. THE INTEGRATED IMAGE CLARITY METHOD

### 3.1 Median Filtering Principle

Non-linear filtering has a wide range of applications in image processing. It has good characteristics on the noise rejection and image edge maintaining. Median filter is a nonlinear filtering operation and belongs to the spatial denoising methods. Its output is related to the distribution of input noise density. The squared difference of output noise is inversely proportional to the square of density function of the input noise. For the rejection of random noise, median filtering is not so good as the mean filter. But in terms of pulse interference, in particular of the pulse width less than half the length of filter window and far away from the narrow pulse, the median filter is very effective. Since there is lots of impulse noise in the coal mine environment, median filtering noise can be used in the preliminary process of the images.

Median filter uses a sliding window containing odd points to sort the neighborhood pixels in terms of pixel gray level and the middle value is the output pixel. Its mathematics description is as following.

If  $S$  is the neighborhood set of pixel  $(x_0, y_0)$  (including  $(x_0, y_0)$ ),  $(x, y) \in S$ ,  $f(x, y)$  means the gray value of point  $(x, y)$ ,  $|S|$  means the number of elements in set  $S$ ,  $Sort$  means to sort the elements in order. Then for smoothing can be expressed as:

$$f'(x_0, y_0) = \left[ \underset{(x, y) \in S}{Sort} f(x, y) \right]_{\frac{|S|+1}{2}} \quad (1)$$

### 3.2 Adaptive Multi-level median filtering based on the direction

As median filter is characterized by simple and fast operation, especially for unipolar or bipolar pulse noise, it is very effective. Adaptive multi-level median filter is an improved algorithm based on the traditional median filter. It sets several strip sub-windows in the rectangular window so to protect the edge information in more directions than the traditional median filter while preserving the characteristics of median filter which can suppress impulse noise effectively.

The adaptive multi-level median filter based on direction is as follows (5\*5, for example).

(1) Supposing the current pixel is located in the center of the pixels and its value is  $x(i, j)$ , Define a set  $D = \{D_i, i = 1, 2, \dots, 8\}$ ,  $D_i$  is the absolute difference value of the current pixel and its 8-point adjacent pixels. Sort the elements of  $D$  in ascending order getting the set  $d_1 \leq d_2 \leq \dots \leq d_8$ ,  $d_1 \leq d_2 \leq \dots \leq d_8$ .

(2) Make the two smallest elements in set  $D_{order}$  as the feature direction in the  $3 \times 3$  window, and using this direction as the trend of the  $5 \times 5$  direction.

(3) Supposing the current pixel value is  $x(i, j)$ , the output  $Y(i, j)$  in the  $5 \times 5$  window in which  $(i, j)$  is the center is as follows:

$$Y(i, j) = \begin{cases} MMF[x(i, j)] \\ SCMMF[x(i, j)] \\ M[x(i, j)] \end{cases} \quad (2)$$

$MMF[x(i, j)]$  is the processing result when the current pixel is an isolate pixel or the direction of related line feature is horizontal or vertical.  $SCMMF[x(i, j)]$  is the processing result after the improved multi-level median filter when the related line feature has an angle in  $(0, \pi/2)$ . In other cases that it's difficult to determine the direction of feature line,  $M[x(i, j)]$  is used to express the pixel value filtered.

In the formula,

$$M[x(i, j)] = med[m_1, m_2, m_3, m_4, x(i, j)] \quad (3)$$

This directional adaptive multi-level median filtering denoise method can adapt to the direction of the image edge better comparing with the traditional median filter and it can keep a better compromise between the denoising and edge protection. Therefore, if the method is applied to underground monitoring for denoising complex environment, not only the effect of impulse noise can be reduced more obviously, but also the part of the Gaussian noise can be reduced.

### 3.3 Image enhancement with multi-wavelet transform

Wavelet transform has many features such as low entropy, multi-resolution, decorrelation and flexible radical selection, which can do the local analysis both in the time and frequency domain at the same time, and extract the local singular characteristics of the signal flexible. At present, denoising and enhancement based on wavelet has become an important way in image processing field.

First, the image after median filtering is denoised with multi-wavelet; this can reduce the most Gaussian noise. Then the image will be enhanced with the multi-wavelet to increase the contrast of the image. For enhancement, the image is transformed with multi-wavelet first, and then the image is decomposed into several parts with different size, position and orientation. Then the coefficients of some different position and orientation are changed according to the need, so that some components which we are interested in are enlarged while unnecessary component reduced. Finally, the

enhancement image was reconstructed with multi-level wavelet. The wavelet enhancement process is shown in Figure1.

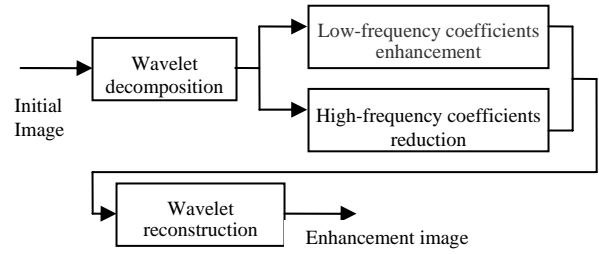


Figure 1: Enhancement process of wavelet

The main information of the decomposed image (i.e. the image contours) is characterized by low frequency part while the details of the image is characterized by high frequency. Therefore, the decomposed low frequency coefficients are weighted while high frequency coefficients are weakened. After such wavelet transform treatment, an enhanced image can be achieved. In the low contrast environment such as coal mine, if the quality of initial denoised image is not so good, the wavelet transform will be used to enhance it, which can improve the image quality greatly.

## 4. EXPERIMENTS ANALYSIS

To verify the integrated algorithm, two images from different coal mine have been used. In order to highlight the noise in the original image, some salt and pepper noise(0.02) and Gaussian noise have been added as shown in Figure 2 (a) and Figure 3 (a). The processing tool is Matlab2007, the processed comparison image effects are shown in Figure 2 and Figure 3 respectively.

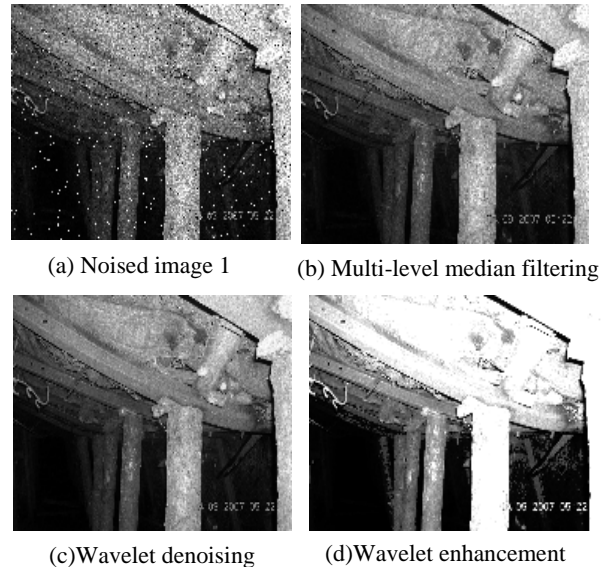
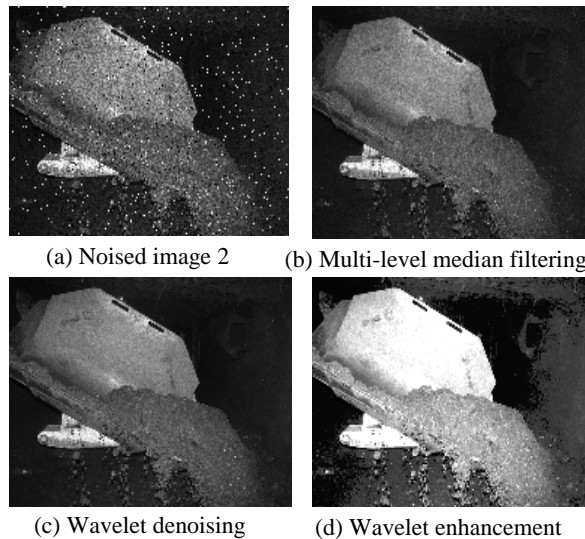


Figure 2 : Image1 process effects

The noise images has been filtered lots of salt and pepper noise by the directional multi-level median filtering, and the image quality has been improved greatly as shown in Figure2 (b) and Figure3 (b). Since the Gaussian noise cannot be reduced effectively by median filter algorithm, the multi-wavelet denoising has been

used to do further denoising, the results are shown in Figure2 (c) and Figure3 (c). From these two parts, we can see the images are clearer than the former. This can also be proved by the increasing of objective parameters Peak Signal to Noise Ratio (PSNR) as shown in Table 1.



**Figure 3:** Image 2 process effects

After processing above, sometimes the background is still dark, so the multi-wavelet transform is used to enhance the images. Figure2 (d) and Figure3 (d) show the images enhanced by multi-wavelet transform. Due to the characteristics of wavelet transform, image quality has improved while the image details are preserved.

**Table 1:** Image quality objective parameters

PSNR (dB)	(a) Noised image	(b) multi-level median filtering	(c) Wavelet denoising image
Image 1	17.9014	26.013	27.6952
Image 2	19.1431	28.8422	29.9378

## 5. CONCLUSIONS

Safety monitoring in the coal mine is an important safeguard to achieve safety production. But the special complex environment of the underground leads to poor image quality thus increasing the difficulty of monitoring for supervisors. This also restricts the application of the intelligent monitoring technology in the coal mine monitoring. An integrated image processing method is presented based on the combination of directional multi-level median filter denoising and wavelet transformation which can improve the readability of the coal mine image effectively. And this is of great significance for the safety monitoring in the coal mine.

## 6. ACKNOWLEDGEMENTS

This work was supported by grant Heilongjiang Provincial Department of Education Planning Project (NO. GBC1212076),

which is greatly acknowledged.

## 7. REFERENCES

- [1] V.Vonikakis and I.Andreadis. *Multi-Scale Image Contrast Enhancement*. 2008 10<sup>th</sup> Intl. Conf on Control, Automation, Robotics and Vision. Hanoi, Vietnam,17-20 December 2008.
- [2] S.M..Mahbubur Rahman, M.Omair Ahmad, M.N.S.Swamy. *Contrast-based fusion of noisy images using discrete wavelet transform*. IET Image Process., 2010, Vol.4, Iss.5, pp.374-384. doi:10.1049/iet-ipr.2009.0163
- [3] Du Li,Zhou Xin-ming, Li Zheng. *An Wavelet-based adaptive threshold image denoising method*. Digital Communication. 2010(4):55-57.(In Chinese)
- [4] Wang Ai-ling, Ye Ming-sheng, Deng Qiu-xiang. *MATLAB R2007 Image Process and Application*. Electronic Industry Press ,Beijing, 2008.(In Chinese)
- [5] PENG Bo, WANG Yi-ming. *Research and implementation of enhancement algorithm for low-illumination image*. Computer Applications, 2007, Vo.l 27 No. 8.(In Chinese)
- [6] LI Xin-feng. *A De-noising and Enhancing Method for Monitor Image in Coal Mine*. CHINA INSTRUMENTATION, 2010(4):48-49.(In Chinese)
- [7] ZHOU Jie. *The Modified Algorithms of Median Filtering Applicatied in Image Processing[D]*.Beijing University of Posts and Telecommunications, 2007:35-38.(In Chinese)
- [8] WANG Xiao-ming, HUANG Chang, ZHOU Chen-chen. *A Fast Adaptive Algorithm for Image Enhancement*. Computer System Application,2010(8):212-213.(In Chinese)
- [9] SHI Mei-hong, ZHANG Jun-ying, LI Yong-gang. *A New Method of Low Contrast Image Enhancement*. Application Research of Computers. 2005(1).235-237.(In Chinese)
- [10] GONG Xiao-lin, MAO Rui-quan, LIU Kai-hua. *Wavelet Images Threshold Value De-noising Based on Adaptive Neighboring Coefficient*. Computer Engineering.2010 (6):206-208.(In Chinese)

## About the author

Yanqin Zhao is a professor at Heilongjiang Institute of Science and Technology, Computer and Information Engineering College. Her contact email is [zyq\\_jean@sina.com](mailto:zyq_jean@sina.com)

# Data Processing for Managing Remote Sensing Images

Jianting Shi, Jiancai Wang, Fujun Zhao  
College of Computer and Information Engineering  
Heilongjiang University of Science and Technology  
Harbin, China  
hotmail8194@163.com

## Abstract

This paper discusses the establishment, management and application of remote sensing image database. A systematical structure for remote sensing image database is presented and analyzed. This architecture is based on web service technology, rather than traditional Browser/Server or Client/Server model. To process and publish mass remote sensing images on the web, appropriate system framework and software platforms are adopted in order to establish stable website and create web map service. In our system, Oracle is used to store the remote sensing images and GIS data, ArcGIS Server to create web services and accomplish web applications, and ArcSDE to manage the geospatial data. WebGIS applications and enterprise applications are carried out by utilizing and developing the geospatial database. The distributed database system provides a new model to archive and manage geospatial images effectively.

**Keywords:** *database management, remote sensing image database, web services*

## 1. INTRODUCTION

With the rapid development of remote sensing technology, the quantity of remote sensing images has increased sharply. Therefore, large commercial relational database management system (RDBMS) is currently adopted to store and manage remote sensing image data [1]. Besides large storage space, high-speed computational capabilities are needed to process and analyze mass remote sensing images [2]. Nowadays, some Geographic Information System (GIS) institutions and corporations have produced their own WebGIS software platform and spatial data engine to support RDBMS such as Oracle, SQL Server et al. At the same time, large GIS companies develop some software to provide web services of the geospatial data. The management and application of mass geospatial data is very important to web users. With the technologies of database and web services, mass geospatial data such as remote sensing images could be archived and managed effectively.

In our system, Oracle 10g is used to store the remote sensing images and GIS data, ArcGIS Server to create web services and accomplish web applications, and ArcSDE to manage the geospatial data. This paper discuss the establishment and management of remote sensing image database system based on web service architecture and demonstrates some applications, in which some key techniques are involved.

## 2. SYSTEM STRUCTURE

For the storage, management and utilization of mass remote sensing images, the traditional Client/Server architecture or Browser/Server architecture couldn't meet the demand effectively. Web service technology that is on Service-Oriented Architecture basis could enhance the application of remote sensing image database.

The W3C has defined a general-purpose web service architecture based on a trio of standards — Simple Object Access Protocol (SOAP), the Web Services Description Language (WSDL), and Universal Description, Discovery, and Integration (UDDI) [3]. Open Geospatial Consortium (OGC) has defined the Web Map Service (WMS), Web Feature Service (WFS), Web Coverage Service (WCS), and OGC Web Service Architecture, which support application developers in integrating a variety of online geoprocessing and location services [4].

WebGIS (also known as web-based GIS and Internet GIS) denotes a type of Geographic Information System, whose client is implemented in a Web browser. WebGIS technology has been developed and used extensively in real-world applications. However, such a complex web-based system involves the dissemination of large volumes of data and/or massive user interactions [5]. So, a stable and effective system structure should be designed and put in practice.

GIS web service clients are traditionally heavy-duty, stand-alone software tools. This made GIS web services more difficult to build than the ordinary business transactions for which general-purpose web services are originally intended [5]. Therefore, it is necessary to build a distributed processing system in which many heterogeneous databases, networks, and applications could be properly integrated.

Fig. 1 illustrates a geospatial data distributed processing system on web services architecture basis. In Fig. 1, Geospatial databases are composed of many heterogeneous databases which are located in different department within an enterprise to store geospatial data of different format. WFS, WMS, and WCS compose the data services. WPS (Web Processing Service) Register Center is the main part of this system for managing the web services. WFS, WMS, and WCS must be registered into WPS Register Center in order to have access to image processing clients in which logical applications are performed.

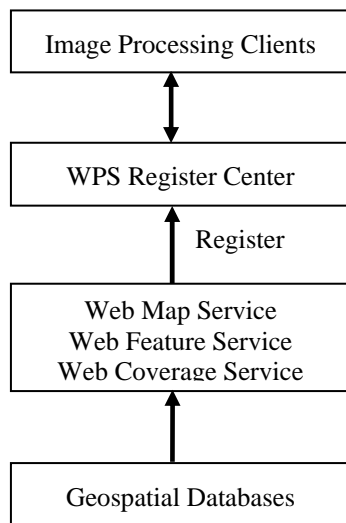


Figure 1: System structure based on web services architecture

### 3. METHODOLOGICAL APPROACH

To build remote sensing images, the database is applied in web services architecture. Following techniques should be under consideration.

#### 3.1 Data Processing and Uploading

Remote sensing data are temporal multi-dimensions, multi-sensors, multi-angle, multi-spectrum and multi-platform. Therefore, image processing such as data fusion, image mosaic and image enhancement should be done before uploading remote sensing image data into database. Web services architecture enables the processing of remote sensing data more easily. In the process of uploading, adjusting proper uploading parameters is of very importance.

#### 3.2 Querying and Retrieving

It is very important to quickly and effectively query data from the database. Remote sensing data differs from other data because of its characteristics of texture, color, and spatial geometry. The query conditions include not only date, location, bands, sensors et al, but the image content as well. Consequently, query standard and retrieve mechanism based on image content should be defined.

#### 3.3 Management and Utilization

Besides uploading and downloading, the remote sensing database should be further exploited for more utilization. With web services architecture, we develop a management information system and extend the utilization of remote sensing database from Intranet to Internet. The management information system could be based on traditional Browser/Server model, Client/Server model, or combination of these two models. With the increasing access demand of web users, web application is the first important utilization of remote sensing image database. Therefore, the management and utilization of remote sensing data should be based on web services technology.

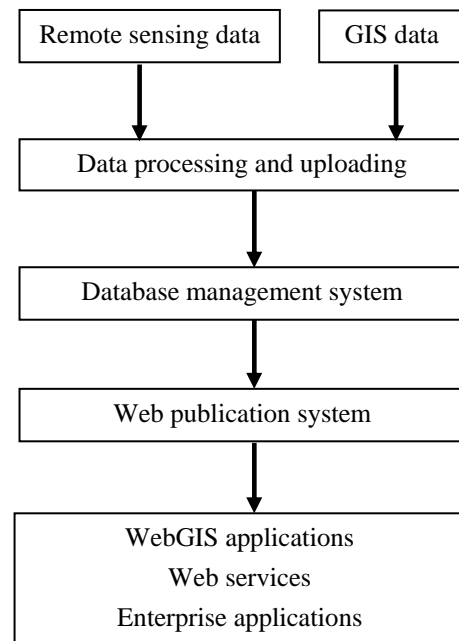


Figure 2: Workflow for geospatial data storage, management and application

Fig. 2 illustrates the workflow for geospatial data storage, management and application. In Fig. 2, mass remote sensing image data and Geographical Information System data may be processed in different departments of an enterprise. So, it is necessary to effectively manage the immense geospatial data [6].

Generally, image processing should be done by remote sensing image processing software such as ENVI, ERDAS IMAGINE et al. In our system, ArcSDE is used to upload geospatial data to database and then establish the remote sensing image database. In order to accomplish data management functions such as adding, deleting and updating, ArcObject and programming language are used to develop Database Management System. Web publication system is established with ArcGIS Server and then web applications could be developed with programming tools such as Java language. ArcGIS Server now supports web services technology, geoprocessing and globe publishing with its Web Application Developer Framework (ADF). As the web access privileges are set properly, the WebGIS applications, Web Service, and enterprise applications for internet users, intranet users and system manager are accomplished.

### 4. APPLICATIONS AND RESULTS

To archive and utilize remote sensing images effectively and systematically, we integrate the database management system and web publication system. The functions of remote sensing image database management system contain display, query, update et al. Web publication system made remote sensing image available on Internet and then web users could share the database ultimately.

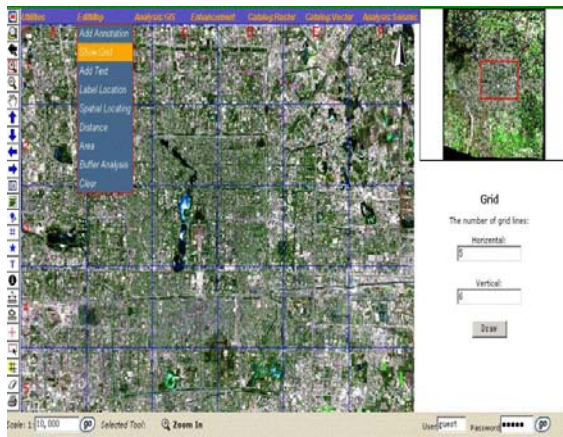


Figure 3: WebGIS application

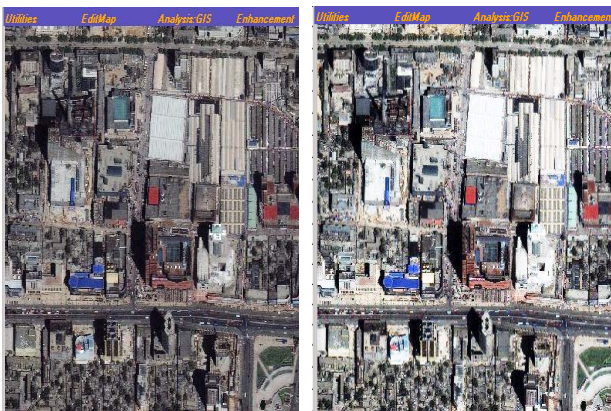


Figure 4: Online image processing

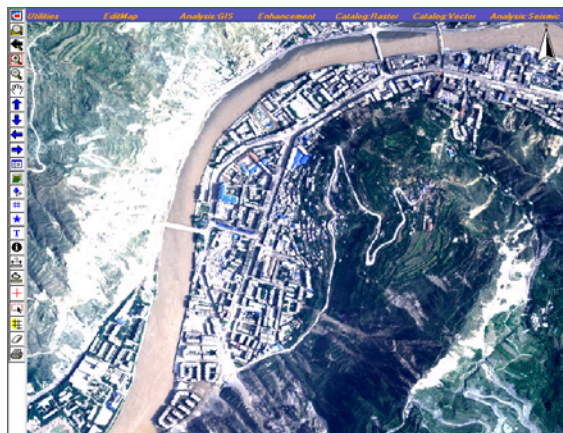


Figure 5: Enterprise application

In our system, we adopt Oracle 10g database which is commercial database software based on Grid technology. Data management such as uploading, updating and maintenance could be completed by developing ArcSDE software within internal network. Retrieving remote sensing images on Internet could be carried out with WebGIS software. The benefits of Web Service lie in facilitating web users to have access to GIS data and remote sensing images data stored in many database. These capabilities extend the power of geographic and descriptive data presentation, and provide the most flexible web publishing solution.

Fig.3, Fig.4, and Fig.5 illustrate some web applications of the distributed remote sensing image database.

Fig.3 represents WebGIS application. As the Fig.3 shows, the web publishing system provides online map edit such as adding annotation, adding grid, adding text, spatial locating, measuring distance, measuring area, buffering and so on.

Fig. 4 represents the function of online image processing. In Fig. 4, the left one is an original image and the right one is the enhanced image of the left one. The image processing clients could make these steps done according to the scheduler program of Web Processing Service (WPS) Register Center.

Fig.5 represents an enterprise application. It is a post-earthquake remote sensing image. Therefore, Web users could be well informed of the latest earthquake information and their applications. The web publication of mass earthquake information could help evaluate earthquake disaster and shorten time of first aid.

## 5. CONCLUSIONS

This paper discusses some techniques in archiving and managing mass remote sensing images. A distributed system structure based on web services technology is presented. In our system, geospatial images stored in Oracle database could be accessed, processed and managed using web services structure. WebGIS software platform ArcGIS Server and spatial data engine ArcSDE are utilized to publish mass remote sensing images on the Internet. The web services architecture provides a new model and powerful support for remote sensing images' distributed processing, management and web application.

## 6. REFERENCES

- [1] F.J. Zhao, J.F. Zhang, and D.Y. Cao. "Dynamic Database Connection and Dynamic Web Map Service for Internet Mapping," Proc. IEEE Symp. International Geoscience and Remote Sensing Symposium(IGARSS 05), IEEE press, July. 2005, pp. 3167-3169.
- [2] Shuo Dong, and Qiaoli Hu. "Building remote sensing database on Grid," Proc. IEEE Symp. International Geoscience and Remote Sensing Symposium(IGARSS 05), IEEE press, July. 2005, pp. 2105-2108.
- [3] D. Booth et al. "Web Services Architecture," W3C Working Group note 11, Feb. 2004.
- [4] J. Lieberman. "OpenGIS Web Services Architecture", Open Geospatial Consortium specification 03-025, Jan. 2003, pp. 21-36.
- [5] Chaowei Phil Yang, David Wong, Ruixin Yang, Menas Kafatos and Qi Li. "Performance-Improving Techniques in Web-Based GIS," International Journal of Geographical Information Science, vol. 19, March. 2005, pp. 319-342.
- [6] Mi Wang, Jianya Gong and Deren Li. "Design and Implementation of Large-Scale Image Database Management System," Geomatics and Information Science of Wuhan University, wuhan university, vol. 3, Sept. 2003, pp. 55-58.

## About the author

Jianting Shi is a lecturer at Heilongjiang University of Science and Technology, Computer and Information Engineering College. Her contact email is [hotmail8194@163.com](mailto:hotmail8194@163.com).







## Technical section (Russian)

## 3D SLAM ПО СТЕРЕОИЗОБРАЖЕНИЯМ

Бобков В.А., Роньшин Ю.И., Кудряшов А.П., Машенцев В.Ю.  
Институт автоматизации и процессов управления ДВО РАН  
{bobkov, ronshin, kudryashovA}@dvo.ru

### Аннотация

Предложены модификации метода визуальной навигации подводного аппарата по стереоизображениям, основанные на интеграции визуальных данных и измерений бортовой навигационной системы, а также на применении 6-ти облачной схемы вычислений. Приведены результаты вычислительных экспериментов с модельными данными.

**Ключевые слова:** стереопары изображений, SLAM, визуальная навигация, сопоставление особенностей, подводный аппарат, траектория

### 1. ВВЕДЕНИЕ

Задача SLAM (Simultaneous Localization and Mapping) продолжает оставаться сегодня актуальной в области компьютерного зрения и робототехники, поскольку от эффективности ее решения зависит точность навигации роботов в пространстве и реконструкции окружающей обстановки. Классическое решение SLAM, применяемое во многих работах, основывается на расширенном фильтре Калмана [1]. Применительно к подводным аппаратам (ПА) наряду с сенсорными данными, получаемыми традиционными средствами навигации (гидроакустические сенсоры, компасы, доплеровский лаг и др.) используются и видеоданные (захватываемый видеокамерами поток изображений). Особенно это важно в условиях локального маневрирования, когда традиционно применяемые гидроакустические сонары не всегда обеспечивают требуемую точность позиционирования подводного робота. В последующих работах рассматриваются постановки задач с использованием стереокамер, что позволяет совместно анализировать 2D информацию и получаемые 3D облака точек. Например, такой подход с применением фильтра Калмана развивался в работах [2,3] для навигации и реконструкции подводной обстановки и в работах [4,5] с вычислением геометрических преобразований между локальными системами координат ПА по сопоставленным 3D облакам.

В настоящем докладе предлагается метод навигации автономного ПА также основанный на сопоставлении точечных особенностей на стереоизображениях и анализе соответствующих 3D облаков точек при движении ПА по траектории. Его отличительными особенностями являются: адаптивная методика отбора видеоданных, применение многоступенчатой фильтрации особенностей, интеграция визуальных данных и показаний навигационной системы ПА, применение 6-ти облачной схемы вычислений. В разделе 2 дается описание визуального метода навигации и его модификаций. В разделе 3 приведены результаты вычислительных экспериментов и их анализ.

### 2. ВИЗУАЛЬНЫЙ SLAM

#### 2.1. Описание подхода

Решение задачи SLAM подразумевает точное вычисление траектории автономного робота (его 3D положение и ориентацию) при движении в априори неизвестной среде, с одновременным построением модели среды по получаемым сенсорным данным. В рассматриваемой постановке автономный подводный аппарат (АПА) оснащен двумя видеокамерами (стереопара), которые выполняют синхронную съемку дна при движении аппарата по траектории. Предлагаемый метод следует уже сложившемуся подходу, основанному на совместном использовании 2D особенностей на изображениях и 3D облаков точек, получаемых на базе стереопар. Вычислительная схема при определении параметров текущей позиции ПА состоит из двух последовательных этапов. На первом этапе анализируются две стереопары изображений, относящиеся к двум последовательным во времени позициям – текущей и предыдущей. Применительно к ним осуществляется поиск и сопоставление единого множества точечных особенностей, одновременно наблюдаемых на всех 4-х изображениях. Для поиска и построения дескрипторов особенностей используется детектор SURF. Для каждой из двух стереопар выполняется построение 3D точек - особенностей (3D облако) методом триангуляции, т.к. известна полная калибровка используемых камер. Заметим, что 3D точки в этих облаках сопоставлены в силу выполненного сопоставления их 2D преобразов. На втором этапе оценивается перемещение ПА из предыдущей позиции в текущую позицию. Поскольку мы наблюдаем одно и то же множество 3D точек из двух систем координат (СК) (текущая и предыдущая позиция ПА), то можно решить задачу вычисления геометрического преобразования (перенос и вращение) между СК этих позиций ПА. Для ее решения формулируется оптимизационная задача, где в качестве целевой функции рассматривается минимизация суммарных расхождений по всем точкам 3D облака. Ее решением является локальное (относительное) преобразование между двумя СК камер. Для получения абсолютного преобразования из мировой СК в СК текущей камеры достаточно объединить матрицы локальных преобразований всех предыдущих позиций ПА, начиная с первой. Вычисленные указанным образом матрицы преобразований позволяют получать в каждой позиции траектории ПА все 6 параметров, характеризующих положение аппарата в пространстве.

В рассматриваемом подходе на получаемую точность визуальной навигации негативно влияют два фактора: а) недостаточное количество точек в 3D облаках и б) ошибочные сопоставления (outliers). Поддержание гарантированного количества точек в 3D облаках обеспечивает адаптивный алгоритм выбора очередной обчитываемой позиции ПА на траектории. Устранение outliers делается как на этапе 2D обработки (пороговое отсечение, кросс-проверка), так и на этапе построения 3D облаков (эпиполярные ограничения в каждой стереопаре,

итерационная фильтрация точек на основе структурной когерентности облаков).

Существенное повышение точности навигации дает интеграция в описываемую вычислительную схему измерений бортовой навигационной системы. Используемые в бортовой навигационной системе ПА приборы (гироскопы, магнитные компасы) дают абсолютные значения углов ориентации, что позволяет воспрепятствовать накоплению ошибки со временем – характерному недостатку многих методов. Еще один элемент новизны предлагаемого подхода – рассмотрена и оценена эффективность альтернативной схемы вычислений с учетом обработки 6-ти облаков точек, которые возникают при анализе 3-х стереопар, соответствующих 3-м последовательным позициям ПА. Программная реализация выполнена с использованием CUDA-версии библиотеки OpenCV.

## 2.2. Сопоставление и фильтрация особенностей на изображениях

Сопоставление точечных особенностей выполняется на 4-х изображениях – 2-х стереопарах (1-2 и 3-4), соответствующих двум последовательным позициям ПА на траектории. Схема сопоставления построена на применении двух алгоритмов (библиотека OpenCV): а) детектора SURF для некалиброванных пар изображений (1-3 и 2-4). В этом случае выполняется построение дескрипторов и вычисление оценки расстояний между ними (с фильтрацией по порогу); и б) алгоритма вычисления диспарантности для калиброванных очищенных пар (1-2 и 3-4). Для каждой пары изображений выполняется сопоставление слева направо и справа налево (cross – checking). Результатом является множество особенностей, сопоставленных по всей цепочке из 4-х изображений. Это позволяет построить на следующем этапе два облака 3D точек, отвечающих двум стереопарам (двум позициям ПА).

*Адаптивная методика.* С учетом того, что от количества сопоставленных особенностей зависит точность последующей калибровки камер, была реализована адаптивная стратегия выбора очередной оцениваемой позиции на траектории ПА. Если для текущей позиции количество сопоставленных особенностей получилось меньше заданного порогового значения, то выбирается более близкая позиция, потенциально обеспечивающая большее число сопоставлений. Соответствующий алгоритм использует для этого переменный шаг во времени для фиксации удовлетворяющей этому условию позиции.

## 2.3. Вычисление положения и ориентации ПА по 3D облакам точек – особенностей

Изменения в положении и ориентации ПА при переходе в очередную позицию можно вычислить, опираясь на сравнение двух сопоставленных (по точкам) 3D облаков, одно из которых построено в системе координат первой стереопары (предыдущая позиция), а другое в системе координат второй стереопары (текущая позиция). Вычисление 3D точек каждого из облаков выполняется обычной триангуляцией (пересечение двух лучей в системе координат стереопары) по сопоставленным особенностям на двух снимках стереопары. Вычислительная задача сводится к решению переопределенной системы уравнений, где данными являются координаты наблюдаемых точек, а неизвестными являются элементы матрицы преобразования.

Пусть перемещение ПА из точки  $\mathbf{p}_{i-1}$  траектории в точку  $\mathbf{p}_i$  определяется некоторым неизвестным преобразованием  $\mathbf{H}_i$ , состоящим из вектора переноса и кватерниона вращения,  $i$  – номер позиции на траектории. Будем искать это преобразование, основываясь на имеющемся взаимно-однозначном сопоставлении двух облаков 3D точек, наблюдаемых парой камер, соответственно, в позициях  $(i-1)$  и  $i$ . Первое облако  $C^1(x,y,z)$  задано в СК 1-ой стереопары, 2-е облако  $C^2(x,y,z)$  – в СК 2-й стереопары. Задача нахождения матрицы  $\mathbf{H}_i$  решается методом нелинейной оптимизации с ограничением (используется библиотека MATLAB). В качестве параметров оптимизации используются 3 координаты вектора переноса и 4 координаты кватерниона, определяющего вращение  $\mathbf{H}_i$ . Ограничение задается условием – норма кватерниона = 1. Целевая функция  $F_i = \sum \| \mathbf{c}_k^2 - \mathbf{c}_k^1 \cdot \mathbf{H}_i \|^2$ ,  $k$  – номер точки в облаке. Здесь  $\{\mathbf{c}_k^1\}$  – множество точек в первом облаке и  $\{\mathbf{c}_k^2\}$  – множество точек во втором облаке. Тогда с учетом вычисленной матрицы  $\mathbf{H}_i$  новое положение ПА вычисляется через предыдущее как  $\mathbf{p}_i = \mathbf{p}_{i-1} \cdot \mathbf{H}_i$ . В качестве точки, описывающей траекторию ПА, можно рассматривать точку начала СК камеры (0,0,0). Поскольку нам нужны координаты точек траектории в мировой СК, необходимо вычислить преобразование из мировой СК в СК  $i$ -ой точки траектории. Если в качестве мировой СК выбрана СК первой пары в последовательности, то  $\mathbf{H}_0$  – единичная матрица (для произвольной мировой СК необходимо знать  $\mathbf{H}_0$ ). С учетом знания локальных преобразований для всех предшествующих точек траектории преобразование из мировой СК в СК  $i$ -той точки вычисляется как последовательное произведение локальных преобразований:  $\mathbf{H}_w = \mathbf{H}_0 \cdot \mathbf{H}_1 \cdot \dots \cdot \mathbf{H}_i$ . Соответственно, мировые координаты  $i$ -ой точки траектории ПА вычисляются как  $\mathbf{p}_i = \mathbf{p}_0 \cdot \mathbf{H}_w$ .

*Структурный фильтр.* Для исключения outliers из 3D облаков реализован итерационный фильтр, основанный на учете структурной когерентности облаков (сохраняющееся взаиморасположение точек в двух облаках). Алгоритмическая реализация аналогична описанной в [6]. Следует отметить, что такого рода фильтрация удаляет, в том числе, и особенности, связанные с динамическими объектами, характерными для подводной среды (рыбы и др. перемещающиеся объекты).

## 2.4. Интегрирование измерений навигационной системы АПА в вычислительную схему

Несмотря на сравнительно высокую точность вычисления локальных матриц преобразования, с течением времени происходит накопление ошибки в абсолютной матрице преобразования, поскольку последняя формируется перемножением локальных матриц предшествующих шагов. Наиболее весомой является ошибка в вычислении угла по курсу в начале траектории движения. Преодолеть указанный недостаток может привлечение дополнительной информации. Рассмотрим в качестве такой дополнительной информации измерение углов ориентации АПА, обеспечиваемое бортовой навигационной системой. Известно, что используемые в навигационных системах АПА приборы обеспечивают точность по углу курса от  $0.1^\circ$  (гироскоп) до  $1^\circ$  и более (магнитный компас) (для углов крена и дифферента эта точность выше). И хотя эта точность (как показали вычислительные эксперименты) ниже точности вычислений локальных изменений ориентации визуальным методом, преимущество компаса заключается в том, что он измеряет

абсолютные значения углов ориентации аппарата. Поэтому целесообразно интегрировать в схему вычислений данные абсолютных угловых измерений таким образом, чтобы, сохраняя преимущество визуального метода в вычислении локальных перемещений, препятствовать накоплению ошибки в вычислении абсолютной ориентации. В настоящей реализации это делается следующим образом.

На каждом шаге будем корректировать ту часть результирующей абсолютной матрицы преобразования, вычисленной визуальным методом, которая отвечает за вращение. Матрицу преобразования  $M$  между двумя системами координат (СК) можно представить как  $M =$

$$\begin{vmatrix} R & 0 \\ t & 1 \end{vmatrix}. \text{ Здесь матрица } R \text{ отвечает за вращение, а вектор } t -$$

за перенос точки. Известно, что, в общем случае,  $t = -d \cdot R$ , где  $d$  – вектор начала одной СК в другой СК. Пусть  $M_{\text{виз}} =$

$$\begin{vmatrix} R_{\text{виз}} & 0 \\ t_{\text{виз}} & 1 \end{vmatrix} \text{ будет абсолютной матрицей преобразования из}$$

мировой СК в СК камеры, вычисленной визуальным методом в текущей позиции. Как было описано выше, она получается умножением абсолютной матрицы предыдущей позиции на матрицу локального преобразования текущей позиции. Сформируем теперь скорректированную матрицу  $M_{\text{кор}} =$

$$\begin{vmatrix} R_{\text{кор}} & 0 \\ t_{\text{кор}} & 1 \end{vmatrix}, \text{ где } R_{\text{кор}} \text{ получена по угловым измерениям}$$

бортовой навигационной системы на данном шаге. Для вычисления  $t_{\text{кор}}$  воспользуемся приведенным выше соотношением, тогда  $t_{\text{виз}} = -d \cdot R_{\text{виз}}$  и  $t_{\text{кор}} = -d \cdot R_{\text{кор}}$ . Отсюда  $t_{\text{кор}} = t_{\text{виз}} \cdot R_{\text{виз}}^{-1} \cdot R_{\text{кор}}$ .

### 2.5. Сопоставление по 6-ти облакам

В описанном выше подходе вычисление искомого преобразования для текущей позиции ПА осуществлялось с учетом сопоставления особенностей применительно к двум стереопарам изображений, соответствующим предшествующей и текущей позиции аппарата. Логично предположить, что совместный анализ информации, связанной с тремя последовательными позициями ПА (3 стереопары изображений) может потенциально повысить точность расчетов. В этом случае при вычислении текущего преобразования могут согласовываться данные, как с предшествующей, так и с последующей позиции ПА. Для исследования и оценки эффективности такого подхода была реализована следующая схема вычислений. Пусть стереопара 1 соответствует предшествующей позиции, стереопара 2 – текущей позиции, для которой вычисляется преобразование, и стереопара 3 – последующей позиции. Для обработки выберем следующие пары стереопар (в каждой 4 изображения): 1-3; 1-2; 2-3. Для каждой из них будет выполняться обработка, описанная выше. То есть, будут построены 6 3D облаков точек. По полученным (6-ти) облакам вычисляется локальное преобразование (матрица  $H_j$ ) таким же образом, как и в случае 2-х облаков, но с другой целевой функцией. Пусть  $\{c_i^{1j}\}$  – множество точек в первом облаке,  $\{c_i^{2j}\}$  – множество точек во втором облаке для  $j$ -той пары стереопар ( $j = 0, 1, 2$ ). Целевой функцией в этом случае будет:

$$F = \sum \|c_i^{20} - c_i^{10} * H12\| + \sum \|c_i^{21} - c_i^{11} * H23\| + \sum \|$$

$$c_i^{22} - c_i^{12} * H13\|$$

Искомым преобразованием является  $H_j = H12$ .

### 3. РЕЗУЛЬТАТЫ ВЫЧИСЛИТЕЛЬНЫХ ЭКСПЕРИМЕНТОВ

Вычислительные эксперименты проводились на PC с графической платой NVIDIA GeForce GTX 470 на ряде модельных сцен с траекториями различной формы и с разными глубинами (от 1м до 10м). Вид одной из сцен показан на рис. 1. Время движения ПА по траектории - 130 сек. Базовое расстояние между камерами стереопары виртуального ПА - 40см. Захват кадров осуществлялся с

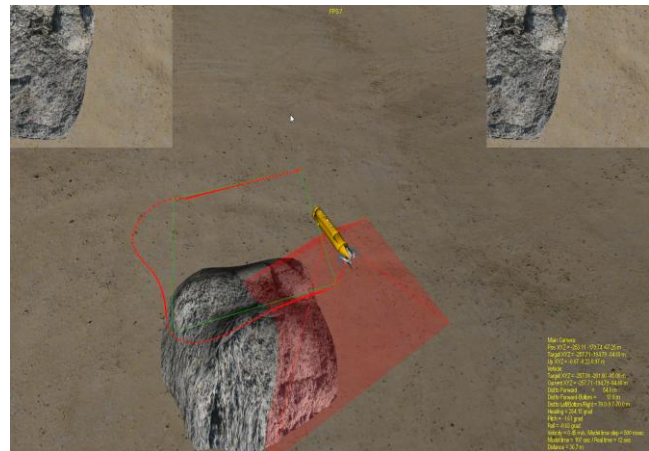
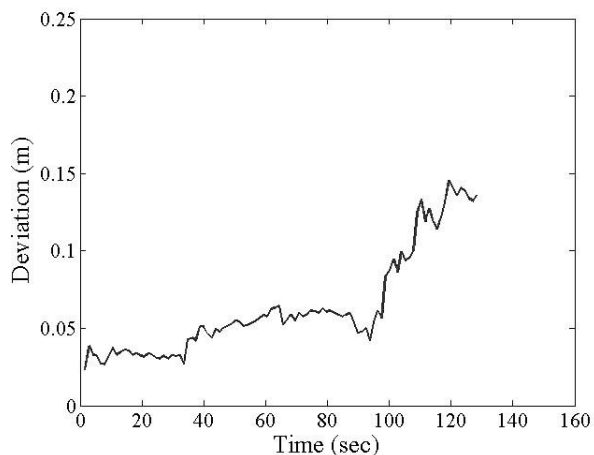
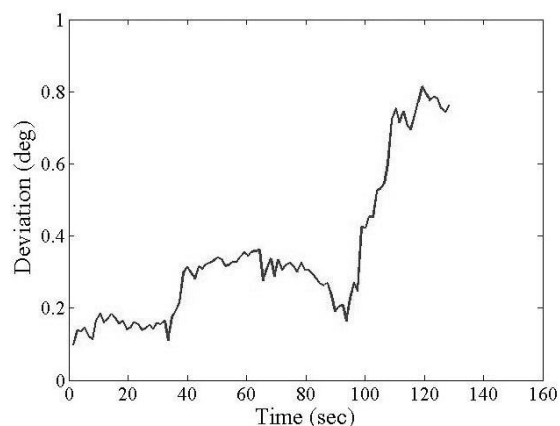


Рис.1. Модельная сцена.

частотой 25 кадров/сек. Разрешение захватываемых изображений 600x400 пикселей. Выбор расчетных позиций на траектории делался с шагом (максимальным) 16 кадров и 32 кадра. Оценивалась точность метода путем сравнения вычисленной траектории с истинной для разных вариантов алгоритмической реализации: 1) базовый (по 2-м облакам); 2) базовый с учетом угловых показаний компаса; 3) 6-ти облачный алгоритм. Аналогичным образом оценивалась точность ориентации ПА (среднее отклонение по 3-м углам). Приводимые ниже результаты получены на реализации программного комплекса с использованием CUDA-версии библиотеки OpenCV. Эксперименты проводились на PC с графической платой NVIDIA GeForce GTX 470. На рис. 2,3 показаны, соответственно, график поведения ошибки  $ds$  (отклонение вычисленной траектории от истинной) и график поведения ошибки  $da$  (усредненное угловое отклонение) в зависимости от времени движения по траектории (для сцены на рис.1). Среднее отклонение по углам  $0.3^\circ$ . Эксперименты показали, что достигаемые точности для шага 16 кадров и 32 кадра практически одинаковы. В таблице 1 приведены средние (по всей длине траектории) значения ошибки  $ds$  для вышеуказанных вариантов алгоритма и общее время вычислений. Использование показаний компаса бортовой навигационной системы дает для базового варианта алгоритма повышение точности до 20%. Анализ показывает, что, 6-ти облачный алгоритм, в сравнении с базовым, дает повышение точности до 40%, однако в 4.5 раза уступает ему по времени.



**Рис. 2.** Ошибка вычисления траектории базовым алгоритмом (шаг по траектории 32 кадра).



**Рис.3.** Ошибка в вычислении углов ориентации ПА.

	Базовый алгоритм	Алгоритм с учетом измерений углов	6-ти облачный алгоритм
Средняя ошибка ds (см)	6.3	5.1	4.2
Время (сек) вычислений по всей траектории	16	16	73

**Табл. 1.** Сравнительная эффективность модификаций метода визуальной навигации ПА. Вычисление параметров траектории в позициях, выбираемых с шагом 32 кадра.

Следует заметить, что приведенные результаты получены в режиме выключенного структурного фильтра, поскольку при незначительном приросте в точности (порядка 10%) он существенно увеличивает время вычислений. Как следует из экспериментов, согласно данному подходу ошибка в вычислении траектории для базового варианта оценивается в среднем в 0.6% по отношению к расстоянию до дна.

## ЗАКЛЮЧЕНИЕ

Результаты вычислительных экспериментов подтвердили потенциальную эффективность предложенных модификаций

визуального подхода к навигации ПА. В дальнейшем планируется оптимизация вычислительной схемы для ускорения вычислений и проведение натурных экспериментов с установленным ПО на борту ПА. Работа поддержана грантом РФФИ (проект № 11-07-00088-а) и грантом ДВО РАН (проект 12-1-П115-04).

## СПИСОК ЛИТЕРАТУРЫ

- [1] G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte and M. Csorba. A Solution to the Simultaneous Localization and Map Building (SLAM) Problem. IEEE Trans. on Robotics and Automation, vol. 17, no. 3, June 2001, pp. 229-241.
- [2] Бобков В.А., Борисов Ю.С. Восстановление траектории подводного аппарата и структуры среды по изображениям // Мехатроника, автоматизация, управление, №9, 2006, С. 25-31.
- [3] J. Salvi, Y. Petillot, E.Battle. Visual SLAM for 3D Large – Scale Seabed Acquisition Employing Underwater Vehicles // IEEE International Conference on Intelligent Robots and Systems, 2008, pp. 1011-1016.
- [4] Miguel Angel Garcia, Agusti Solanas. 3D Simultaneous Localization and Modeling From Stereo Vision // International Conference on Robotics and Automation - ICRA, 2004, vol. 1, pp. 847-853.
- [5] Matthew Johnson-Roberson, Oscar Pizarro, Stefan B. Williams, Ian Mahon. Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys // Journal of Field Robotics, Special Issue: Three-Dimensional Mapping, Part 3, 2010, Volume 27, Issue 1, pp. 21–51.
- [6] J.M. Saez, A. Hogue, F. Escolano, M. Jenkin. Underwater 3D SLAM through Entropy Minimization // Proceedings 2006 IEEE International Conference on Robotics and Automation, ICRA 2006, pp. 3562-3567.

## Об авторах

Бобков Валерий Александрович, доктор технических наук, зав. лабораторией машинной графики Института автоматизации и процессов управления ДВО РАН.

E-mail: bobkov@iacp.dvo.ru

Роншин Юрий Иванович, ведущий инж.-прогр. лаборатории машинной графики ИАПУ ДВО РАН.

E-mail: ronshin@iacp.dvo.ru

Кудряшов Алексей Павлович, младший научный сотрудник ИАПУ ДВО РАН.

E-mail: kudryashova@iacp.dvo.ru

Машенцев Владимир Юрьевич, аспирант, ИАПУ ДВО РАН;

E-mail: v.mashentsev@gmail.com.

## 3D SLAM from stereo vision

### Abstract

We propose modifications of the visual navigation method of the underwater vehicle by stereo images based on the integration visual data and board navigation measurements, as well as the use of 6 cloud computation scheme.

The results of computational experiments with synthetic data are given.

**Keywords:** stereo vision, SLAM, visual navigation, feature matching, underwater vehicle, trajectory

# Распознавание динамических жестов руки посредством обработки дальностных изображений человека

Ваагн Нагапетян

Факультет физико-математических и естественных наук  
Российский университет дружбы народов, Москва, Россия

## Аннотация

Рассматривается задача распознавания динамических жестов руки человека для создания человеко-машинного интерфейса, в котором взаимодействие осуществляется посредством естественных жестов руки без прикосновения к каким либо контроллерам и сенсорным экранам. Распознавание позиции и ориентации рук осуществляется посредством обработки каждого кадра видеоряда, полученного от трехмерного сенсора. Идентификация жеста осуществляется посредством сравнения траекторий центра ладони с траекториями эталонных жестов. Для сравнения траекторий жестов используется алгоритм динамической трансформации шкалы времени (Dynamic Time Warping - DTW).

Рассматриваются две системы, реализованные на основе предложенных алгоритмов, позволяющие бесконтактно рисовать на экране персонального компьютера с помощью жестов рук и пальцев, а также распознавать жесты, траектории которых представляют собой геометрические фигуры или цифры.

**Ключевые слова:** Распознавание жестов, дальностное изображение, DTW

## 1. ВВЕДЕНИЕ

Распознавание жестов рук является довольно актуальной задачей в таких приложениях, как например, автоматизированный сурдоперевод; управление компьютером, роботом или искусственной рукой; естественное взаимодействие с трехмерными компьютерными моделями объектов и т.д. Подходы к решению данной задачи отличаются друг от друга используемой аппаратурой и алгоритмами обработки данных о жесте руки. Например, в работе [1] для распознавания позиции руки и пальцев используется цветная камера. Изображение рук отделяется от фона с учетом отличия цвета кожи человека от заднего фона. Полученное изображение сглаживается методом медианной фильтрации, контуры руки выделяются с применением алгоритма поиска контуров связанных компонент, а пальцы выделяются на основе анализа изгибов контура руки. В предложенной системе жесты руки используются для управления видеокamerой посредством взаимодействия с графическим интерфейсом пользователя. В работе [2] для распознавания динамических жестов руки система сначала обучается на моделях 20 жестов, используя скрытую Марковскую модель (СММ), где в качестве дескрипторов жеста выступают коэффициенты Фурье. В работе приводится точность распознавания - 90%. Существуют также подходы, основанные на применении перчаток, оснащенных сенсорами [3], анализе дальностных

изображений и распознавании позиции руки посредством использования деревьев решений [4].

В настоящей работе в качестве устройства ввода жестов был выбран трехмерный сенсор Asus Xtion Pro Live [5], который снабжен одной RGB камерой, излучателем структурированного инфракрасного света и приемником, который принимает отраженный свет от поверхностей объектов. В результате сенсор возвращает цветное изображение и дальностное изображение с разрешением 640x480 со скоростью 30 кадров в секунду, что вполне приемлемо для создания приложений, работающих в реальном времени.

## 2. ЗАДАЧА РАСПОЗНАВАНИЯ ДИНАМИЧЕСКИХ ЖЕСТОВ РУКИ

В зависимости от выбора человеко-машинного интерфейса, задачей распознавания жеста руки может быть:

1. Вычисление позиции руки/ладони.
2. Определение ориентации ладони.
3. Идентификация жеста по заданным эталонным образцам.

В системах, требующих непосредственного управления персональным компьютером, роботом или виртуальной кистью с помощью жестов рук, достаточным является вычисление позиции и ориентации руки. При этом должны учитываться такие факторы, как различие форм рук и цвета кожи у разных людей, изменение освещения и возможные изменения заднего фона наблюдаемого человека. Поскольку такие системы требуют мгновенного отклика от графического интерфейса пользователя, время обработки каждого кадра видеоряда не должно превышать 1/24 секунды.

В системах автоматизированного сурдоперевода, задачей распознавания является идентификация показанного жеста по заданным эталонным образцам. В данном случае должны учитываться разные скорости показа жеста и возможные отклонения распознаваемых жестов от эталонных образцов.

Далее рассмотрим основные этапы разработанного алгоритма для решения приведенных задач.

## 3. ВЫЧИСЛЕНИЕ ПОЗИЦИИ И ОРИЕНТАЦИИ ЛАДОНИ В КАДРЕ ВИДЕОРЯДА

Кадр видеоряда трехмерного сенсора представляет собой дальностное изображение, каждый пиксель которого характеризуется расстоянием до камеры наблюдения. На жестикулирующего человека ставится естественное ограничение – чтобы система рассматривала руку как управляющее устройство, расстояние руки до трехмерного сенсора должно быть не больше фиксированного значения  $d$ .

Для каждого кадра видеоряда выполняются следующие действия:

- (1) Удаление всех точек, которые не входят в рассматриваемую зону (пороговая обработка);
- (2) Поиск связанных компонентов;
- (3) Вычисление центров компонентов;
- (4) Фильтрация компонентов.

Первый шаг алгоритма осуществляется путем сравнение значения каждой точки кадра с заранее известным значением  $d$ . Если значение точки превышает  $d$ , то точке присваивается значение ноль. Обозначим полученное после пороговой обработки дальностное изображение буквой  $S$ .

**Определение 1.** В изображении  $S$  назовем точки  $(x, y)$  и  $(x', y')$  связанными, если существует последовательность точек  $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$  таких что,  $(x_0, y_0) = (x, y), (x_n, y_n) = (x', y')$ , точки  $(x_{i-1}, y_{i-1})$  и  $(x_i, y_i)$  являются 8-связанными соседями и выполняются условия  $S(x_{i-1}, y_{i-1}) > 0, S(x_i, y_i) > 0$  для всех значений  $i \in \{1, 2, \dots, n\}$ .

**Определение 2.** Связанной компонентой  $C$  в изображении  $S$  называется множество точек  $C = \{(x, y) : S(x, y) > 0\}$ , где любые две точки из  $C$  являются связанными друг с другом и все связанные точки  $(x', y')$  с точкой  $(x, y) \in C$  принадлежат множеству  $C$ .

Поиск связанных компонентов осуществляется в два прохода, посредством алгоритма поиска связанных компонент в графе [6]. Во время первого прохода все точки изображения  $S$  помечаются временными метками, где метки представляют собой цифровые значения. Параллельно, создается множество эквивалентных меток. Например, на рисунке 1 метки 2 и 3 являются эквивалентными. Во втором проходе все временные метки меняются на метку с минимальным значением из числа эквивалентных меток. Например, на рисунке 1 метка 3 поменяется на метку 2. После второго прохода, множество точек, помеченных эквивалентными метками, будет представлять собой связанную компоненту в изображении  $S$ . Например, на рисунке 1 множество точек, помеченных меткой "1" и меткой "2" являются связанными компонентами в приведенном изображении.

•	•	•	•	0	0	0	1	1	1	1	0	0	0	1	1	1	1	0	0	0	
•	•	•	•	0	0	0	1	1	1	1	0	0	0	1	1	1	1	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	•	•	0	•	•	•	0	2	2	0	3	3	0	0	2	2	0	2	2	0	
0	•	•	0	•	•	•	0	0	2	2	0	3	3	0	0	2	2	0	2	2	0
0	•	•	•	•	0	0	0	0	2	2	2	2	0	0	0	2	2	2	2	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Рисунок 1 – Пример маркировки точек связанных компонент в дальностном изображении. Слева направо показаны соответственно дальностное изображение, где точками обозначены пиксели, значения которых больше нуля; маркировка после первого прохода и маркировка после второго прохода.

Обозначим через  $K(x, y)$  метку точки  $(x, y)$  в изображении  $S$ . Следующим шагом алгоритма является вычисление центров найденных связанных компонентов. Центры вычисляются отдельно для каждого компонента посредством моментов. Пусть  $C$  - связанная компонента в изображении  $S$ , точки которого были маркированы меткой  $c$ .

Определим моменты первого порядка  $M_{0,0}, M_{0,1}, M_{1,0}$  компонента  $C$  следующим образом:

$$M_{0,0} = \sum_x \sum_y I(x, y), \quad M_{0,1} = \sum_x \sum_y y \cdot I(x, y),$$

$$M_{1,0} = \sum_x \sum_y x \cdot I(x, y), \quad \text{где}$$

$$I(x, y) = \begin{cases} 1, & \text{если } K(x, y) = c \\ 0, & \text{в противном случае} \end{cases}$$

Вычислить центр компонента  $C$  можно следующим образом:

$$(x_c, y_c, z_c) = \left( \frac{M_{1,0}}{M_{0,0}}, \frac{M_{0,1}}{M_{0,0}}, S(x_c, y_c) \right). \quad \text{Ориентация}$$

ладони может быть вычислена посредством моментов второго порядка.

Следующим шагом алгоритма является фильтрация найденных связанных компонентов. Из всех компонентов удаляются:

- 1) компоненты, размер которых слишком мал, чтобы быть изображением ладони человека;
- 2) компоненты, центры которых не меняют расположения в течение времени. Примечание: для отслеживания местоположения центров рассматривается фиксированное число предыдущих кадров видеоряда.

На основе предложенного алгоритма было разработано программное приложение, позволяющее создавать рисунки на персональном компьютере посредством динамических жестов руки. На рисунке 2 показан интерфейс программы. Объекты разного цвета на рисунке 2(a) - это распознанные связанные компоненты в кадрах видеоряда. Для каждого компонента цвет выбирается случайным образом, но так, чтобы в одном видеокадре два разных компонента не имели одинакового цвета. Распознавание позиции руки в каждом кадре видеоряда осуществляется в течение нескольких миллисекунд, что позволяет рисовать на экране компьютера в реальном времени, без ощущения задержки отклика. Заметим, что количество людей и рук не ограничено. Видео-демонстрация программы доступна в [7].

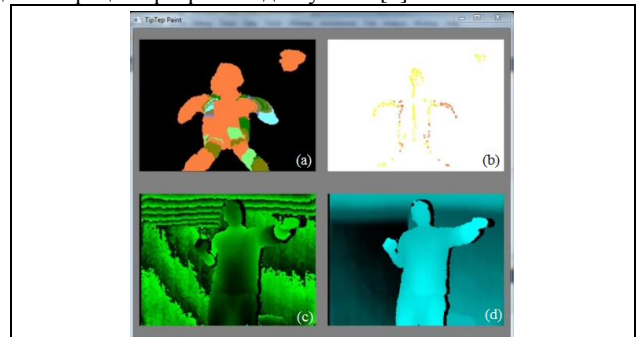


Рисунок 2 – Интерфейс программы для рисования с помощью жестов руки.  
 (а) Множество всех распознанных компонентов во всех кадрах видеоряда. (б) Множество центров всех распознанных компонентов во всех кадрах видеоряда. (с) Дальностное изображение. (д) Сглаженное дальностное изображение.

#### 4. ИДЕНТИФИКАЦИЯ ДИНАМИЧЕСКИХ ЖЕСТОВ РУКИ

Задачей идентификации жеста является поиск эталонного образца, который имеет минимальное расстояние до показанного жеста. В случае динамических жестов задача идентификации усложняется неопределенностью начала и конца показанного жеста. Обойти эту трудность можно посредством наложения ограничений на жестикулирующего человека – началом жеста определить время, когда расстояние руки от сенсора становится меньше чем заданное значение  $d$  и концом жестикуляции - время, когда расстояние становится больше чем  $d$ . В этом случае вычислить координаты центров ладоней можно посредством алгоритма приведенной в разделе 3.

Идентификация динамического жеста осуществляется в два этапа:

1. Создание эталонных жестов.
2. Сопоставление показанного жеста с эталонными образцами.

Представим динамический жест в виде временного ряда (рис. 3).



Обозначим буквой  $P$  временной ряд  $\{p_1, p_2, \dots, p_m\}$ , где  $p_1 = (x_1, y_1)$  представляет собой координаты центра ладони в первом кадре видеоряда и  $p_m = (x_m, y_m)$  - в последнем кадре. Заметим, что цифра  $m$ , представляющая собой количество кадров в видеоряде при показе одного жеста, может меняться во время разных показов одного и того же жеста.

Сопоставление двух динамических жестов осуществляется путем нормализации и вычисления расстояния между соответствующими временными рядами с помощью алгоритма динамической трансформации шкалы времени.

Введем обозначения:  $S_x = \min\{x_1, x_2, \dots, x_m\}$ ,  
 $E_x = \max\{x_1, x_2, \dots, x_m\}$ ,  $S_y = \min\{y_1, y_2, \dots, y_m\}$ ,  
 $E_y = \max\{y_1, y_2, \dots, y_m\}$ ,  $D_x = 1/(E_x - S_x)$ ,  
 $D_y = 1/(E_y - S_y)$ ,  $C_x = (E_x + S_x)/2$ ,

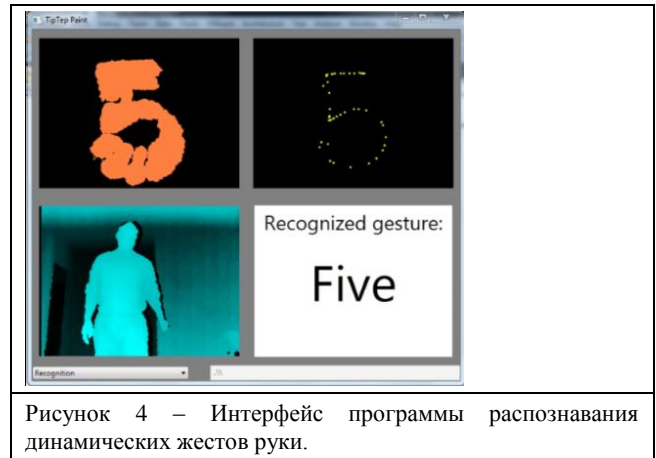
$C_y = (E_y + S_y)/2$ . Нормализованный временной ряд записывается в виде  $P' = \{p'_1, p'_2, \dots, p'_m\}$ , где  $p'_i = ((x_i - C_x) \cdot D_x, (y_i - C_y) \cdot D_y)$ , для всех значений  $i \in \{1, 2, \dots, m\}$ . Сравнение двух нормализованных временных рядов  $P' = \{p'_1, p'_2, \dots, p'_m\}$  и  $Q' = \{q'_1, q'_2, \dots, q'_n\}$  осуществляется посредством применения алгоритма DTW [8]. Для заданных временных рядов строится матрица расстояний  $A \in R^{m \times n}$  на метрике Манхэттена:

$$a_{i,j} = d(p'_i, q'_j) = |p'_i(x) - q'_j(x)| + |p'_i(y) - q'_j(y)|$$

Следующим шагом является поиск пути в матрице  $A \in R^{m \times n}$ , начинающегося с элемента  $a_{1,1}$  и достигающего

элемента  $a_{m,n}$ , для которого сумма значений элементов данного пути минимальна. Найти такой путь за полиномиальное время можно посредством алгоритма динамического программирования. Для матрицы  $A \in R^{m \times n}$  создается новая матрица  $B \in R^{(m+1) \times (n+1)}$ . Элементу  $b_{1,1}$  присваивается значение 0, а прочим элементам первой строки и первого столбца матрицы  $B$  - значение  $\infty$ . Остальные элементы матрицы вычисляются следующим образом:  $b_{i,j} = a_{i,j} + \min\{b_{i-1,j}, b_{i,j-1}, b_{i-1,j-1}\}$ . В качестве коэффициента схожести двух временных рядов выбирается значение элемента  $b_{m+1,n+1}$ .

На основе предложенного алгоритма была разработана система распознавания динамических жестов руки человека. Система распознает 12 жестов – 10 цифр и две геометрические фигуры. Человек с помощью движения руки рисует цифры и геометрические фигуры. Каждый жест сопоставляется со всеми эталонными жестами. В качестве распознанного жеста выбирается эталонный жест, который имеет наибольший коэффициент схожести с показанным жестом. Интерфейс реализованной системы показан на рисунке 4.



Заметим, что направление показа жеста влияет на результаты распознавания. Например, если показать цифру ноль по



часовой стрелке и против часовой, то программа примет эти жесты за два разных. Распознать жесты независимо от направления движения руки можно посредством хранения для каждого эталонного жеста временных рядов, выполненных по разным направлениям.

## 5. РЕЗУЛЬТАТЫ

Предложенный алгоритм распознавания динамических жестов руки был протестирован на базе жестов двух разных людей. Тестовая база включала в себе 2400 жестов двух человек, показывающие цифры от нуля до девяти и две геометрические фигуры – квадрат и треугольник. В качестве эталонных жестов из тестовой базы произвольным образом были выбраны 12 образцов, по одному для каждого класса. В таблице 1 приведены характеристики качества распознавания алгоритма, где точность распознавания определяется как доля жестов действительно принадлежащих данному классу относительно всех жестов, которые система отнесла к этому классу. Полнота распознавания определяется как доля найденных классификатором жестов принадлежащих классу относительно всех жестов этого класса в тестовой выборке.

Таблица 1 – Характеристики качества распознавания

Характеристики качества распознавания	Тестовая выборка											
	0	1	2	3	4	5	6	7	8	9	Δ	□
Точность	.75	.83	1	1	.79	.82	.81	1	1	.73	.88	.95
Полнота	.88	.98	.86	.9	1	.76	1	1	.7	.64	1	.7

Из таблицы 1 видно, что средняя точность распознавания составляет 88 процентов, а средняя полнота – 87, что является приемлемым результатом для использования предложенного алгоритма во многих системах взаимодействия человека с машиной. Ошибки классификатора можно объяснить как ошибками оператора во время жестикуляции, так и схожестью временных рядов некоторых жестов. Количество шагов, требуемых для сравнения двух жестов с числом кадров  $m$  и  $n$  соответственно, оценивается как  $O(m \times n)$ , что позволяет распознать жест сразу же после завершения его показа.

## 6. ЗАКЛЮЧЕНИЕ

Выполненные эксперименты показывают, что предложенные алгоритмы и методы могут быть использованы для создания новых типов человеко-машинного интерфейса. Они могут быть расширены и использованы для рисования с помощью жестов рук как альтернатива сенсорным экраном и как способ перевода жестового языка глухонемых на естественный язык. В дальнейшем планируется создать прототип программы, которая позволит распознавать статические и динамические жесты языка глухонемых.

## 7. БЛАГОДАРНОСТИ

Работа выполнена при поддержке проекта РФФИ №13-07-00025 А.

## 8. ССЫЛКИ

- [1] Chu S., Tanaka J. Hand Gesture for Taking Self Portrait // Proceedings of the 14th international conference on Human-computer interaction: interaction techniques and environments - Part II. Springer-Verlag: 2011. — P.238-247
- [2] Chen F., Fu C., Huang C. Hand gesture recognition using a real-time tracking method and hidden Markov models // Journal Image and Vision Computing. 2003. — P. 745-758
- [3] Kim J., Thang N., Kim T. 3-D hand motion tracking and gesture recognition using a data glove // Industrial Electronics, 2009. ISIE 2009. IEEE International Symposium on. 2009. — P.1013-1018
- [4] Shotton J., Fitzgibbon A., Cook M., Sharp T., Finocchio M., Moore R. Real-time human pose recognition in parts from single depth images // Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society: 2011. — P. 1297-1304
- [5] Asus Xtion Pro Live. URL: [http://www.asus.com/Multimedia/Xtion\\_PRO\\_LIVE/](http://www.asus.com/Multimedia/Xtion_PRO_LIVE/)
- [6] Shapiro L. Computer Vision. New Jersey: Prentice Hall, 2001. — p.608
- [7] TipTap — Humanizing of digital environment. URL: <http://tiptep.com/index.php/research>
- [8] Theodoridis S., Pikrakis A., Koutroumbas A., Cavouras D. Introduction to Pattern Recognition: A Matlab Approach. Academic Press, 2010 — p.231

## Об авторах

Ваагн Нагапетян – аспирант кафедры информационных технологий факультета физико-математических и естественных наук Российского университета дружбы народов. Его адрес: [vahagnahapetyan@gmail.com](mailto:vahagnahapetyan@gmail.com)

# Система локализации робота на основе полусферической камеры

Александр Овчинников, Хоа Фан  
Кафедра Радиоэлектронники

Тульский Государственный Университет, Тула, Россия  
[admin\\_telex@mail.ru](mailto:admin_telex@mail.ru), [dolphin22a@yahoo.com](mailto:dolphin22a@yahoo.com)

## Аннотация

В данной статье рассматривается задача локализации робота с помощью комбинированной системы на основе одометра и полусферической камеры. Измерения азимутальных углов выполненные на основе данных от полусферической камеры корректируют накапливаемую погрешность одометра по предложенному алгоритму, основанному на расширенном фильтре Калмана. На основе Unscented Transform предложен алгоритм инициализации вертикальных линий, который обеспечивает более точную оценку плотности распределения вероятностей координат вертикальных линий по сравнению с часто используемым фильтром Калмана.

**Ключевые слова:** локализация, полусферическая камера, расширенный фильтр Калмана, Unscented Transform.

## 1. ВВЕДЕНИЕ

Для управления мобильными объектами (далее роботами) в автономном режиме главной является задача локализации, которая позволяет оценивать текущие координаты робота с максимальной возможной точностью при наличии различного рода шумов и помех. Данная задача часто решается с помощью встроенных в роботе датчиков (например, одометр, датчики расстояний) или внешних ориентиров. Самостоятельное использование систем локализации на базе датчиков расстояний или одометра затруднительно из-за наличия накапливаемой во времени погрешности по определению параметров движения робота. Это, в свою очередь, породило разработку комбинированных систем. Принцип работы таких систем, в основном, основан на калмановской фильтрации, которая рекурсивно оценивает вектор состояний системы по последовательной двухэтапной процедуре - экстраполяции и коррекции.

В последнее время всё больший интерес вызывает разработка комбинированных систем на основе систем технического зрения (СТЗ). Однако двумерные изображения, формируемые СТЗ, не позволяют полностью решить задачу локализации, т.к. они не содержат информации о глубине изображения. Для устранения данного недостатка требуются два или более изображения, полученные с различных ракурсов для определения координат интересующих объектов, что влечет за собой использование в СТЗ нескольких камер размещенных в пространстве. Модель измерения на основе СТЗ с двумя и более камерами характеризуется высокой степенью нелинейности, которая приводит к неточной оценке плотности распределения вероятностей измерений по методу расширенного фильтра Калмана [1]. Кроме того, точность определения координат объектов зависит от смещения между кадрами, которое связано с углом параллакса между позициями СТЗ до объекта.

В работе [1] представлен стандартный алгоритм локализации на основе СТЗ, но не рассмотрено влияние нелинейности модели наблюдения на результат обработки алгоритма.

Игнорирование нелинейностей и предельное упрощение ситуации может существенно снизить эффективность алгоритмов оценивания координат в реальных системах локализации. В работе [2] предложен метод оценки одномерной плотности распределения вероятностей наблюдений, который по времени наблюдения улучшает её аппроксимацию, но при этом требует высоких вычислительных затрат.

В данной работе предлагается алгоритм локализации робота на основе комбинированной системы включающей одометр и полусферическую камеру. Рассмотрены вопросы инициализации характерных объектов, приведены результаты экспериментальных исследований.

## 2. РАЗРАБОТКА РАСШИРЕННОГО ФИЛЬТРА КАЛМАНА ДЛЯ ЛОКАЛИЗАЦИИ РОБОТА

Известные системы локализации при обработке информации от датчиков опираются на оптимальную фильтрацию на основе фильтра Калмана, который рассматривает состояние системы как гауссовое распределение, и постоянно оценивает математическое ожидание и ковариационную матрицу его апостериорной плотности распределения вероятностей [3]. Для нелинейных систем используется расширенный фильтр Калмана - нелинейная система линеаризуется с помощью разложения в ряд Тейлора с последующим отбрасыванием членов ряда выше первой степени.

Использование полусферической камеры в системе локализации робота подразумевает трансформацию известных алгоритмов обработки данных на основе расширенного фильтра Калмана с целью обеспечения возможности использования последовательных измерений выполняемых видеокамерой для формирования оптимальной оценки состояния робота.

Объектами измерений полусферической камеры являются вертикальные линии в пространстве, изображение которых имеют вид радиальной линии на плоскости изображения, формируемого полусферической камерой. Вертикальные линии выделяются на каждом кадре изображения и идентифицируются между кадрами (рис.1) на основе алгоритма ранее разработанного авторами. Это позволяет инициализировать координаты вертикальных линий и корректировать априорные измерения.

Пусть состояние робота описывается вектором  $\mathbf{x}_R = [x_r \ y_r \ \phi_r]^T$ , где  $x_r$ ,  $y_r$ ,  $\phi_r$  - соответственно координаты и направление робота глобальной системе координат; под координатами вертикальных линий  $\mathbf{x}_L = [x_L \ y_L]^T$  подразумевается проекция данных линий на горизонтальную плоскость. Вектор состояний системы в шаге  $k$   $\mathbf{x}[k] = [x_R^T[k] \ x_{L_1}^T[k] \ \dots \ x_{L_n}^T[k]]^T$  состоит из состояния робота и координат совокупности наблюдаемых вертикальных линий. В расширенном фильтре Кальмана вектор состояний  $\mathbf{x}[k]$

рассматривается как гауссовская переменная с ковариационной матрицей  $\mathbf{P}[k]$ , имеющей следующий вид:

$$\mathbf{P}[k] = \begin{bmatrix} rr & rl \\ lr & ll \end{bmatrix},$$

где  $rr$  – ковариационная матрица состояния робота;  $ll$  – ковариационная матрица координат совокупности наблюдаемых вертикальных линий;  $rl$ ,  $lr$  – ковариационная матрица состояния робота и координат вертикальных линий, при этом  $\mathbf{P}_{rl} = \mathbf{P}_{lr}^T$ .

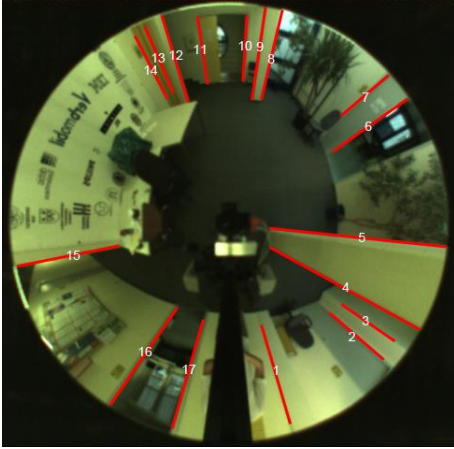


Рис. 1. Результат выделения и инициализации вертикальных линий на полусферическом изображении

Структурная схема алгоритма локализации робота на основе расширенного фильтра Калмана представлена на рис. 2.



Рис. 2. Структурная схема предложенного алгоритма.

В шаге  $k$  сначала проводится экстраполяция, при которой получается априорная оценка вектора состояний системы  $\mathbf{x}^-[k]$  по его апостериорной оценке с предыдущего шага на

основе динамической модели системы. Процесс экстраполяции описывается динамической моделью изменения вектора состояний:

$$\mathbf{x}^-[k] = \begin{bmatrix} x_r^-[k] \\ y_r^-[k] \\ \phi_r^-[k] \\ x_{L_1}^-[k] \\ \vdots \\ x_{L_n}^-[k] \end{bmatrix} = \begin{bmatrix} f(x_r^+[k-1], \mathbf{u}[k], \mathbf{w}[k]) \\ x_{L_1}^+[k-1] \\ \vdots \\ x_{L_n}^+[k-1] \end{bmatrix}$$

$$= \begin{bmatrix} x_r^+[k-1] + \Delta t[k]V[k]\cos(\phi_r^+[k-1] + \gamma[k]) \\ y_r^+[k-1] + \Delta t[k]V[k]\sin(\phi_r^+[k-1] + \gamma[k]) \\ \phi_r^+[k-1] + \Delta t[k]V[k]\frac{\sin(\gamma[k])}{L} \\ x_{L_1}^+[k-1] \\ \vdots \\ x_{L_n}^+[k-1] \end{bmatrix} + \begin{bmatrix} w_x[k] \\ w_y[k] \\ w_\phi[k] \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

где  $\mathbf{u}[k] = [V[k] \ \gamma[k]]^T$  – вектор управляющих воздействий, который состоит из скорости движения и угла поворота робота;  $\mathbf{w}[k] = [w_x[k] \ w_y[k] \ w_\phi[k]]^T$  – вектор шумов состояния робота;  $L$  – длина робота.

Ковариационная матрица экстраполированного вектора состояний определяется:

$$\mathbf{P}^-[k] = \begin{bmatrix} \bar{rr}[k] & \bar{rl}[k] \\ \bar{lr}[k] & \bar{ll}[k] \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{F}_R \bar{rr}^+[k-1]\mathbf{F}_R^T + \mathbf{F}_U \mathbf{U}[k]\mathbf{F}_U^T + \mathbf{Q}[k] & \mathbf{F}_R \bar{rl}^+[k-1] \\ \bar{lr}^+[k-1]\mathbf{F}_R^T & \bar{ll}^+[k-1] \end{bmatrix},$$

где  $\mathbf{U}[k] = \begin{bmatrix} \sigma_V^2 & 0 \\ 0 & \sigma_\gamma^2 \end{bmatrix}$ ,  $\mathbf{Q}[k] = \begin{bmatrix} \sigma_{w_x}^2 & 0 & 0 \\ 0 & \sigma_{w_y}^2 & 0 \\ 0 & 0 & \sigma_{w_\phi}^2 \end{bmatrix}$  –

ковариационная матрица шумов управляющих воздействий и состояния робота;  $\mathbf{F}_R = \nabla_{x_R} \mathbf{f}$ ,  $\mathbf{F}_U = \nabla_{\mathbf{u}} \mathbf{f}$  – матрицы Якоби векторной функции  $\mathbf{f}(\mathbf{x}_R, \mathbf{u}, \mathbf{w})$  относительно состояния робота и вектора управляющих воздействий.

В шаге  $k$  полусферическая камера наблюдает набор вертикальных линий, координаты некоторых из них были оценены в предыдущих шагах. Измерения азимутальных углов вертикальных линий, которые повторно наблюдаются, используются для вычисления ошибки от измерений, ожидаемых при произведенной экстраполяции вектора состояний системы:

$$\mathbf{v}[k] = \mathbf{z}[k] - \mathbf{z}^-[k] = \mathbf{z}[k] - \mathbf{h}(\mathbf{x}^-[k], \mathbf{v}[k]);$$

$$\mathbf{h}(\mathbf{x}^-[k], \mathbf{v}[k]) = \arctg\left(\frac{y_r^-[k] - y_L[k]}{x_r^-[k] - x_L[k]}\right) - \phi_r^-[k] + \mathbf{v}[k],$$

где  $\mathbf{h}(\mathbf{x}[k], \mathbf{v}[k])$  – векторная функция наблюдения; и  $\mathbf{v}[k]$  – вектор гауссовских шумов измерений.

Аналогично, ковариационная матрица вектора ошибки вычисляется с помощью матрицы Якоби функции наблюдения в  $\mathbf{x}^-[k]$ :

$$\mathbf{S}[k] = \mathbf{H}_R \bar{rr}[k]\mathbf{H}_R^T + \mathbf{H}_R \bar{rl}[k]\mathbf{H}_L^T + \mathbf{H}_L \bar{lr}[k]\mathbf{H}_R^T + \mathbf{H}_L \bar{ll}[k]\mathbf{H}_L^T + \mathbf{R}[k],$$

где  $\mathbf{H}_R = \nabla_{x_R} \mathbf{h}$ ,  $\mathbf{H}_L = \nabla_{x_L} \mathbf{h}$  – матрицы Якоби векторной функции  $\mathbf{h}(\mathbf{x}_v, \mathbf{v})$  относительно вектора состояния робота и координат вертикальных линий;  $\mathbf{R}[k]$  – ковариационная матрица шумов наблюдений.

В случае, что инициализированные вертикальные линии в предыдущих шагах наблюдаются в текущем шаге, априорную оценку вектора состояний системы могут корректировать на основании ковариационной матрицы имеющейся

экстраполяции вектора состояний и полученных измерений. Коррекция ранее полученной априорной оценки вектора состояний с помощью текущими измерениями является апостериорной оценкой вектора состояний системы:

$$\mathbf{x}^+[k] = \mathbf{x}^-[k] + \mathbf{K}[k]\mathbf{v}[k].$$

Матрица коэффициентов усиления Калмана  $\mathbf{K}[k]$  определяется по формуле:

$$\mathbf{K}[k] = -[k](\nabla_x \mathbf{h}[k])^T (\mathbf{S}[k])^{-1}.$$

Наконец, ковариационная матрица апостериорной оценки вектора состояний системы определяется:

$$\mathbf{S}^+[k] = \mathbf{S}^-[k] - \mathbf{K}[k]\mathbf{S}[k](\mathbf{K}[k])^T.$$

### 3. АЛГОРИТМ ИНИЦИАЛИЗАЦИИ ВЕРТИКАЛЬНЫХ ЛИНИЙ

Под инициализацией вертикальных линий подразумевается оценка их координат; оценка ковариационной матрицы их координат и ковариационной матрицы их координат с компонентами текущего вектора состояний системы, чтобы далее прибавить их к текущему вектору состояний и ковариационной матрице системы.

Известно, что для инициализации вертикальной линии необходимо её наблюдение с различных позиций робота. Такая модель, как было показано выше, характеризуется высокой степенью нелинейности, особенно при малом угле параллакса [5]. Поэтому при прохождении через эту модель измерения гауссовская форма плотности распределения вероятностей координат вертикальных линий сильно искажена и неточно оценивается с помощью только двух моментов (мат. ожидание и ковариационная матрица) по методу расширенного фильтра Калмана. Для преодоления данной проблемы предлагается алгоритм инициализации вертикальных линий, который улучшает оценку плотности распределения вероятности координат вертикальных линий.

#### 3.1 Оценка координат вертикальных линий

Пусть робот наблюдает вертикальные линии в  $n$  разных позициях, которые выражаются проекционными матрицами  $\mathbf{M}_i$  ( $i = 1 \dots n$ ) относительно первой позиции наблюдения вертикальной линии. Матрица  $\mathbf{M}_i$  определяет пространственное преобразование (поворот, трансляция) между позициями робота относительно первой позиции:

$$\mathbf{M}_1 = [\mathbf{I} | \mathbf{0}];$$

$$\mathbf{M}_i = [\mathbf{R}_i | \mathbf{t}_i] \quad (i = 2 \dots n),$$

где  $\mathbf{R}_i = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix}$  - матрица поворота;  $\mathbf{t}_i = [t_{ix} \quad t_{iy}]^T$  - вектор трансляции;  $\theta_i, t_{ix}, t_{iy}$  являются углом поворота, трансляцией по  $Ox, Oy$  робота в позиции  $i$  относительно первой и определяются апостериорными оценками соответствующих шагов.

Пусть координаты вертикальной линии относительно первой позиции робота выражаются в двухмерном проективном пространстве в виде  $\mathbf{x}_L = [x_L \quad y_L \quad 1]^T$ . Изображение вертикальной линии, формируемое полусферической камерой, является радиальной линией, которая характеризуется измерением азимутального угла  $z = \theta$ . Поэтому координаты изображения вертикальной линии могут быть выражены по формуле  $\mathbf{u} = [\cos\theta \quad \sin\theta]^T = [u_1 \quad u_2]^T$ .

За счет того, что азимутальные углы вертикальной линии и её изображения одинаковы, соотношение между их координатами в одной системе координат определяется скалярным параметром  $\lambda$ :

$$\lambda_i \mathbf{u}_i = \mathbf{M}_i \mathbf{x}_L.$$

Координаты вертикальной линии  $\mathbf{x}_L$  определяются пересечением прямых  $\lambda_i \mathbf{u}_i$ . Система уравнений в матричном виде имеет вид:

$$\begin{bmatrix} \mathbf{M}_1 & \mathbf{u}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{M}_2 & \mathbf{0} & \mathbf{u}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{M}_n & \mathbf{0} & \mathbf{0} & \dots & \mathbf{u}_n \end{bmatrix} [\mathbf{x}_L \quad -\lambda_1 \quad -\lambda_2 \quad \dots \quad -\lambda_n]^T = \mathbf{0}.$$

За счет наличия шумов оценок состояния робота и измерений азимутальных углов, прямые  $\lambda_i \mathbf{u}_i$ , в общем случае, не пересекаются в одной точке. Для решения данной системы уравнений предлагается применить метод сингулярного разложения, который обеспечивает решение с минимальной ошибкой соответствующее наименьшему сингулярному числу матрицы коэффициентов системы.

#### 3.2 Оценки плотности распределения вероятностей вертикальных линий

Оценка плотности распределения вероятностей координат вертикальных линий основана на методе Unscented Transform [4]. Поскольку не существует единого отечественного терминологического аналога, мы предлагаем, что использовано оригинальное авторское название Unscented Transform (UT). Сущность метода заключается в том, что ансамбль  $\mathbf{x}$  содержащий  $2n + 1$  взвешенных точек, называемых сигма-точками [4], используется для описания плотности распределения вероятностей случайной величины; вместо математического ожидания и ковариационной матрицы сигма-точки преобразуются нелинейной системой и в результате математическое описание и ковариационная матрица преобразованных сигма-точек описывает аппроксимацию искомой плотности распределения вероятностей.

Для применения метода UT необходимо задать вектор оценки  $\boldsymbol{\mu}$ , который состоит из состояний робота  $\mathbf{x}_{r_i}$  в  $n$  позициях и измерений азимутальных углов  $\theta_i$ :

$$\boldsymbol{\mu} = [\mathbf{x}_{r_1}^T \quad \dots \quad \mathbf{x}_{r_n}^T \quad \theta_1 \quad \dots \quad \theta_n].$$

Ковариационная матрица вектора оценки  $\boldsymbol{\mu}$  является диагональной матрицей, ненулевые элементы которой соответствуют дисперсии состояний робота и измерений. Взвешенные коэффициенты сигма-точек ансамбля  $\mathbf{x}$  определяются следующим образом:

$$\boldsymbol{\chi}_0 = \boldsymbol{\mu};$$

$$\boldsymbol{\chi}_i = \boldsymbol{\mu} + (\sqrt{(n + \rho)\mathbf{P}})_i \quad \text{для } i = 1 \dots n;$$

$$\boldsymbol{\chi}_i = \boldsymbol{\mu} - (\sqrt{(n + \rho)\mathbf{P}})_{i-n} \quad \text{для } i = n + 1 \dots 2n;$$

$$W_0^{(m)} = \frac{\rho}{n + \rho};$$

$$W_0^{(c)} = \frac{\rho}{n + \rho} + (1 - \alpha^2 + \beta);$$

$$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(n + \rho)} \quad \text{для } i = 1 \dots 2n,$$

где  $\rho = \alpha^2(n + \kappa) - n$  представляет собой коэффициент масштаба, при этом по [4]  $\alpha = 10^{-3}$ ,  $\kappa$  часто равняется 0;  $\beta = 2$ ;  $(\sqrt{(n + \rho)\mathbf{P}})_i$  обозначает  $i$ -ую строку матрицы  $\sqrt{(n + \rho)\mathbf{P}}$ .

Далее проводится вычисление оценки координат вертикальных линий  $\mathbf{x}_{L_i}$  на основе соответствующего набора сигма-точек  $\boldsymbol{\chi}_i$  по алгоритму показанному в п.3.1. Среднее

значение  $\bar{x}_L$  и ковариационная матрица  $P_l$  координат вертикальных линий определяются по формуле:

$$\bar{x}_L = \sum_{i=0}^{2n} W_i^{(m)} x_{L_i};$$

$$P_l = \sum_{i=0}^{2n} W_i^{(c)} (x_{L_i} - \bar{x}_L)(x_{L_i} - \bar{x}_L)^T.$$

$\bar{x}_L$  является оценкой координат вертикальной линии относительно первой позиции робота, поэтому для нахождения её координат в глобальной системе координат необходимо провести геометрическое преобразование  $g(\bar{x}_L, x_{R_1}, \theta_1)$ , где  $x_{R_1}$  – вектор состояния робота на первой позиции:

$$\bar{x}_{L_n} = \begin{bmatrix} x_L \\ y_L \end{bmatrix} = \begin{bmatrix} x_{r_1} + \bar{x}_L \cos(\phi_{r_1} + \theta_1) \\ y_{r_1} + \bar{y}_L \sin(\phi_{r_1} + \theta_1) \end{bmatrix}.$$

Ковариационная матрица новой вертикальной линии определяется с помощью матриц Якоби  $G_{x_L} = \nabla_{\bar{x}_L} g$ ,  $G_\theta = \nabla_{\theta_1} g$  и  $G_{x_R} = \nabla_{x_{R_1}} g$ :

$$P_n = G_{x_L} P_l G_{x_L}^T + G_{x_R} P_{r_1} G_{x_R}^T + G_\theta P_{\theta_1} G_\theta^T.$$

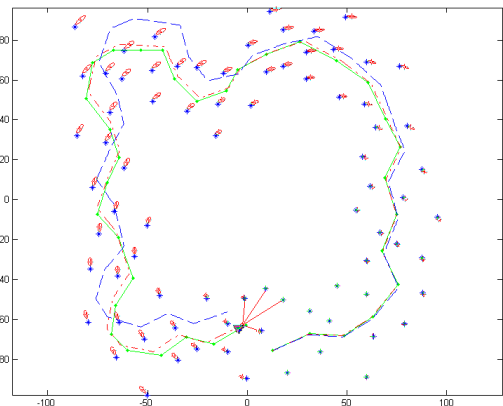
После инициализации математическое ожидание и ковариационная матрица вертикальных линий добавляются к вектору состояний и ковариационной матрице системы, соответственно. Ковариационные матрицы для существующих компонентов вектора состояний системы и новой вертикальной линии добавляются следующим образом:

$$x^+[k] = [x_r^+[k] \quad x_{l_1}^+[k] \quad \dots \quad x_{l_n}^+[k] \quad \bar{x}_{L_n}]^T,$$

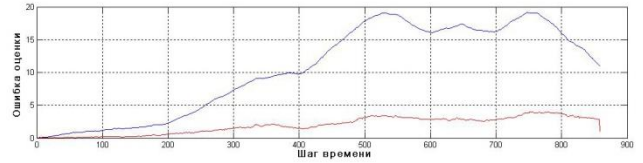
$$P^+[k] = \begin{bmatrix} P_{rr}^+ & P_{lr}^+ & P_{ll}^+ \\ P_{lr}^+ & P_{ll}^+ & P_{ln}^+ \\ P_{ln}^+ & P_{ll}^+ & P_{ln}^+ \end{bmatrix} = \begin{bmatrix} P_{rr}^+ & P_{rl}^+ & P_{rr}^+ G_{x_R}^T \\ P_{lr}^+ & P_{ll}^+ & P_{lr}^+ G_{x_R}^T \\ G_{x_R} P_{rr}^+ & G_{x_R} P_{rl}^+ & P_{ln}^+ \end{bmatrix}.$$

#### 4. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ

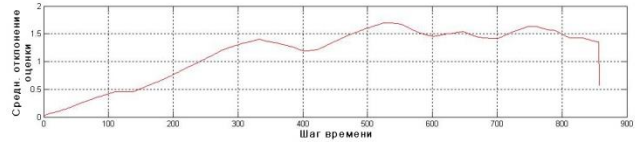
Для апробации предлагаемого алгоритма было разработано программное обеспечение для пакета MatLab моделирующее исследуемую среду размера 80м × 80м с заданными вертикальными линиями и траекторией движения робота. Робот движется по траектории с линейной скоростью 3 м/с и угловой скоростью 3 м/с. Среднеквадратические отклонения шумов одометра, установленного на робот составляют  $\sigma_V = 0.3$  м/с,  $\sigma_\gamma = 3$  градуса; и шума измерений полусферической камеры -  $\sigma_\theta = 1$  градус.



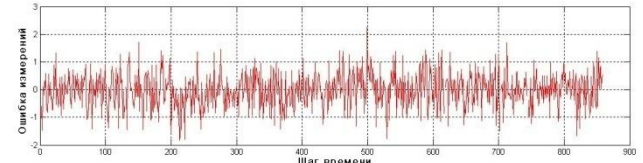
а)



б)



в)



г)

**Рис. 3.** а) Результаты моделирование алгоритма локализации (зеленная траектория – заданная; красная, штриховая – оцененная предложенным алгоритмом траектория; синий, штрихпунктирная – траектория одометра; синие звезды - заданные вертикальные линии; красные плюсы – оцененные; красные эллипсы – ковариационная матрица координат вертикальных линий); б) ошибка оценки координат робота (красная – алгоритма, синяя - одометра); в) среднеквадратическое отклонение оценки координат робота; г) ошибка измерений

На рис. 3а показан результат моделирования предложенного алгоритма локализации. Очевидно, что при повышении накапливаемой погрешности одометра, адаптируя коэффициенты усиления Калмана для ошибки измерений, алгоритм корректирует погрешность одометра. Рис. 3б, 3в показаны ошибки и среднеквадратическое отклонение оценки координат робота для предложенного алгоритма и одометра. Очевидно, что неопределенность оценки координат робота стабилизируется после установления и сильно снижается при возвращении робота в начальное место. Это объясняется тем, что в момент возвращения робота в начальное место соотношение между вертикальными линиями станет тесным, и коррекция для первых наблюдаемых вертикальных линий приводит к коррекции и других существенных в карте вертикальных линий. Рис. 3г показана ошибка измерений ожидаемых при произведенной экстраполяции вектора состояний системы. Небольшая ошибка измерений показывает, что априорная оценка хорошо корректируется так, что дает ожидаемое измерение близко к реальному измерению.

#### 5. ЗАКЛЮЧЕНИЕ

Предложенный в данной статье алгоритм локализации позволяет корректировать накапливаемую погрешность одометра и обеспечивает стабильную неопределенность оценки координат робота.

б)

## 6. ССЫЛКИ

- [1] T. Bailey. Constrained Initialisation for Bearing-only SLAM. In *Proceedings of IEEE International Conference on Robotics and Automation*, 2003.
- [2] A.J. Davison. Real-time Simultaneous Localisation and Mapping with a Single Camera. In *Proceedings of IEEE Conference on Computer Vision*, 2003.
- [3] R.E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME*, 1960.
- [4] J.J. Simon, J.K. Uhlmann. New extension of the Kalman filter to nonlinear systems. In *Proc. of AeroSense: The 11th Int. Symp. on Aerospace/Defence Sensing, Simulation and Controls*, 1997.
- [5] J. Sola, A. Monin, M. Devy, T. Lemaire. Undelayed Initialisation in Bearing-Only SLAM. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

### Об авторах

Александр В. Овчинников – к.т.н., доцент Тульского государственного университета. E-mail: [admin\\_telex@mail.ru](mailto:admin_telex@mail.ru).  
Хоа Д. Фан – аспирант Тульского государственного университета. E-mail: [dolphin22a@yahoo.com](mailto:dolphin22a@yahoo.com).

# Современные направления развития визуализации данных

А.Е.Бондарев, В.А. Галактионов  
Институт прикладной математики им. М.В.Келдыша РАН  
Москва, Россия  
[bond@keldysh.ru](mailto:bond@keldysh.ru); [vlgal@gin.keldysh.ru](mailto:vlgal@gin.keldysh.ru)

## Аннотация

Данный доклад представляет собой обзор современных направлений развития визуализации данных. Отмечены факторы, определяющие общие тенденции развития анализа численных данных. Рассматриваются примеры синтеза методов и подходов, применяемых в различных разделах анализа и обработки данных. Обсуждаются перспективы применения подобного синтеза к результатам математического моделирования задач вычислительной физики.

**Ключевые слова:** научная визуализация, задачи вычислительной физики, математическое моделирование

## 1. ВВЕДЕНИЕ

Данная работа представляет собой авторский взгляд на основные современные тенденции развития методов, алгоритмов и подходов визуального представления численных данных. Под численными данными подразумеваются результаты математического моделирования задач вычислительной физики, а точнее, вычислительной механики жидкости и газа. В данной обзорной работе использованы материалы трудов международной конференции Графикон за последние годы, а также работы, опубликованные в журнале «Научная визуализация».

Визуальное представление численных данных, являющихся результатами научных исследований или прикладных, инженерных и технологических расчетов, на сегодняшний день представляет собой важнейшую отрасль знания и технологического развития. Без использования современных концепций, методов и программных средств научной визуализации невозможно провести полный и достоверный анализ результатов численного моделирования сложных физических и технологических процессов, проводимый на современной вычислительной технике.

Алгоритмы визуального представления численных результатов развивались сообразно необходимости отображать решения прямых задач математического моделирования – преимущественно двумерных и трехмерных нестационарных процессов. По мере развития алгоритмов численного решения подобных задач развивались алгоритмы визуализации. Эволюция основных концепций, методов и подходов научной визуализации для задач вычислительной механики жидкости и газа и основные стадии исторического развития подробно рассмотрены в обзорной работе [1].

Для прямых задач математического моделирования общая схема проведения вычислительного эксперимента выглядит традиционно. Выбирается математическая модель и

численный метод ее реализации, затем проводится численный расчет. Визуальное представление результатов расчета обеспечивает пользователю возможность анализа и адекватной трактовки результатов. Одновременно с этим визуализация позволяет верифицировать полученные результаты и при необходимости вносить коррективы в математическую модель и метод расчета. Подобная схема вычислительного эксперимента складывалась на протяжении долгого времени. Однако в последние годы стремительное развитие технологий, алгоритмов и математических методов выдвигает новые задачи, вызванные появлением новых вычислительных возможностей.

## 2. ОСНОВНЫЕ ФАКТОРЫ ВЛИЯНИЯ

Можно выделить несколько главных факторов, определяющих современные направления развития математического моделирования и связанные с ними новые задачи визуального представления результатов.

Основополагающим фактором является интенсивное развитие высокопроизводительных вычислений и параллельных алгоритмов. Параллельные вычисления позволяют не только увеличить скорость расчетов и обеспечить применение подробных сеток. Они также предоставляют возможность проведения параметрических исследований и решения мультидисциплинарных и обратных задач. Эти задачи постепенно выходят на первый план ввиду их перспективности. Мультидисциплинарные задачи предполагают проведение численного эксперимента с использованием различных моделей. Решение обратных задач имеет своей целью поиск причин, вызывающих изучаемое физическое явление. Параметрические численные исследования позволяют получать решение не для одной конкретной задачи математического моделирования, а для класса задач, заданного в многомерном пространстве определяющих параметров. Также применение параллельных алгоритмов на высокопроизводительной вычислительной технике предоставляет возможность численного исследования задач оптимизационного анализа, когда обратная задача решается в каждой точке сеточного разбиения многомерного пространства определяющих параметров.

Основная особенность с точки зрения задач визуализации в подобных вычислениях заключается в том, что их результаты представляют собой многомерные массивы, размерность которых соответствует количеству определяющих параметров. Эти массивы нуждаются в обработке и визуальном представлении для анализа данных. Для этого необходима интеграция алгоритмов визуального представления с методами Data Analysis в целях понижения

размерности [2], так как устоявшихся концепций визуализации многомерных данных на сегодняшний день нет. Также необходимо учесть, что большинство вышеперечисленных задач рассматриваются в оптимизационной постановке. Это приводит к необходимости организации визуального отображения оптимизационного процесса в вычислениях.

Другим не менее важным фактором в последнее десятилетие явилась произошедшая революция в экспериментальной визуализации физических процессов [3], которая заключалась в переходе от аналоговых механических и электронных технологий к цифровым технологиям в методах регистрации изображений полей течений и представления результатов экспериментальных исследований. Согласно [3], это привело к стремительному сближению визуализации результатов численного моделирования с визуализацией результатов экспериментов. В связи с внедрением цифровых технологий в методы регистрации и анализа потоков стало возможным проведение прямого сравнения результатов расчетов и результатов экспериментальной визуализации полей течений. Это обстоятельство открывает новые возможности применения методов обработки и анализа изображений для выделения скрытых структур и их трассировки.

Третьим важным фактором является интенсивно развивающийся в последние годы синтез подходов, алгоритмов и математических методов, разработанных в различных областях знания. Подобный синтез обеспечивает получение новых результатов. Так, например, для проявления скрытых структур в результатах газодинамических расчетов могут применяться методы анализа и обработки изображений [7,8]. В задачах идентификации и управления течениями с использованием сопряженных уравнений визуальное представление сопряженных параметров и управляющего функционала позволяет реализовать контроль процесса [10]. Или наоборот, использование сопряженных параметров применяется для создания полностью управляемого изображения [9]. Также в целях выявления скрытых взаимосвязей и классификации полученных данных широко применяются алгоритмы кластерного и факторного анализа, метод главных компонент и его модификации для существенно нелинейных случаев. Подобное взаимообогащение методов и подходов является весьма характерным и несет в себе огромный потенциал.

### 3. СОВРЕМЕННЫЕ ТЕНДЕНЦИИ

В данном разделе приведены некоторые новые направления исследований и классы задач, требующих решения, возникающие вследствие влияния вышеперечисленных факторов. Также приводятся отдельные примеры исследований в этих направлениях.

Проблемы решения задач оптимизационного анализа и визуализации результатов, получаемых в виде многомерного объема данных, рассмотрены в работах [2,4]. В этих работах представлен приближенный подход, предназначенный для исследования процессов образования пространственно-временных структур в нестационарных задачах вычислительной механики жидкости и газа с помощью параллельных вычислений. Применение подхода позволяет проводить быструю приближенную оценку зависимости

возникновения нестационарных структур в потоке от определяющих параметров задачи, выраженную в виде квазианалитических соотношений. Приводится пример решения конкретной задачи исследования образования пространственно-временных структур при взаимодействии нестационарных потоков. Полученные результаты представляют собой решение для класса задач, заданного в многомерном объеме определяющих параметров.

Решение подобных задач делает крайне важным смежное направление – реализацию поиска, выделения, визуального представления и трассировки образующихся в течениях нестационарных структур, а именно, вихревых зон и разрывов. Для непосредственного решения поиска вихрей необходимо использование математического аппарата, разработанного в математической физике для этих целей. Подробный обзор методов и подходов, применяемых для исследования процессов анализа и визуализации вихревых течений, приведен в работе [5]. Представленные методы позволяют эффективно проследивать критические точки течения, контролировать нестационарные процессы зарождения и распада вихрей в потоке. Конкретный пример применения данных методов для решения практических задач локализации и визуального представления вихревых структур в нестационарных трехмерных течениях представлен в работе [6].

Не менее важной задачей для общего анализа течений является обнаружение, визуализация и трассировка разрывов. Это направление исследований является ярким примером синтеза методов и подходов, разработанных в различных прикладных областях. Локализация структур в результатах газодинамических расчетов, полученных методами сквозного счета без выделения разрывов, является чрезвычайно актуальным и интенсивно развивающимся направлением. Впервые способ выделения структур в решении, основанный на применении алгоритмов обработки изображений к полю данных, полученных в процессе решения задач математической физики, был предложен в работе [7]. В данном подходе предлагалось рассматривать газодинамические функции как интенсивность изображения, а значения функций в каждой точке как элементы изображения (пиксели). Суть основной проблемы заключается в необходимости обнаружения тонких структур течения, которые нельзя обнаружить другими средствами, в том числе и прямой визуализацией. Необходимо применять методы обработки изображений, чтобы «проявить» визуально эти структуры на обычном изображении данных. Данный подход получил развитие, представленное в работе [8], где реализовано обобщение подхода на трехмерный случай с использованием методов вейвлет анализа. Также в [8] рассмотрено применение разработанного алгоритма к вязким моделям и построению адаптивных сеток на основании обнаруженных скрытых разрывов. Данная работа чрезвычайно расширяет область применения основного подхода [7].

Полученные с помощью вышеуказанных подходов нестационарные структуры течения нуждаются в верификации путем сравнения с результатами физических экспериментов. Современный уровень методов экспериментальной визуализации представлен в обзорной работе [3], где рассматривается анализ возможностей, предоставляемых при использовании сравнения



экспериментальных изображений потоков, полученных различными оптическими методами, с численной визуализацией газодинамических течений. В работе [3] приведены характеристики основных современных методов экспериментальной визуализации потоков: оптические методы, базирующиеся на рефракции, рассеянии, поглощении, переизлучении, поляризации излучения; методы поверхностной визуализации (PSP – Pressure Sensitive Paints); трассирование (включая PIV - Particle Image Velocimetry модификации). Показана возможность проведения прямого сравнения экспериментальных и численных результатов. Также в [3] приведен пример анализа течений с импульсным энергокладом в поток на основе сравнения теневых, ТФМ (теневой фоновый метод) и расчетных изображений визуализированных ударно-волновых конфигураций.

Отдельным важным направлением являются задачи, основанные на синтезе математического аппарата теории сопряженных операторов и уравнений и методов визуального представления. Применение сопряженных параметров вкупе с решением оптимизационных задач позволяет создавать новые эффективные алгоритмы визуального представления. Подход, представленный в работе [9], позволяет создавать эффективные, полностью управляемые и физически точные (основанные на решении уравнений Навье-Стокса) визуальные представления распространения дыма и свободных поверхностей жидкости. Использование сопряженных уравнений позволяет решать широкий спектр обратных задач вычислительной механики жидкости и газа, в том числе задачи идентификации течения и управления течением. Непосредственная визуализация градиента целевого функционала и связанных с ним полей сопряженных параметров, подобно представленной в [10], позволяет получать новую информацию, являющуюся ключевой для рассматриваемой конкретной задачи. В задачах управления течением, идентификации течения (определения начальных или граничных условий, коэффициентов) поле управляющих параметров и градиент целевого функционала используются для поиска зон наиболее эффективного контроля или для выбора зон измерения. Примеры реализации такого подхода представлены в работе [10], где особо отмечается перспективность используемых методов для практического решения актуальных задач верификации и валидации.

Важным и перспективным направлением является интеграция современных методов анализа данных (Data Analysis) и методов визуализации применительно к исследованию многомерных объемов данных, являющихся результатами численного моделирования задач вычислительной механики жидкости и газа. Применение методов и подходов анализа данных позволяет проводить в исследуемом многомерном объеме выявление основных факторов влияния и скрытых взаимосвязей между ними, понижать размерность объема, решать задачи классификации объектов внутри объема. Однако простое механическое применение методов анализа данных к результатам газодинамических вычислений является затруднительным в силу специфики задач. Перспективным направлением реализации такой интеграции может служить адаптация и реализация методов Data Analysis в рамках подхода функционального представления исследуемых объектов, представленного в работах [11,12]. Огромным преимуществом данного подхода является возможность изначального представления исследуемых данных в виде функций многих переменных и построения

системы операторов, позволяющих определять преобразования объектов и отношения между ними.

#### 4. ЗАКЛЮЧЕНИЕ

Представленные выше направления и примеры исследований совместно с четко наметившейся тенденцией слияния и синтеза методов, подходов и алгоритмов из различных областей дают основание полагать, что в будущем возможно создание совокупности алгоритмов, программных и технических средств, дающей возможность реализовать *обобщенный вычислительный эксперимент*. Подобный эксперимент, опирающийся на прямые задачи математического моделирования, реализующий алгоритмы поиска и трассировки нестационарных структур в течениях, использующий обратные и оптимизационные постановки задач и верифицируемый путем прямого сравнения с результатами экспериментов, позволит получать численные решения для классов задач. Полученные результаты позволят решать практические задачи контроля и управления нестационарными процессами в вычислительной механике жидкости и газа.

В выстраиваемой технологической цепочке предполагаемого эксперимента основополагающую роль для исследования, обработки, трактовки и верификации численных результатов будет играть визуализация данных. Используемая на всех стадиях вычислительного эксперимента, она предполагает реализацию по следующим направлениям: визуализация решений прямых задач, визуализация результатов поиска нестационарных структур в течениях, визуализация процесса оптимизации, визуальное представление скрытых взаимосвязей в многомерном объеме данных, обработка и прямое сравнение экспериментальных и численных результатов с целью верификации и совершенствования моделей расчета и эксперимента.

#### 5. БЛАГОДАРНОСТИ

Данная работа выполнена при поддержке РФФИ (проект № 13-01-0367А)

#### 6. ЛИТЕРАТУРА

- [1] Бондарев А.Е., Галактионов В.А., Четкин В.М. *Анализ развития концепций и методов визуального представления данных в задачах вычислительной физики / Журнал вычислительной математики и математической физики, 2011, Т. 51, N 4, С. 669–683.*
- [2] Бондарев А.Е., Галактионов В.А. *Анализ многомерных данных в задачах многопараметрической оптимизации с применением методов визуализации / Научная визуализация. Т.4, № 2, с.1-13, 2012, URL: <http://sv-journal.com/2012-2/01.php?lang=ru>*
- [3] Знаменская И.А. *Взаимодействие численной и экспериментальной визуализации потоков / Научная визуализация. Т.5, № 3, 2013.*
- [4] Bondarev A.E., Galaktionov V.A. *Parametric Optimizing Analysis of Unsteady Structures and Visualization of*

*Multidimensional Data // International Journal of Modeling, Simulation and Scientific Computing, Vol. 4, suppl. issue 1, 2013, DOI: 10.1142/S1793962313410043*  
<http://www.worldscientific.com/doi/abs/10.1142/S1793962313410043>

[5] Tricoche X., Garth C. *Topological Methods for Visualizing Vortical Flows // Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration, Ed. by Möller T., Hamann B., Russell R., Springer-Verlag, 2009, pp.89-108.*

[6] Илина Т.В., Лутский А.Е. *Numerical Simulation And Visualization Of Wing Vortices / Proc. of PSFVIP-8: The 8th Pacific Symposium on Flow Visualization and Image Processing, Moscow, , 2011*

[7] 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.*

[8] Афондииков А.Л., Луцкий А.Е., Плёнкин А.В. *Применение вейвлет анализа для выделения структур в расчетах газодинамических течений и для адаптации сеток / Научная визуализация. Т.4, № 3, с.8-25, 2012, URL: <http://sv-journal.com/2012-3/02.php?lang=ru>*

[9] McNamara A., Treuille A., Popovic Z. , Stam J. *Fluid control using the adjoint method / ACM Transactions on Graphics (SIGGRAPH 2004), 23(3), August 2004.*

[10] Алексеев А.К., Бондарев А.Е. *Применение сопряженных уравнений и визуальное представление сопряженных параметров в задачах идентификации и управления течением, верификации и валидации расчетов / Научная визуализация. Т.4, № 3, с.1-24, 2011, URL: <http://sv-journal.com/2011-3/01.php?lang=ru>*

[11] Pasko A. , Adzhiev V., Malikova E., Pilyugin V. *Scientific Visualization and FRep Geometric Modeling: a Survey / Proc. 22th International Conference on Computer Graphics & Vision GRAPHICON'2012, Moscow, 2012, p.189-192.*

[12] Пилюгин В.В., Маликова Е.Е., Пасько А.А., Аджиев В.Д. *Научная визуализация как метод анализа научных данных / Научная визуализация. Т.4, № 4, с.8-25, 2012, URL: <http://sv-journal.com/2012-4/06.php?lang=ru>*

## Об авторах

Бондарев Александр Евгеньевич – к.ф.-м.н., старший научный сотрудник ИПМ им. М.В. Келдыша РАН  
E-mail: [bond@keldysh.ru](mailto:bond@keldysh.ru)

Галактионов Владимир Александрович – д.ф.-м.н., профессор, заведующий отделом компьютерной графики ИПМ им. М.В. Келдыша РАН.  
E-mail: [vlgal@gin.keldysh.ru](mailto:vlgal@gin.keldysh.ru)

## STATE-OF-THE-ART IN DATA VISUALIZATION

### Abstract

The article presents a review of main modern trends for data visualization. The main factors of influence are marked for these trends. The examples of methods and approaches combination for different fields of research are considered. The perspectives of combined approaches application to numerical results in computational physics are discussed.

**Keywords:** *scientific visualization, computational physics, mathematical modeling*

# Опыт создания систем визуализации реального времени и их применение в тренажерных и обучающих системах

Б.С. Долговесов, Б.Б. Морозов

Институт Автоматики и Электрометрии СО РАН, Новосибирск, Россия

bsd@iae.nsk.su

## Аннотация

В докладе представлены системы виртуальной реальности, созданные в разные годы в Институте автоматики и электрометрии СО РАН для практического применения в тренажерах РГНИИЦПК им. Ю.А.Гагарина для подготовки космонавтов по программам ОС «Мир» и МКС. Рассмотрен принцип создания системы обучения и презентаций на основе технологии интегрированной виртуальной реальности.

**Ключевые слова:** системы виртуальной реальности, интерактивное виртуальное окружение, 3D визуализация.

## 1. ВВЕДЕНИЕ

В пилотируемой космонавтике одним из основных средств подготовки космонавтов являются космические тренажеры (КТ), в структуре которых важное место занимают системы имитации внешней визуальной обстановки на всех этапах полета (выведение на орбиту, стыковка корабля с орбитальным пилотируемым комплексом, задачи орбитального полета, а также возвращение экипажа на Землю).

С развитием компьютерной графики появилась возможность замены использовавшихся ранее для тренажеров имитаторов визуальной обстановки с физическими моделями и телевизионными камерами на имитаторы с математическим моделированием визуальной обстановки. Это так называемые системы виртуальной реальности (СВР), где визуальная обстановка формируется методами компьютерного синтеза трехмерных сцен. Современные компьютерные технологии обеспечивают широкие возможности моделирования в реальном масштабе времени динамических сюжетов визуальной обстановки и универсальность тренажерных СВР. Особенно это важно при моделировании сценариев развития внештатных ситуаций, отработка выхода из которых занимает большую часть времени от всей подготовки космонавта. Появилась возможность имитировать различные состояния атмосферы, погодные явления, световые эффекты, тени и т. д.

С начала 80-х годов XX века начинается активное сотрудничество Института автоматики и электрометрии СО РАН (ИАиЭ СО РАН) с Центром подготовки космонавтов им. Ю. А. Гагарина по созданию СВР для космических тренажеров. На базе разработок института по формированию и отображению виртуальных сцен в реальном времени создано несколько поколений СВР для практического применения (рис. 1) в имитационно-тренажерных комплексах РГНИИЦПК им. Ю.А.Гагарина и РКК "Энергия" им. С. П. Королева. Системы успешно эксплуатировались при подготовке космонавтов орбитальной станции «Мир», а затем - по программе международной космической станции (МКС). За создание систем «Аксай» и «Альбатрос» коллектив разработчиков отмечен дипломами им. Ю. А. Гагарина.



Рис 1: Виртуальная сцена сближения транспортного корабля с МКС.

## 2. СИСТЕМЫ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ ТРЕНАЖЕРНЫХ КОМПЛЕКСОВ

Для практического использования в тренажерах Центра подготовки космонавтов им. Ю.А.Гагарина в ИАиЭ разработан и создан образец трехканальной СВР «Аксай» (1981-1985 гг.) Это была первая в СССР профессиональная система для космического тренажера. Разработанная оригинальная структура синтеза трехмерных сцен на основе приоритетно-упорядоченных объектов обеспечивала визуализацию динамических сцен в реальном масштабе времени. Аппаратная часть системы, используя существующую в то время элементную базу, размещалась в девяти типовых стойках. Создан комплекс программных средств, поддерживающий систему от этапа создания до работы в составе тренажера ЦПК им. Ю. А. Гагарина. При разработке системы «Аксай» были учтены и рекомендации космонавтов, передавшие свой летный и космический «визуальный» опыт. Система «Аксай» успешно эксплуатировалась при подготовке космонавтов по программе космической станции «Мир».

Работы по совершенствованию СВР были продолжены. Разрабатываются новые структурные решения и алгоритмы формирования изображений, ориентированные на создание СВР с меньшими аппаратными затратами (по сравнению, например, с «Аксаем»), но с возможностью расширения по производительности и по набору функций в зависимости от решаемых задач. Эти разработки вызвали интерес в ЦПК им. Ю. А. Гагарина, где начинается процесс модернизации тренажерного парка. Создается ряд СВР «Альбатрос» (1986-1990 гг.), в которых использовались разработанные на отечественной элементной базе специализированные графические устройства (рис.2). В основу архитектуры видеопроцессора системы «Альбатрос» [1] положена оригинальная идея рекурсивной процедуры деления экрана. В системе реализован разработанный метод растрирования и

многоуровневого маскирования, увеличивший производительность системы и позволивший эффективно обрабатывать трехмерные сцены с большой глубиной сложностью. Многоканальные СВР на базе «Альбатроса» со специальными функциональными возможностями (текстурирование поверхностей, атмосферные эффекты, подвижные объекты, имитация атмосферных эффектов и различного рода источников света) использовались в авиакосмических тренажерных комплексах ЦПК им. Ю.А. Гагарина, ПКБМ г. Пензы. Архитектурные и алгоритмические решения, обеспечили модульность систем, существенно меньший объем оборудования при характеристиках, сравнимых с лучшими образцами подобных систем зарубежных фирм того времени. Однородность архитектуры упростила изготовление, настройку и тестирование системы, повысила показатели надежности. Система обрабатывала до 40000 полигонов в секунду. С появлением высокоинтегрированной элементной базы разрабатывается СВР нового поколения «Ариус» (рис.3), где в качестве основного вычислительного ядра используются программируемые цифровые сигнальные процессоры [2]. В основу разработки этой системы положены такие принципы, как открытость архитектуры, программируемость на всех уровнях вычислений, однотипность структурных модулей, позволяющая легко изменять конфигурацию системы. Это, наряду с высокой производительностью, обеспечивалось специально разработанным модулем (рис. 4) на базе процессора TMS320C80.

На базе данной разработки создается ряд СВР с различными модификациями для решения задач специализированных тренажеров подготовки космонавтов по программе космической станции «Мир», а затем и МКС (рис. 1). Функции этих систем значительно расширены. Это имитация различных средств наблюдения с характерными для них эффектами (дисторсионные искажения, расфокусировка и т. д.), моделирование различного времени суток и состояния атмосферы. Имитация приборов наблюдения (рис.5) – одна из функций СВР в тренажерах для подготовки экипажей к визуальной оценке различных стадий полета транспортного корабля (полет по траектории, сближение и стыковка с МКС и т. д.), а также для визуальной оценки поверхности Земли с МКС по соответствующим приборам наблюдения.

Системы «Ариус» явились базой для перехода на стандартные графические акселераторы, что определило дальнейшее развитие и совершенствование СВР с учетом новых задач тренажерных комплексов. За счет программируемости на всех уровнях обработки данных такие системы универсальны и легко адаптируются для решения различных тренажерных задач. Это очень важно в связи с растущими требованиями к подготовке космонавтов по мере совершенствования космических транспортных кораблей и расширения функциональности МКС. На сегодняшний день проведена модернизация СВР различных тренажеров ЦПК им. Ю. А. Гагарина. В качестве компьютерных генераторов изображений используются многопроцессорные акселераторы нового поколения, высокая производительность которых и программная реализации многих функций позволяет полностью удовлетворить современные требования к функциональности и качеству визуальной обстановки в современных тренажерах.



Рис 2: Общий вид системы «Альбатрос».



Рис 3: Общий вид трехканальной системы «Ариус»

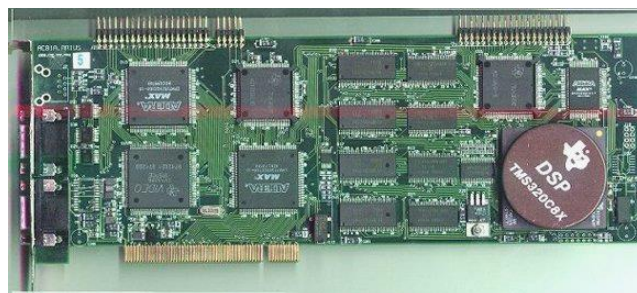


Рис 3: Внешний вид функционального модуля СВР «Ариус»

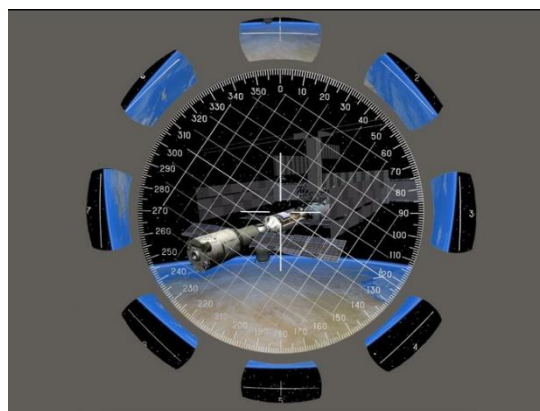


Рис 5: Компьютерная модель прибора ВСК-4

### 3. СИСТЕМЫ ОБУЧЕНИЯ И ПРЕЗЕНТАЦИЙ НА ОСНОВЕ ТЕХНОЛОГИИ ИНТЕГРИРОВАННОЙ ВИРТУАЛЬНОЙ СРЕДЫ

Одно из перспективных направлений в развитии СВР, получившее развитие в ИАиЭ СО РАН, – использование в обучающих и презентационных системах технологии интегрированной виртуальной реальности (ИВР). Данная технология предполагает не только пассивную демонстрацию компьютерных моделей пространственных объектов, явлений и процессов, но и активное «присутствие» лектора (преподавателя) в предметной виртуальной среде с непосредственным взаимодействием с моделями изучаемых объектов и акцентирующего внимание на проблемных аспектах изучаемого материала [3-5]. Исследованы методы управления трёхмерными объектами виртуальной сцены с помощью отслеживания движений человека без использования специальных сенсорных датчиков. Разработан метод распознавания движений, используя изображение с видеокамеры. В результате разработаны программно-алгоритмические средства, позволяющие из входных видеоданных выделять подвижные объекты, определять смещение этих объектов на экране и по общему смещению формировать сообщение с двумерным вектором скорости смещения. Система отслеживания движений обрабатывает это сообщение и использует для управления выбранным виртуальным объектом. За счет распознавания движений докладчика можно создана эффективная система управления презентацией без привлечения устройств типа пульта, джойстика или сенсорного экрана. Управлять презентацией становится возможным, используя как виртуальные кнопки, так и непосредственное управление 3D-объектами в виртуальной сцене. Выбор объекта, его перемещение, вращение, изменение масштаба осуществляются за счет анализа движений руки.

Структура многофункционального программно-аппаратного комплекса системы обучения и презентаций включает следующие программно-аппаратные компоненты: генератор изображений, систему отображения, интерактивные средства управления презентацией. Программное обеспечение системы включает следующие компоненты: менеджер ресурсов, модуль визуализации, менеджер сцены. Системы на основе ИВР могут быть использованы в учебных классах при обучении космонавтов и персонала оперативных групп. Изучение на виртуальных моделях компоновки орбитальных станций, отработка на виртуальных моделях процесса создания лабораторий в космосе, сборки сложных конструкций в космосе и т.д. – не полный перечень тематических примеров, которые наглядно могут быть продемонстрированы в интерактивном режиме. На рис. 6 приводится пример использования ИВР: совмещенное изображение виртуальной модели функционально-грузового блока «Заря» МКС и лектора, что обеспечивает наглядность интерактивной демонстрации.



Рис 6: Пример использования ИВР

### 4. ЗАКЛЮЧЕНИЕ

Дальнейшее развитие систем на основе ИВР предполагает создание ряда систем различной сложности для обучения и презентаций, в том числе и простой вариант на обычном портативном компьютере. Разработанные программно-алгоритмические решения позволяют использовать дополнительные подсистемы, расширяющие функциональные возможности и круг пользователей подобных систем. Это подсистема локализации и слежения (трекинга) за движениями рук пользователя; подсистема синтеза звуковых эффектов, чувствительной к положению и ориентации пользователя; подсистемой генерации силовых и тактильных ощущений, создающей иллюзию прикосновения к виртуальным объектам; устройства ввода для построения интерфейса пользователя прямого манипулирования данными (указки, панели управления, кинетические сенсоры и др.).

Сфера применения систем на основе ИВР: всевозможные презентации, в ситуационных центрах, музеях, обучение персонала управлению сложными технологическими процессами и техническими системами, дистанционное обучение и телеконференции.

### 5. ЛИТЕРАТУРА

- [1] Долговесов Б.С. Семейство компьютерных систем визуализации «Альбатрос» //Автометрия, 1994, N 6. С.3.
- [2] Вяткин С.И., Долговесов Б.С. и др. Архитектурные особенности системы визуализации реального времени на основе сигнальных процессоров // Автометрия ,1999, № 1. С. 110.
- [3] Dolgovesov B.S., Mazurok B.S и др. Some Aspects of Creating Presentation Systems based on the Technology of Integrated Virtual Reality // Proceedings of the IASTED International Conf. on Automation, Control, and Information Technology (ACIT-ICT-2010), June 15-18, 2010, Novosibirsk, Russia. 2010. vol.1. P. 229-232.
- [4] Долговесов Б. С. Лаврентьев М. М., Морозов Б. Б., Тарасовский А. Н. Система обучения и презентаций с использованием технологии «погружения» лектора в интерактивную виртуальную среду // Тр.междунар. конф. по вопросам обучения с применением технологий E-Learning «Online EDUCA 2007, Moscow», 30 сентября - 3 октября 2007, Москва.
- [5] Долговесов Б, Лаврентьев М. и др. Технология «погружения» лектора в интерактивную виртуальную среду // Высшее образование в России. – 2008. - №2. - С. 134-138.

## The experience of developing real-time rendering systems for the simulation and training

### Abstract

This paper is about implementation of the presentations system based on integrated virtual reality technology. This system provides the combining of a virtual environment and a lecturer who can interact with object's models of a virtual environment real-time. Furthermore the system can combine different multimedia data (3D objects' models, video, Microsoft PowerPoint presentations, histograms, tomography data, etc.). This option expands the potential of interactive presentation process.

**Keywords:** *virtual reality systems, interactive virtual environment, 3D render, keying.*

### About the author

Boris S. Dolgovesov (Ph.D.) is a head of Synthesizing Visualization Systems Laboratory at Institute of Automation and Electrometry SB RAS. His contact email is [bsd@iae.nsk.su](mailto:bsd@iae.nsk.su)

Boris B. Morozov is a scientific researcher of Software Systems for Computer Graphics Laboratory at Institute of Automation and Electrometry SB RAS. His contact email is [bbm@sl.iae.nsk.su](mailto:bbm@sl.iae.nsk.su)

# Восстановление 3D поверхности лица

Щеголева Надежда  
Санкт-Петербургский государственный электротехнический университет (ЛЭТИ)  
кафедра МО ЭВМ  
Санкт-Петербург, Россия  
stilhope2009@gmail.com

## Аннотация

Обсуждается способ улучшения качества реконструкции 3D поверхности лица или его частей с помощью метода обратного расстояния. Предлагаемый метод позволяет устранить «черные дыры», появляющиеся при отсутствии измерений. Применение метода позволяет построить 3D поверхности необходимого разрешения без решения задачи триангуляции, что обуславливает существенное сокращение вычислительных затрат.

**Ключевые слова:** 3D-реконструкция поверхности, range image, метод обратных расстояний, «черные дыры», распознавание лиц.

## 1. ВВЕДЕНИЕ

В последнее время 3D модели лиц находят все более широкое применение в системах распознавания изображений лиц. Однако, как отмечается в работах [1, 5, 6, 7], существует ряд проблем. Например, не точно подобранное освещение сканируемого лица, чрезмерно освещенные участки лица или области лица с очень низким освещением (и/или низким контрастом) могут негативно повлиять на реконструкцию 3D формы лица. Это приводит к тому, что на гладких поверхностях лица могут появляться артефакты в виде «выбросов» (шипов и наростов), а на острых формах и резких переходах (кончик носа, надбровные дуги, крылья носа и носогубные складки) могут образовываться провалы в 3D поверхности, называемые «черными дырами» [7].

## 2. ПОЛУЧЕНИЕ ИСХОДНЫХ 3D ФОРМ ЛИЦ

Шаг 1 – сканирование. Положение человека относительно камеры фиксируется в течение нескольких секунд лицо находится на довольно близком расстоянии от камеры и может подсвечиваться (см. рис. 1, а [7]).

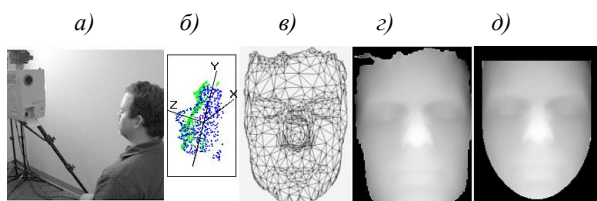


Рис. 1. Процесс получения 3D форм лиц

В результате сканирования области лица получается облако точек  $p_l$ , где  $l=1, 2, \dots, L$ , показанное на рис. 1, б. Значение  $L$  может варьироваться от нескольких сотен до нескольких десятков тысяч. Каждая точка  $p_l$  определяется в пространстве

3D своими координатами  $\{x_l, y_l, z_l\}$ , где  $x$  – координата перпендикулярная оси симметрии лица,  $y$  – параллельная оси симметрии лица, а  $z$  – определяет высоту точки относительно плоскости  $XY$ . Таким образом, облако точек описывается тремя векторами размера  $L$ .

Шаг 2 – связь точек в сетку треугольников. Поскольку переменные  $\{x_l, y_l\}$  представляют собой не связанные между собой координаты на плоскости  $XY$ , размещенные на нерегулярной сетке, то выполняется процедура триангуляции с линейной интерполяцией. Координаты  $z_l$ , определяющие вершины треугольников, создают опорные точки сетки, узлы которой определяют высоту рельефа области лица. Пример такой сетки показан на рис. 1, в [7].

Шаг 3 – перенос координат. Выполняется перенос координат  $\{x_l, y_l\}$  на регулярную сетку и пересчет координат  $z_l$ . Полученные значения проектируются ортографически на регулярную сетку. Вместе с полученными новыми вершинами  $z_l$  они образуют range image (RI) – рис. 1, г.

Шаг 4 – создание RI. RI выравнивается по всем направлениям (с помощью поворота, сдвига и т.д.) и переводится, например, в полутоновое изображение (рис. 1, д).

## 3. ВОССТАНОВЛЕНИЕ 3D ПОВЕРХНОСТИ НА ОСНОВЕ МЕТОДА ОБРАТНЫХ РАССТОЯНИЙ

Триангуляция и связанный с ней пересчет координат  $z_l$  дают очень хорошие результаты точного представления поверхностей лиц в пространстве 3D и в форме RI. Однако для этого необходимо очень большое число треугольников исходной сетки, что неминуемо приводит к высоким вычислительным затратам. Сокращение числа треугольников (и соответственно вычислительных затрат) достигается за счет «адаптивной триангуляции» - размещение большего числа треугольников там, где есть резкие перепады уровня на поверхности лица (области бровей, глаз, носа) и меньшего - в областях с плавными изменениями (лоб, скулы), что показано на рис. 1, в. Возможно построение сетки треугольников только по узлам, которые являются антропометрическими точками, тогда число исходных точек будет не более 80-100. Однако если исходное облако  $\{x_l, y_l, z_l\}$  имеет значительные пропуски «измерений по глубине» от сканера до области лица, то и сетка треугольников будет иметь большие «области пропусков». В результате на поверхности RI появятся «черные дыры», соответствующие этим пропускам информации, которые не позволят получить точной формы поверхности лица. Пример этих ошибок представлен на рис. 2, изображения б и в из [1].

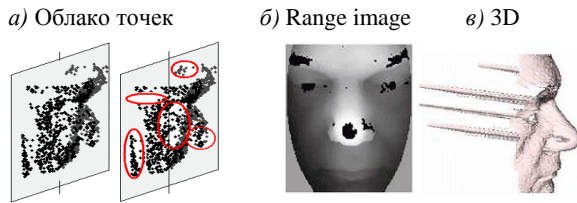


Рис. 2. Проблемы реконструкции RI

Образ, представленный на рис. 2 а), получен из значений  $\{z_l\}$  RI для координат  $x_l$  и  $y_l$  2000 точек, сформированных генератором случайных чисел. Тем самым совершен «обратный переход» от RI к облаку исходных данных. В построенном облаке точек отчетливо видны «рваные края» и значительные пропуски, которые и создают «дыры» на изображениях RI. Отсутствие данных о глубине в изображениях RI приводит к появлению «шипов» с обратной стороны 3D форм лиц. Необходимость устранения «черных дыр» в изображениях RI, которая выполняется, как правило, в интерактивном режиме с помощью сплайн-интерполяции, является одной из особенностей метода триангуляции. Ниже будет показано, что реконструкция поверхности лица по методу обратных расстояний (МОР) может успешно бороться с устранением «черных дыр» в изображениях RI.

Если необходима не сама модель лица, а решение задачи распознавания людей по лицам, представленным в форме 3D, то есть способы построения всей поверхности лица или его частей с последующим их использованием в процессе распознавания [6]. Можно также обойтись без объединения точек  $p_l$  в треугольники, не решая, таким образом, задачу триангуляции вообще. Вместо этого, можно по полученным измерениям  $\{x_l, y_l, z_l\}$ , принадлежащим нерегулярной сетке измерений, сразу построить поверхность 3D с требуемым разрешением (в приложении к RI - с требуемым размером  $M \times N$  этого изображения). Наиболее просто это реализуется в рамках метода «КРИКИНГ» (Kriging) в основе которого лежит метод обратных расстояний (МОР) [4, 9]. Решение задачи переноса данных с сетки нерегулярной на сетку регулярную реализуется в два этапа.

1. Исходной информацией для I этапа являются заданные значения  $\{x_l, y_l, z_l\}$ , представляющие векторы  $X, Y, Z$  размером  $L$  каждый. Целью этого этапа является построение модели  $z_l = f(x_l, y_l)$ . Для достижения этой цели выполняются следующие действия.

1.1. Координаты  $\{x_l, y_l\}$  всех исходных данных представляются вектором  $K$  в комплексной форме:

$$K = X + jY, \text{ где } j = \sqrt{-1}, \quad (1)$$

причем предполагается, что вектор  $K$  не содержит таких данных, что  $k_l \neq k_d$  для  $l \neq d$ . Это условие «некратности данных» является единственным при решении поставленной задачи.

1.2. Формируется матрица  $K$ , размером  $L \times L$ , посредством повторения  $L$  раз полученного вектора  $K$ :

$$K = [K \ K \ \dots \ K]. \quad (2)$$

1.3. Вычисляется дистанция между всеми точками нерегулярной сетки:

$$D = \text{abs}(K - K^T). \quad (3)$$

Заметим, что только при выполнении условия «некратности данных», ранг матрицы  $D$  будет равен ее порядку и возможен переход к следующему шагу алгоритма.

1. 4. Вычисляется вектор параметров регрессии  $B$ , размера  $L$

$$B = D^{-1}Z. \quad (4)$$

2. Целью II этапа является собственно перенос исходных данных (облака) на заданную регулярную сетку размера  $M \times N$ , в результате чего и будет получено RI. Вычисления каждого значения  $I(m, n)$  RI реализуются следующим образом:

$$I(m, n) = \sum_{l=1}^L b_l \cdot |k(n, m) - k_l|, \quad m=1, 2, \dots, M, \quad n=1, 2, \dots, N, \quad (5)$$

где:  $k(n, m) = x_n + jy_m, \quad \forall n=1, 2, \dots, N, \quad m=1, 2, \dots, M;$

$|k(n, m) - k_l|$  – модуль разницы между новыми и старыми координатами;  $x_n = \min(x) + n(\max(x) - \min(x))/(N - 1);$

$$y_m = \min(y) + m(\max(y) - \min(y))/(M - 1); \quad k_l = x_l + jy_l.$$

Решение (5) также достигается при условии «некратности данных»:  $|k(n, m) - k_l| \neq 0, \quad \forall l \neq n \text{ и } l \neq m.$  Алгоритм реализации (5) будет следующим.

2.1. Для векторов  $X$  и  $Y$ , рассчитываются границы координат:

$$\max\_Y = \max(Y); \quad \min\_Y = \min(Y);$$

$$\max\_X = \max(X); \quad \min\_X = \min(X).$$

2.2. Вычисляется шаг дискретизации регулярной сетки с учетом размеров  $M$  и  $N$  для RI:

$$\text{deltaY} = (\max\_Y - \min\_Y) / (M - 1);$$

$$\text{deltaX} = (\max\_X - \min\_X) / (N - 1).$$

2.3. Определяются все значения координат регулярной сетки для заданного размера RI и записываются в соответствующие векторы:

$$X_r = (\min\_X : \text{deltaX} : \max\_X);$$

$$Y_r = (\min\_Y : \text{deltaY} : \max\_Y).$$

2.4. Создается нулевой рабочий массив размера  $M \times N$  для RI:

$$I = [0]_{M \times N};$$

2.5. Для всех текущих значений координат  $m=1, 2, \dots, M$  и  $n=1, 2, \dots, N$  формируется вектор  $X_y$ , состоящий из  $L$  строк координат  $[X_r(n) + jY_r(m)]$  так, что:

$$X_y = \begin{bmatrix} X_r(n) + jY_r(m) \\ X_r(n) + jY_r(m) \\ \dots \\ X_r(n) + jY_r(m) \end{bmatrix}. \quad (6)$$

2.6. Вычисляются расстояния между новыми и исходными координатами

$$D_{new} = \text{abs}(X_y - K). \quad (7)$$

2.7. Вычисляются значения RI для пикселя  $(m, n)$ :

$$I(m, n) = D_{new}^T B, \quad (8)$$

далее осуществляется выбор нового текущего значения  $m$  и  $n$ , повторяются шаги (6)-(8). Результат приведен на рис. 3.



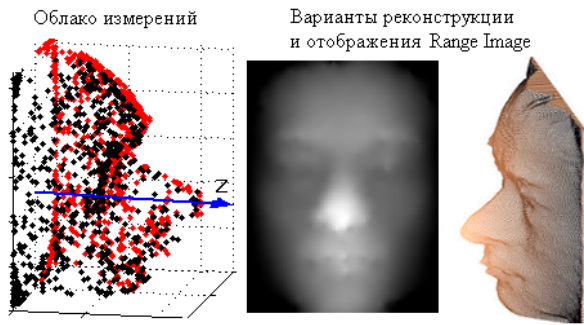


Рис. 3. Перенос данных по предложенному алгоритму

#### 4. ЭКСПЕРИМЕНТЫ

В экспериментах использовались изображения «Texas 3-D Face Recognition Database» [3], а также изображения из цитируемых статей [1, 5, 7, 8]. Все эти изображения показаны на рис. 4 и 5.

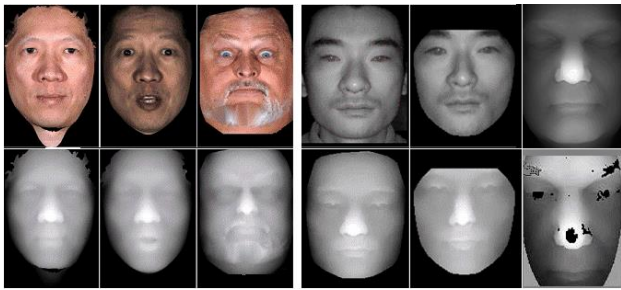


Рис. 4. Примеры изображений

##### 4.1 Реконструкция полного изображения range image

К сожалению, экспериментальные координаты  $\{x_i, y_i, z_i\}$ , создающие облако исходных данных, отсутствовали, поэтому во всех представленных ниже экспериментах эти данные будут создаваться из имеющихся оригиналов – RI.

Значения  $\{z_i\}$  облака исходных данных будем получать из RI для координат  $x_i$  и  $y_i$ , сформированных генератором равномерно распределенных случайных чисел. Координаты  $x_i$  и  $y_i$  в этом случае не являются целыми и, таким образом, принимают значения на непрерывной шкале значений  $1, N$  для  $x_i$  и  $1, M$  для  $y_i$ . Это позволяет получить  $K \gg NM$  различных пар координат, соответствующих нерегулярной сетке в заданных границах. При этом  $z_i$  принимают целые значения, соответствующие пикселю RI с координатами  $x_i$  и  $y_i$ , округленным до ближайшего целого. Для придания значениям  $z_i$  свойств «реальных измерений» накладывается нормальный. На рис. 5, а приведено исходное изображение RI [3, 8] размером  $270 \times 210$  пикселей, а на рис. 5 б – то же изображение, представленное в пространстве 3D.

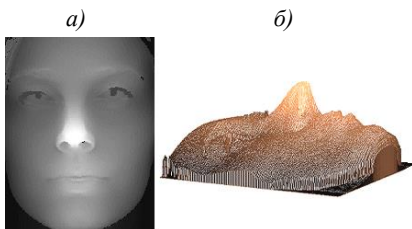


Рис. 5. Две формы представления исходного изображения RI

Стоит отметить прорисовку линий носа, глаз и области рта слева и «грубую зернистость» поверхности 3D изображения справа. Однако видно, что значения  $\{z_i\}$  RI достаточно точно представляют высоту всех точек на правом изображении по всей поверхности 3D. RI не содержит «черных дыр» и поэтому на изображении справа отсутствуют шипы или неопределенные значения.

На основе представленного выше подхода, получим из этого же изображения 550 значений  $\{z_i\}$  для всех вычисленных координат  $\{x_i, y_i\}$  (рис. 6, а). В результате получим облако исходных данных  $\{x_i, y_i, z_i\}$ , в котором координаты  $\{x_i, y_i\}$  размещены на нерегулярной сетке. Выполним обратный перенос координат  $\{x_i, y_i, z_i\}$  на регулярную сетку по МОР. Полученный результат показан на рис. 6, б. При этом для реконструкции всего изображения было использовано только 550 троек координат  $\{x_i, y_i, z_i\}$ . Результат достаточно точно отображает основные формы 3D в RI. Среднеквадратическая ошибка составила величину  $\approx 8$ , а вектор уравнения регрессии содержит всего 550 коэффициентов. В приложении к задаче редукции размерности пространства признаков, сокращение составило более 100 раз (поскольку  $270 \times 210 / 550 \approx 103$ ).

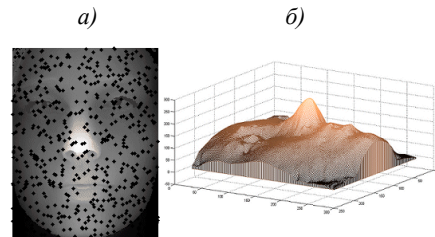


Рис. 6. Выбранные координаты и результат реконструкции

Другой способ сравнения результатов представлен на рис. 7. В верхней строке показаны исходное и реконструированное изображения, а в нижней строке – профили лиц для этих изображений по линии симметрии. По оси X отмечены номера строк RI, а по оси Y – значение пикселя, находящегося на линии симметрии RI. Тонкая линия – профиль лица для исходного изображения, а толстая – для реконструированного изображения. Хорошо видно, что профили идеально совпадают на участках лба, носа, рта и подбородка, несмотря на то, что новое – реконструированное изображение получено всего по 550 координатам, полученным случайным образом. Подобие линий профиля на участках лба, носа и подбородка можно использовать для сравнения двух изображений RI [6].

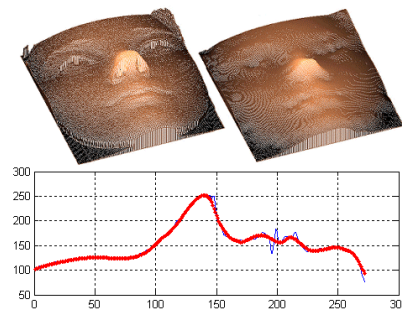


Рис. 7. Исходное изображение RI и результат реконструкции

## 4.2 Реконструкция при наличии «черных дыр»

Стоит отметить, что в принципе можно попытаться улучшить качество реконструкции RI за счет увеличения исходного числа точек  $K$  в облаке измерений, однако практически это трудно осуществить.

Действительно, чем больше число точек облака измерений, тем лучше можно реконструировать поверхность 3D области лица. Но, на практике мы столкнемся со следующими проблемами: при вычислении параметров регрессии (4) необходимо обращать матрицу расстояний  $D$ , порядок которой прямо связан с числом точек  $K$  в облаке измерений; если  $K < 1000$ , что обычно имеет место при моделировании задач реконструкции, то проблем с обращением матрицы нет, однако генерирование  $K$  троек координат  $\{x_i, y_i, z_i\}$  для  $K \approx 1000$  и при условии *некратности данных* в рамках ограниченных размеров  $M$  и  $N$  может вызвать определенные трудности.

Если осуществляется сканирование лица, то  $K \gg 1000$ , и тогда рассматриваемый класс задач уже не решить в рамках предложенного МОР. В случаях, когда  $K \gg 1000$  используются методы реконструкции, основанные на триангуляции. Однако, с другой стороны, этот метод не может быть эффективно использован в случаях, когда физическое сканирование отдельных областей лица сопровождается пропуском измерительных данных для этих областей. Именно в этом случае на поверхности RI появляются «черные дыры». «Латание этих дыр» основано на методах сплайн-интерполяции, но именно здесь возможно применение метода реконструкции, основанного на МОР.

Будем исходить из предположения, что 3D поверхность лица не имеет вертикальных (то есть по оси  $Z$ ) перепадов высот, поэтому в облаке данных не может быть точек с координатами  $\{x_i, y_i, 0\}$  для любых  $i \in L$  [9]. Следовательно, эти точки исключаются из дальнейшего рассмотрения. Кроме того, исключаются координаты, не удовлетворяющие условию «некратности данных». По оставшимся данным выполняется реконструкция RI с использованием МОР.

Пример исходного изображения (RI) с «черными дырами» показан в левом верхнем углу на рис. 9 [1]. «Дыры» на RI выделены черными локальными областями: всего здесь имеется 6 крупных дыр (центр носа, две области глаз и две области над бровями) и 6 мелких дыр. На оригинальном изображении RI проведена вертикальная линия по оси симметрии области лица и по ней определен профиль. В верхней части рис. 9 показано размещение точек с координатами  $\{x_i, y_i\}$  на плоскости  $XY$ , полученных генератором случайных чисел. При этом можно заметить, что некоторое количество точек попало на области «черных дыр». В правой верхней части рисунка показан результат реконструкции RI по координатам  $\{x_i, y_i, z_i\}$ , где  $z_i \neq 0$  и удовлетворяющих условию *некратности данных*. В эксперименте число таких координат было не более 300.

Нижняя часть рис. 9 представляет исходное изображение в форме 3D (слева), на котором выделены края видимых «черных дыр». С правой стороны показан результат 3D реконструкции RI, в котором уже отчетливо виден профиль человеческого лица (область носа и область над верхней губой, глазные ямы, лоб), что совсем не видно на изображении слева.

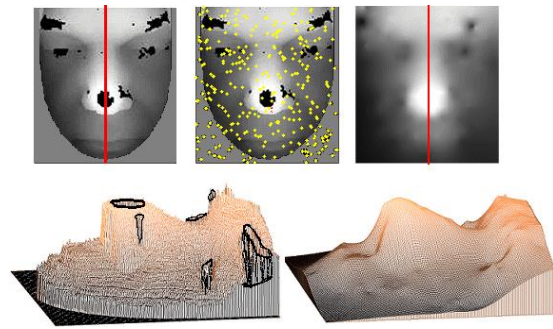


Рис. 9. Исходное изображение и результат его реконструкции

На рис. 10 тонкой линией показан профиль по линии симметрии для оригинала RI. Резкий перепад линии профиля (от 115 до 135) образован наибольшей «черной дырой», находящейся прямо в центре кончика носа на оригинале. Толстой линией показан профиль по линии симметрии для результата реконструкции RI. Видно, что профили линий симметрии практически везде точно совпали за исключением области «черной дыры». Именно в этом месте произошла реконструкция отсутствующей поверхности RI, и толстая линия прошла через реконструированный (предполагаемый) кончик носа. Среднеквадратическая ошибка в данном случае не превышала значения, равного 35.

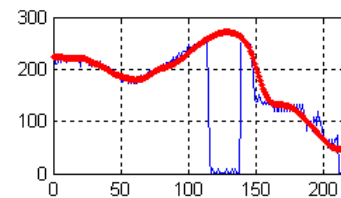


Рис. 10. Профили по линии симметрии RI и реконструкции

Время, затраченное на весь процесс реконструкции RI, составило не более 8с на компьютере Pentium Dual-Core CPU T4200, с тактовой частотой 2ГГц и 3ГБ ОЗУ.

## 5. ЗАКЛЮЧЕНИЕ

Исследования показали, что результат реконструкции RI по МОР не содержит *того количества и того качества* «черных дыр», которые были на исходном изображении, хорошо виден профиль человеческого лица, область носа, глазные ямы, лоб, что совсем не видно на исходном изображении RI, форма области лица по линии симметрии полностью восстановлена. Метод не требует значительных ресурсов. Все перечисленное является очевидным преимуществом предлагаемого метода и определяет возможность его использования на практике для восстановления 3D поверхности лица.

## 6. REFERENCES

- [1] Bowyer K.W., Chang K., Flynn P. A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition - Computer Vision and Image Understanding, Vol. 101/1, 2006, p. 1–15.
- [2] Gangwar S., Kumar K. 3D Face Recognition Based On Extracting PCA Methods - International Journal of Engineering Research and Applications, Vol. 2, Mar-Apr 2012, p.693-696.

- [3] Gupta K.R., Castleman M.K., Markey A.C. B., Texas 3D Face Recognition Database, IEEE Southwest Symposium on Image Analysis and Interpretation, Austin, Texas USA, May 2010, p. 97-100.
- [4] Kukharev G. Przetwarzanie i analiza obrazów cyfrowych – Wydawnictwo Uczelniane Politechniki Szczecińskiej, Szczecin, Poland, 1998, 137 s.
- [5] Kyong C. I., and et. Multiple Nose Region Matching for 3D Face Recognition under Varying Facial Expression - IEEE Transactions on PAMI, Vol. 28, № 10, 2006, p.1695-1700.
- [6] Манолов А.И. и др. Некооперативная биометрическая идентификация по 3D- моделям лица с использованием видеокамер высокого разрешения - [http://gc2009.raphicon.ru/files/gc2009/proceedings/conference/se8/139/139\\_Paper.pdf](http://gc2009.raphicon.ru/files/gc2009/proceedings/conference/se8/139/139_Paper.pdf)
- [7] Srivastava A., Liu X., Heshner C. Face Recognition Using Optimal Linear Components of Range Images – Image and Vision Computing, Volume 24 Issue 3, 2006, p. 291-299.
- [8] Szeptycki P., Ardabilian M., Chen L. A coarse-to-fine curvature analysis-based rotation invariant 3D face landmarking - <http://liris.cnrs.fr/Documents/Liris-4503.pdf>AuthorA.Reference1.
- [9] Shchegoleva N. Facial surface reconstruction in 3D format - Journal of Theoretical and Applied Computer Science. V.6, № 4. P. 37-50.

# Манипулирование Свободными Формами с Функциями Возмущения

С.И. Вяткин, Б.С. Долговесов, М.А. Городилов  
Институт автоматики и электрометрии, СО РАН,  
Новосибирск, Россия

[sivser@mail.ru](mailto:sivser@mail.ru), [bsd@iae.nsk.su](mailto:bsd@iae.nsk.su), [gorodilovm@gmail.com](mailto:gorodilovm@gmail.com)

## Аннотация

В данной статье рассматриваются свободные формы, синтезированные с помощью функций возмущения. Свободные формы имеют высокую степень гладкости и компактное описание. Для создания формы предлагается набор алгоритмов и программного обеспечения на основе функций возмущения. Интерактивная модификация модели и быстрая визуализация позволяют обеспечить манипулирование объектом с необходимой степенью детализации, для синтеза изображений реалистичного качества.

*Ключевые слова:* Функции возмущения, Геометрические объекты и операции, Интерактивное моделирование.

## 1. ВВЕДЕНИЕ

При формировании трехмерных сцен наиболее часто используется полигональное задание моделей объектов, которое на современном уровне развития компьютерной графики имеет ряд ограничений. Каркасные модели трехмерных объектов являются приближенными. Для достижения фотореализма необходимо использование свыше миллиона полигонов в сцене, причем наблюдается тенденция к дальнейшему увеличению детализации. Уже сегодня во многих приложениях количество треугольников сцены сопоставимо или превышает число пикселей, занимаемых на экране, что нивелирует преимущества полигонального подхода. При возрастании сложности сцены эффективность полигонального метода экспоненциально падает. Структура полигональных сеток линейна и они не обеспечивают поддержки многомасштабности, поэтому работа с большими сетками затруднена и требует вычислительно сложных методов упрощения. Полигональная модель принципиально не позволяет получить многих визуальных эффектов, необходимых для реалистичного отображения сцены. С помощью скелетной анимации нельзя сделать качественную анимацию гибких материалов, а также выполнить сложный морфинг объектов.

От этих недостатков можно избавиться, применяя аналитическое задание объектов и растеризацию их при помощи алгоритмов трассировки лучей. Аналитическое задание геометрических объектов не требует большого объема памяти. Существуют работы по визуализации функционально заданных поверхностей, таких как F-гер [1], поверхностей свертки [2], неявно заданных поверхностей, известных в компьютерной графике как капельные модели [3], метасфер [4], мягких объектов [5] и т. д. Однако их применение ограничено довольно узким классом моделируемых поверхностей и медленной визуализацией. Используемые алгоритмы сложно оптимизировать для визуализации в реальном времени, что также накладывает ограничения на их

практическое применение. Один из главных недостатков известных методов визуализации - сложность вычисления точек поверхности. Так, метод маршировки по лучу не гарантирует обнаружение поверхности, кроме того, он медленный [6]. В работе [7] описан метод вычисления пересечения луча с поверхностью, заданной в неявном виде, однако вычисления  $L$ - и  $G$ - параметров очень сложные. В методе сферы-трэйсинга нахождение наибольшего радиуса, чтобы ни одна точка объема не лежала внутри сферы - нетривиальная задача [8]. Такой метод можно эффективно применять только тогда, когда объекты статичны. То есть, когда форма и масштаб самих объектов не изменяются. В трассировке луча с анализом интервала для сложных функций требуется делать много вычислений, поскольку они необходимы индивидуально для каждого луча и для каждого интервала вдоль этого луча [9]. В быстрой трассировке луча поиск лучей, пересекающих поверхности сложен и не достаточно эффективен, поскольку способы кластеризации этого метода не решают данную проблему полностью [10]. В функциональных заданиях поверхностей вычисление определяющей функции в заданной точке часто оказывается весьма трудоемкой задачей, особенно в случае использования R-функций с квадратными корнями [1].

В работе [11] описан метод отслеживания лучей для визуализации поверхностей, заданных алгебраически полиномами высокой степени. Чтобы с помощью обычного полинома задать сложную поверхность необходимо увеличивать его степень. В случае с кривыми Безье гарантируется высокая точность нахождения пересечения с кривой, но не гарантируется то, насколько точно начальная функция будет приближена этой кривой. Ещё одним недостатком этого метода является то, что перевод объекта в другую систему координат – не простая задача. Поэтому создание динамических сцен становится проблематичным.

Целью данной работы является интерактивное моделирование и визуализация функционально заданных объектов на основе функций возмущения.

В последнее время компьютерная графика, связанная с интерактивным моделированием и редактированием трехмерных объектов развивается быстрыми темпами. Известны такие коммерческие системы как SoftImage, MAYA, 3DStudioMax и др. для интерактивного редактирования полигональных моделей. Существуют интерактивные системы на базе полиномов, неявных поверхностей [12], треугольных сеток [13], изображений [14], объемов [15], функционально заданных моделей [16, 17].

Конструирование объекта в нашем случае сводится к задаче деформации базовой поверхности нужным образом, а не к аппроксимации ее примитивами, данный процесс напоминает лепку модели из пластилина с применением геометрических операций, представленных ниже и деформации. Основное положительное отличие от известных интерактивных систем

на базе функционально заданных моделей это то, что нет необходимости в полигонизации функциональной модели перед деформацией.

## 2. ГЕОМЕТРИЧЕСКИЕ ОБЪЕКТЫ

Для описания сложных геометрических объектов используются функции отклонения (второго порядка) от базовой квадрики [18]. На базе квадрик строятся свободные формы. Свободная форма представляется композицией базовой квадрики и возмущений

$$F'(x, y, z) = F(x, y, z) + \sum_{i=1}^N f_i R_i(x, y, z) \quad (1)$$

где  $R_i(x, y, z)$  – возмущение,  $f_i$  – форм-фактор.

$$R_i(x, y, z) = \begin{cases} Q_i^3(x, y, z), & \text{if } Q_i(x, y, z) \geq 0 \\ 0, & \text{if } Q_i(x, y, z) < 0 \end{cases} \quad (2)$$

где  $Q_i(x, y, z)$  – возмущающая квадрика.

Для того чтобы поверхность была гладкой, степень должна быть больше двух (2). Это условие гарантирует непрерывность функции и её производной.

## 3. ГЕОМЕТРИЧЕСКИЕ ОПЕРАЦИИ

Геометрическая модель должна обеспечивать конструирование объектов и их композиций различной сложности. Для этого используется множество геометрических операций  $\Phi_j$ , определяемое математически следующим образом [1]:

$$\Phi_j: M^1 + M^2 + \dots + M^n \rightarrow M, \quad (3)$$

Для формирования моделей сложных объектов на базе функций возмущения используются такие операции, как проекции, офсеттинг, метаморфозис, кручение и заметание движущимся твердым телом [19], осуществляемые с применением булевых операций объединения и пересечения. Бинарная операция ( $n=2$ ) (3) объектов  $G_1$  и  $G_2$  означает операцию  $G_3 = \Phi_j(G_1, G_2)$  с определением:

$$f_3 = \psi(f_1(x, y, z), f_2(x, y, z)) \geq 0, \quad (4)$$

где  $\psi$  – непрерывная вещественная функция двух переменных. С помощью операции смещения (offsetting) можно создавать увеличенную или уменьшенную копию исходного объекта, то есть делать положительный или отрицательный офсеттинг соответственно. При метаморфозисе или морфинге осуществляется плавный переход одного объекта в другой. При функциональном задании объектов с применением функций возмущения можно осуществлять трехмерный морфинг негомеоморфных объектов (например, тора и сферы) и морфинг с ограничениями без разрыва поверхности и последующего “склеивания” последней. Два тела гомеоморфны, если из исходного тела путем взаимно однозначного непрерывного преобразования описывающей его функции можно получить второе. То есть для негомеоморфных объектов мы не можем простым приближением квадрик вычислить неразрывный морфинг. В процессе морфинга могут участвовать не два исходных объекта, а четыре. В этом случае схема интерполяции будет билинейной, а геометрическая операция – кватернарной ( $n=4$ ) (3). Кручение – это деформация тела, являющаяся частным

случаем биективного отображения. Биективное отображение служит для определения деформаций исходных объектов. Заметание – это проекция движущегося тела из  $4D(x, y, z, t)$  в  $3D(x, y, z)$  пространство. Эти операции не изменяют степень функций заданных моделей. Одним из примеров отношений может служить определение столкновений между объектами [20]. Бинарное отношение есть множество множества

$M^2 = M \times M$ . Оно может быть определено как:

$$Sj: M \times M \rightarrow I \quad (5)$$

С помощью особого теста на пересечение и бинарного поиска можно за постоянное число шагов (определяется заданной точностью) определить точку столкновения объектов, если таковое происходит. В целях расчета времени обнаружения столкновений тестировались объекты, различавшиеся как по степени сложности (форме), так и по виду столкновения (имеется в виду столкновение различными сторонами и частями объектов).

Более подробное описание трехмерных объектов на базе функций возмущения в монографии [21].

## 4. ВИЗУАЛИЗАЦИЯ

Главным на этом этапе является эффективное нахождение первого пересечения луча с поверхностью. Данная задача сходна с визуализацией объёмных томографических данных, где функция плотности задается в виде дискретных данных. А в нашем случае используется аналитически заданная функция, что позволяет более эффективно осуществлять поиск точек поверхности.

Для вычисления пересечения лучей с поверхностями трехмерных объектов предлагается метод, в котором отсутствуют вышеперечисленные недостатки, характерные для рассмотренных ранее известных методов аналитического задания поверхностей.

Для простоты понимания будем считать, что сцена находится в единичном трёхмерном кубе. Перспектива рассматриваться не будет, ввиду того, что она сводится к переходу в другую систему координат. Поэтому опустим начальные преобразования и уделим больше внимания основной части метода. Будем считать, что наблюдатель смотрит вдоль оси  $Z$ . Необходимо получить проекцию сцены на плоскость  $XY$ . Проекция должна представлять собой конечный набор значений. Через плоскость  $XY$  куба проходят лучи, и каждому лучу соответствует пиксел на изображении. Лучи ограничены передней и задней гранями куба. В процессе поиска точки пересечения луча и поверхности, каждый луч делится вдоль оси  $Z$ , образуя набор вокселей. Найдя ближайшую к наблюдателю точку для каждого луча, можно вычислить координату  $Z$ . Далее в каждом пикселе вычисляется нормаль. Имея все координаты и нормали в каждом пикселе, можно использовать модель локального освещения. В итоге получится изображение гладкого объекта с учётом освещения.

Уменьшение времени на визуализацию достигается за счёт эффективного использования вычислительных ресурсов графического акселератора с архитектурой CUDA от компании NVIDIA. Большое количество вычислительных процессоров позволяет параллельно проверять пересечение с объектом одновременно нескольких лучей. Следовательно, будет параллельно вычисляться достаточно большая часть куба [22].

## 5. ИНТЕРАКТИВНОЕ МОДЕЛИРОВАНИЕ

До недавнего времени удовлетворительные методы создания функциональных объектов отсутствовали. Объекты описывались вручную в текстовом виде. Эта операция чрезвычайно трудоемка. Поэтому задача создания инструментальной среды для редактирования объектов с применением функций возмущения стала очень актуальной. Были предложены, разработаны и реализованы структурные и алгоритмические решения по интерактивному объемно-ориентированному геометрическому моделированию функционально заданных объектов на основе аналитических функций возмущения, позволяющие пользователям легко и быстро создавать, моделировать и редактировать трехмерные объекты в интерактивном режиме. Для достижения поставленной задачи было сделано следующее: сформулированы требования к разрабатываемому программному обеспечению; разработан формат хранения данных в памяти; разработаны и реализованы алгоритмы работы с файлами и алгоритмы построения сцены из памяти. А также, разработан и реализован алгоритм добавления объекта в сцену; разработан и реализован алгоритм нахождения объектов в сцене; создан интерфейс к программе.

В результате была разработана интерактивная инструментальная среда, которая значительно упрощает процесс создания и редактирования функционально заданных объектов. Для визуализации был использован оптимизированный алгоритм многоуровневого отслеживания лучей, который осуществляет эффективный поиск точек поверхностей, участвующих в формировании изображения. Были разработаны и реализованы необходимые алгоритмы и C++ классы для интерактивного объемно-ориентированного геометрического моделирования: алгоритм рекурсивного многоуровневого отслеживания лучей (оптимизированный алгоритм рендеринга) сцен, содержащие функционально заданные объекты (включая OpenGL буферную совместимость - цвет/глубина); C++ классы функционально-заданных объектов; C++ классы рендеринга функционально заданных объектов; C++ классы интерфейса системы интерактивного объемно-ориентированного геометрического моделирования с возможностью простого механизма расширения новых алгоритмов и характеристик.

Эти иерархические классы составляют ядро пакета, который может быть расширен для добавления функциональности или изменения характеристик.

Другие задачи, включающие преобразование полигональных объектов в функциональные модели и обратное преобразование (триангуляцию), разработаны с использованием этих классов. В данной работе они не рассматриваются.

Для интерактивного моделирования необходимо было решить следующие задачи.

1) Реакцию программы на события от мыши (и, соответственно, выбор и модифицирование объектов). Необходимо уметь выделять объект (или несколько объектов) в сцене и предоставлять пользователю возможность проделать с ним (с ними) некоторые операции.

2) Выбор операции – аффинные преобразования MRS (move, rotate, scale), геометрические операции или деформация. Деформация состоит в возможности добавлять в любую точку на поверхности возмущение с параметрами

задаваемыми *инструментом*. *Инструмент* задает область действия и вид возмущения. Для этого необходимо графически предоставить пользователю некоторую информацию (в том числе выделять объект, например, цветом, или с помощью bounding box, рисовать оси и т.д.). Всю отрисовку «поддержки» работы с операциями лучше производить с помощью OpenGL – для скорости и функциональности.

3) Возможность работы со списком инструментов.

4) Запись в файл и загрузка дерева сцены.

5) Возможность создания новых тэгов и их распознавания.

Ниже показан псевдокод алгоритма добавления возмущения.

```
Setup temporary rendering context – camera matrix to restrict drawing to a small region of the viewport around the center of a picking region; clearing array of objects selected;
```

```
Render scene with RENDER_NAMES flag filling CVxManipulator::m_setObj set by calling VxDepthRende::PerformRender;
```

```
Sort objects in the CVxManipulator::m_setObj set
```

```
if the current design (manipulation) mode is “deformation” or Ctrl is pressed and there is any perturbation in the CVxManipulator::m_setObj set
```

```
remove all non-perturbation objects to ensure that will change only existing perturbations;
```

```
else remove all perturbation objects from CVxManipulator::m_setObj to ensure that will change only existing ‘base’ objects;
```

```
remove any descendant objects (since they would be altered automatically since their parents change).
```

Алгоритм 1: Алгоритм добавления возмущения.

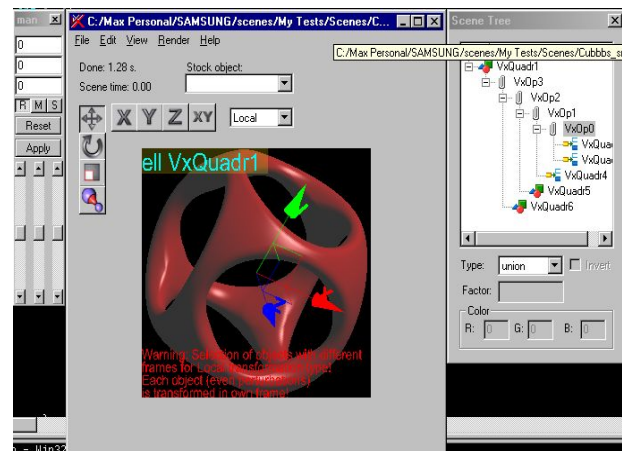


Рис 1: Главное окно программы.

## 6. ЗАКЛЮЧЕНИЕ

К основным достоинствам предлагаемых способа задания объектов и метода их визуализации следует отнести простоту вычисления точек поверхности. Задание объектов

поверхностями свободных форм сокращает в 100 и более раз описание баз данных по сравнению с заданием их полигонами.

Время визуализации зависит от сложности объекта. Для задания более сложных объектов требуется больше возмущений. Производительность сильно зависит от скорости работы памяти. Поэтому наибольшей производительности можно достичь при использовании только регистров графического акселератора.

В настоящее время в связи с развитием вычислительной техники численное решение задач трехмерной геометрии, требующих больших вычислительных ресурсов, находит все большее применение в реальных приложениях. Это моделирование динамики твердых и деформируемых тел, решение задач в области вычислительной биологии, молекулярного моделирования и т.д. Компьютерное трехмерное геометрическое моделирование позволяет получить и исследовать геометрические параметры объекта, проанализировать механические свойства, динамику поведения и взаимодействия объектов. Многообразие задач геометрического моделирования требует разработки эффективных методов и алгоритмов формирования компьютерных трехмерных геометрических моделей, способных обеспечить высококачественную и информативную визуализацию в реальном времени, используя стандартные современные программно-аппаратные средства. Функциональное задание объектов особенно актуально в ряде задач компьютерной графики, включая моделирование мягких или органических объектов, трехмерного морфинга, определения столкновений объектов и конструктивной блочной геометрии. Области применения функционально заданных объектов: молекулярная биология, интерактивные графические системы визуализации, CAD-системы, системы 3D-моделирования, 3D веб-визуализация, системы прототипирования и т. д.

## 7. ССЫЛКИ

- [1] Pasko A., Adzhiev V., Sourin, A., et al. Function representation in geometric modeling: concepts, implementation and applications //The Visual Computer, 11, 6, 1995, pp. 429-446.
- [2] McCormack J., Sherstyuk A. Creating and rendering convolution surfaces. Computing Graphics Forum. - 1998.-Vol. 17. - No.2. – pp. 113-120.
- [3] S. Muraki. Volumetric shape description of range data using “blobby model”. Computer Graphics. - July 1991. - 25(4) - pp. 227-235.
- [4] H. Nishimura, M. Hirai, T. Kawai, T. Kawata, I. Shirakawa, and K. Omura. Object modeling by distribution function and a method of image generation. The Transactions of the Institute of Electronics and Communication Engineers of Japan. - 1985.-J68-D (4) - pp. 718-725.
- [5] G. Wyvill, C. McPheeters, and B. Wyvill. Data structure for soft objects. The Visual Computer. - 1986. - 2(4) - pp. 227-234.
- [6] K. Perlin, E. M. Hoffert. Hypertexture. Proceedings of the 1989 ACM SIGGRAPH conference, Volume 23, Issue 3 (July 1989), pp. 253 – 262.
- [7] D. Karla and A.H. Barr. Guaranteed ray intersections with implicit surfaces. *Computer Graphics*, 23:297-306, November 1989.
- [8] J. C. Hart. Sphere tracing: a geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, 12:527-545, 1994.
- [9] D. Mitchel. Robust ray intersection with interval arithmetic. In *Proceedings on Graphics Interface 1990*, pp. 68-74. 1990.
- [10] A. Sherstyuk. Fast ray tracing of implicit surfaces. *Computer Graphics Forum*, Volume 18, Number 2, 1999, pp. 139-147.
- [11] M. Reimers, J. Seland. Ray Casting Algebraic Surfaces using the Frustum Form. *Eurographics*, Vol. 27 (2008), No. 2, pp. 361-370.
- [12] R.N. Perry and S.F. Frisken. Kizami: A System for Sculpting Digital Characters, in *SIGGRAPH'01*- 2001. – pp. 47-56.
- [13] M. Agrawala, A.C. Beers and M. Levoy, M. 1995. 3D Painting on Scanned Surfaces. - ISBN 0-89791-736-7. 1995- pp. 145-150.
- [14] B.M. OH, M. Chen, J. Dorsey and F. Durand. Image-based modeling and photoediting, in *SIGGRAPH'01*. - 2001. - pp. 433-442.
- [15] E. Ferley, M.-P. Cani and J.-D. Gascuel. Virtual Sculpture, *Visual Computer* 16:8. - 2000. - pp. 469-480.
- [16] Sourin. Functionally based virtual embossing. *The Visual Computer*.-7-2001. – pp. 258-271.
- [17] K.Levinski and A.Sourin, Interactive Function-Based Shape Modeling for Cyberworlds, 2004 International Conference on Cyberworlds, Tokyo, 18-20 November, 2004.-pp.54-61.
- [18] S. I. Vyatkin, “Complex Surface Modeling Using Perturbation Functions,” *Optoelectronics, Instrumentation and Data Processing*, Vol. 43, No. 3, 2007. pp. 40-47.
- [19] S. I. Vyatkin, B. S. Dolgovesov and A. T. Valetov, “Geometric Operations for Functionally Defined Objects Using Perturbation Functions,” *Optoelectronics, Instrumentation and Data Processing*, Vol. 40, No. 1, 2004, pp. 65-73.
- [20] S. I. Vyatkin, B. S. Dolgovesov and A. S. Korsun, “Collision Detection of Functionally Defined Objects in Computer Graphics Tasks,” *Optoelectronics, Instrumentation and Data Processing*, Vol. 39, No. 6. 2003, pp. 119-126.
- [21] Вяткин С.И. Трехмерные объекты на базе функций возмущения - LAP LAMBERT Academic Publishing GmbH & Co. KG, Saarbrücken Germany (ID: 45569) ISBN: 978-3-8484-2398-9 – 2012. (128 стр.)
- [22] Vyatkin S.I., Gorodilov M., Dolgovesov B.S. GPU-based Binary Adaptive Ray Casting for Freeform Objects with Perturbation Functions// Proceedings of the IASTED International Conferences on Automation, Control, and Information Technology (ACIT 2010), vol. “Information and Communication Technology”, “Optical Information Technology”. Acta Press, Calgary, 2010. pp. 223–228.

# ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ МОДЕЛЬ ПОСТРОЕНИЯ ФОТОРЕАЛИСТИЧНЫХ ИЗОБРАЖЕНИЙ

Жданов Дмитрий Дмитриевич

Санкт-Петербургский Национальный Исследовательский Университет ИТМО, Санкт-Петербург

Ершов Сергей Валентинович, Поздняков Сергей Георгиевич

Институт Прикладной Математики им. М.В. Келдыша РАН, Москва

## Аннотация

Данная статья посвящена разработке объектно-ориентированной компьютерной модели построения фотореалистичных изображений геометрически и оптически сложных сцен. В работе проводится классификация объектов сцены и предлагается универсальный программный интерфейс, достаточный для проведения физически корректных светотехнических вычислений и построений фотореалистичных изображений. Кроме того, иерархическая модель объектов сцены и оптических элементов позволяет на уровне стохастического лучевого моделирования строить геометрические и оптические модели практически не ограниченной сложности. В качестве примера приводится результат моделирования изображения, наблюдаемого на жидкокристаллическом дисплее автомобиля при разных условиях внешнего освещения.

**Ключевые слова:** Сцена, объект сцены, оптический элемент, программный интерфейс, рендеринг, стохастическое лучевое моделирование, фотореалистичное изображение.

## 1. ВВЕДЕНИЕ

Решение задачи построения физически корректного изображения включает в себя два основных программно-методических компонента. Во-первых, это методические, алгоритмические и программные модели трассировки лучей и построения изображения сцен с потенциально сложными компонентами. Во-вторых, это методические, алгоритмические и программные модели объектов сцены, используемые в программных алгоритмах трассировки лучей и решениях уравнения рендеринга, то есть базовые компоненты модели рендеринга. Оба программных компонента должны обеспечивать эффективную и физически корректную трассировку лучей в сценах, содержащих десятки и сотни миллионов геометрических объектов, допускающих сложные условия освещения и имеющих сложные оптические свойства, такие как двунаправленное рассеивание, поляризационное рассеивание, объемное рассеивание, двойное лучепреломление и т.п.

Однако эффективность и физическая корректность моделей отдельных программных компонентов рендеринга не гарантирует высокую эффективность и физическую корректность общей программной реализации рендеринга. Обязательным условием общей высокой эффективности и физической аккуратности рендеринга является согласованность программных интерфейсов всех объектов, формирующих общий программный продукт. Отсутствие согласованности может привести к потере физической корректности или эффективности вычислений. Например, если модель рассеивания света на поверхности обеспечивает физически корректное спектрально-поляризационное преобразование излучения, а программа трассировки лучей не позволяет передать состояние поляризации света от одного объекта сцены до другого, то физическая корректность модели данного объекта просто теряется.

Правильное проектирование программных интерфейсов должно учитывать сложную физическую организацию сцены. Сцена может содержать объекты, организация и программный интерфейс которых отличается от организации основных объектов сцены. Это так называемые оптические элементы [6], которые для сцены являются «черным ящиком», видимым только посредством определенного программного интерфейса.

Данная работа ориентирована главным образом на проектирование программных интерфейсов, связанных с вычислительной частью, оставляя за кадром вопросы, связанные с подготовкой, хранением и графическим представлением данных в пользовательском интерфейсе. Кроме того, вопросы поляризации (поляризационного рассеивания и распространения света) рассматриваются только для прямой трассировки лучей (от источника света до изображения).

## 2. ПРИМЕНЕНИЕ ДВУНАПРАВЛЕННОЙ СТОХАСТИЧЕСКОЙ ТРАССИРОВКИ ЛУЧЕЙ ДЛЯ РЕШЕНИЯ УРАВНЕНИЯ РЕНДЕРИНГА

Разработка программных интерфейсов рендеринга должна базироваться на выбранном подходе решения уравнения рендеринга. Уравнение рендеринга определяет яркость в направлении наблюдения как сумму собственной яркости наблюдаемого объекта и яркости, рассеянной данным объектом в направлении наблюдения [4]. Для статических сцен уравнение рендеринга, определяющее яркость объекта в точке  $\vec{p}$ , направлении  $\vec{v}$  и для цветовой компоненты  $c$ , можно записать следующим образом:

$$L(\vec{p}, \vec{v}, c) = \tau(\vec{p}, \vec{v}, c) \left( L_0(\vec{p}, \vec{v}, c) + \frac{1}{\pi} \int_{4\pi} BSDF(\vec{p}, \vec{v}, \vec{v}', c) E_\omega(\vec{p}, \vec{v}', c) d\omega \right) \quad (1)$$

где:

$L_0(\vec{p}, \vec{v}, c)$  – собственная яркость объекта в точке наблюдения,

$\tau(\vec{p}, \vec{v}, c)$  – пропускание (прозрачность) среды между наблюдателем и точкой наблюдения,

$BSDF(\vec{p}, \vec{v}, \vec{v}', c)$  – функция двунаправленного рассеивания от источника освещения в направлении  $\vec{v}'$  на наблюдателя,

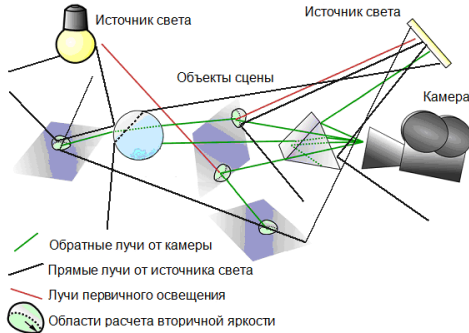
$E_\omega(\vec{p}, \vec{v}', c)$  – локальная освещенность объекта в точке наблюдения по направлению  $\vec{v}'$ , созданная источником света в телесном угле  $d\omega$ .

Основным методом решения уравнения рендеринга является метод трассировки лучей. В лучевом подходе интегрирование по сфере выполняется посредством трассировки лучей в направлении всех возможных источников излучения. На данный момент наиболее эффективным и универсальным решением уравнения (1) является двунаправленная стохастическая трассировка лучей, когда составляющая



первичной яркости вычисляется на трассах обратных лучей, а составляющая вторичной яркости – в областях пересечения трасс прямых и обратных лучей. Кроме того, данное решение является физически корректным, поскольку учитывает условия освещения и наблюдения в точках интегрирования яркости и в состоянии корректно выполнить интегрирование бесконечного цикла лучей на трассе от источников освещения до наблюдателя [1-3].

Двунаправленная стохастическая трассировка лучей схематично представлена на рис. 1.



**Рис 1:** Двунаправленная стохастическая трассировка лучей.

Циклы прямой и обратной трассировки лучей повторяются до тех пор, пока требуемое качество изображения не будет достигнуто.

Естественно, приведенная схема показывает лишь общую картину процесса построения изображения, однако она позволяет выделить основные программные компоненты модели рендеринга, участвующие в построении изображения:

- Программные модули трассировки прямых и обратных лучей.
- Программные модули сцены и ее компонентов.

Программная модель стохастической трассировки луча (прямого и обратного) полностью определяется программными моделями элементов сцены, определенными на уровне из базовых интерфейсов. Схематично алгоритм трассировки луча сводится к бесконечному циклу прямолинейных сегментов. Конечные и начальные параметры луча (координаты, направление, цвет, состояние поляризации) каждого сегмента обуславливаются граничными условиями сцены, такими как геометрия объектов сцены и оптические свойства геометрических поверхностей и сред. Стохастический выбор события, переводящий луч из одного сегмента в другой, осуществляется с помощью метода «русской рулетки», позволяющего математически корректно решать уравнение рендеринга (1), являющееся бесконечной суммой сегментов трассы луча.

### 3. БАЗОВЫЕ ИНТЕРФЕЙСЫ ОБЪЕКТОВ СЦЕНЫ

Основной задачей объектно-ориентированного проектирования является обеспечение взаимодействия объектов на уровне их базовых интерфейсов. Применительно к задаче рендеринга это значит, что объекты сцены и компоненты рендеринга взаимодействуют друг с другом только на уровне базовых интерфейсов. Например, при поиске пересечения траектории луча с геометрическим

объектом сцены, алгоритм не знает (и не должен знать), геометрический объект какого типа оказался на пути луча (элемент треугольной сетки, поверхность сплайна, аналитическая поверхность и т.п.). Алгоритму достаточно знать координаты точки пересечения луча с поверхностью и локальную информацию о поверхности в точке пересечения. Основные преимущества объектно-ориентированной организации это:

- Возможность изменения реализации объектов сцены без изменения алгоритма работы программы. Например, изменение способа хранения данных, определяющих двунаправленную функцию рассеивания, ограничивается данным объектом и не затрагивает компоненты рендеринга.
- Возможность реализации принципиально новых объектов сцены без изменения алгоритма работы программы. Например, для того, чтобы добавить новый тип аналитической геометрии сцены, достаточно лишь реализовать базовый интерфейс геометрического объекта для нового типа аналитической геометрии.

Реализация алгоритмов на уровне базовых интерфейсов приводит к тому, что рендеринг становится не зависимым от данных сцены. Как следствие упрощается алгоритм и программная реализация рендеринга, поскольку отпадает необходимость проверки типов объектов сцены и ветвлений алгоритма в зависимости от типа объекта сцены (упрощение алгоритма всегда приводит к большей надежности и устойчивости работы программы).

Специфика проектирования базовых интерфейсов объектов сцены и рендеринга заключается в том, физические принципы преобразования светового излучения на объектах сцены накладываются на модель стохастической двунаправленной трассировки луча, используемую в рендеринге. Поэтому программный интерфейс рендеринга и объектов сцены это своего рода парадигма, на которой строится процесс построения физически корректного изображения. То есть модель рендеринга и степень его физической корректности полностью определяется программными интерфейсами его компонентов.

Рассмотрим основные типы объектов сцены и их базовые интерфейсы:

- Геометрия сцены. Базовый метод геометрического объекта это поиск пересечения луча с поверхностью. Кроме факта пересечения данный метод должен возвращать локальную информацию о поверхности (дистанцию до поверхности, нормаль в точке пересечения, ссылку на оптические свойства поверхности и т.п.). Для ускорения трассировки лучей геометрия может использовать пространственную иерархию [7].
- Источники света. Базовые методы источника света должны информировать сцену о мощности излучения, генерировать случайный луч, испускаемый с заданной функцией распределения вероятностей, вычислять интенсивность и яркость излучения в заданном направлении и искать (аналогично геометрическому объекту) пересечение луча с объектом источника света.
- Приемник излучения. Базовые методы приемника излучения это прием и накопление лучей, идущих от источника света, испускание обратных лучей и вычисление

значений накопленной световой величины для заданных лучей.

- Оптические свойства поверхности. Базовые методы данного объекта должны обеспечивать вычисление первичной (вторичной) яркости поверхности и вычислять направление луча, рассеянного на поверхности с заданной функцией распределения вероятностей.
- Оптические свойства среды. Базовые методы данного объекта должны обеспечивать преломление / отражение луча на гладкой границе раздела двух сред, вычисление прямолинейной длины свободного пробега и цветового ослабления в среде, переносить луч на заданную дистанцию и вычислять направление луча, «рассеянного» на неоднородностях среды.

Хотя реальная программная система имеет более «широкий» программный интерфейс, приведенный базовый интерфейс является достаточным для построения физически корректных изображений.

#### 4. СПЕЦИАЛЬНЫЕ ОПТИЧЕСКИЕ ЭЛЕМЕНТЫ И ИХ ПРОГРАММНЫЕ ИНТЕРФЕЙСЫ

В ряде случаев, объекты сцены являются физически более сложными элементами, например, дифракционными элементами, и не могут быть описаны в рамках текущего объектно-ориентированного программного интерфейса. В качестве решения данной проблемы предлагается использовать специализированные объекты, которые, с одной стороны, позволяют решать задачи физически аккуратного моделирования, а с другой стороны, могут быть интегрированы в существующую объектно-ориентированную систему построения изображения. Для решения этой проблемы вводится концепция оптических элементов.

Оптический элемент – это объект программно моделируемой оптической системы или устройства, который позволяет использовать оптически более сложные модели взаимодействия света с объектами сцены, чем модели, предлагаемые текущим объектно-ориентированным программным интерфейсом [6].

Оптический элемент расширяет функциональность систем компьютерной графики, но при этом не раскрывает собственную функциональность. Поэтому с точки зрения сцены оптический элемент – это «черный ящик», который принимает на вход лучи, преобразует их и затем выдает их назад программе построения изображения, трассирующей лучи в сцене. Схематично процесс взаимодействия лучей с оптическим элементом показан на рис. 2.

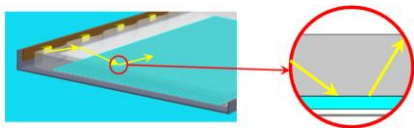


Рис 2: Распространение лучей в сцене, содержащей оптический элемент.

Поскольку оптический элемент является «черным ящиком», его содержимое скрыто от рендеринга, то есть изображение сцены не будет содержать изображения деталей оптического элемента. Такое поведение напоминает взгляд сквозь линзу. Мы не видим поверхности линзы, но видим изображение, которое она формирует.

По местоположению в сцене оптический элемент может быть независимым компонентом с собственной внешней границей или быть присоединенным к поверхности сцены. На рис. 3 представлены примеры микроструктурных рассеивающих оптических элементов, нанесенных на нижнюю поверхность светопроводящей пластины.

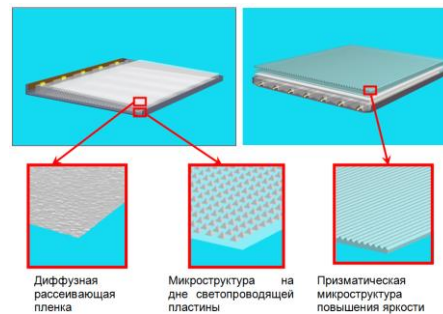


Рис 3: Микроструктурные рассеивающие оптические элементы.

Особенностью данных элементов является сложность рассеивающей микроструктуры, содержащей десятки (и сотни) миллионов независимых рассеивающих геометрических элементов сложной формы.

С точки зрения программной реализации оптический элемент это независимая специализированная сцена и ее базовый программный интерфейс должен покрывать основные объекты сцены:

- Геометрия сцены (поиск пересечения луча с оптическим элементом).
- Оптические свойства («расширенное» преобразование луча на оптическом элементе, когда точка входа луча в и выхода из оптического элемента могут не совпадать).
- Источник света (мощность излучения, генерация случайного луча и вычисление интенсивности и яркости излучения оптического элемента в заданном направлении).
- Приемник излучения (сохранение луча, падающего от источника света, испускание обратного луча и вычисление накопленных световых характеристик для заданного луча).

Необходимо отметить, что базовый интерфейс оптического элемента не предполагает методов расчета первичной и вторичной яркости на самом элементе, поскольку оптический элемент «прозрачен» для падающих на него лучей.

#### 5. ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ РЕАЛИЗАЦИЯ МЕТОДА ДВУНАПРАВЛЕННОЙ ТРАССИРОВКИ ЛУЧЕЙ

Объектно-ориентированный алгоритм двунаправленной стохастической трассировки лучей был реализован в программном комплексе INSPIRER2 [5]. Отличительными чертами разработанной программной модели являются:

- Реализация моделей объектов сцены и алгоритмов трассировки лучей в спектральном пространстве.
- Реализация поляризационных вычислений в моделях объектов сцены и методах прямой трассировки луча.
- Включение моделей оптических элементов как специальных объектов сцены и их поддержка в методах прямой и обратной стохастической трассировки лучей.

Программная модель позволяет создавать фотореалистичные изображения геометрически и оптически сложных сцен. В качестве примера приводятся результаты моделирования изображения экрана жидкокристаллического дисплея в салоне автомобиля при различных условиях внешнего освещения. Наиболее сложным элементом модели является жидкокристаллический дисплей, схематично представленный на рис. 4. Свет от светодиодных источников распространяется в среде светопроводящей пластины, рассеивается на микроструктуре, заданной с помощью соответствующего оптического элемента, и выходит наружу через ряд рассеивающих и поляризационных фильтров, включая TFT матрицу, формирующую изображение.

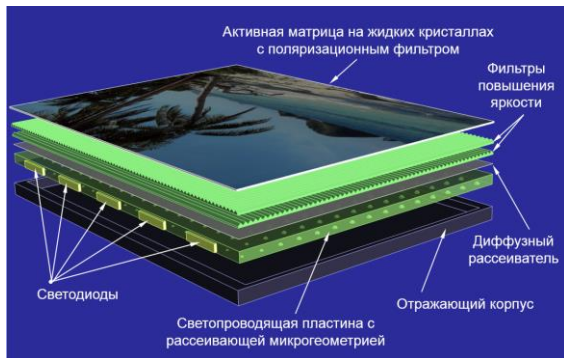


Рис 4: Модель жидкокристаллического дисплея.

Результаты моделирования при различных условиях внешнего освещения салона автомобиля представлены на рис. 5 (а-г).

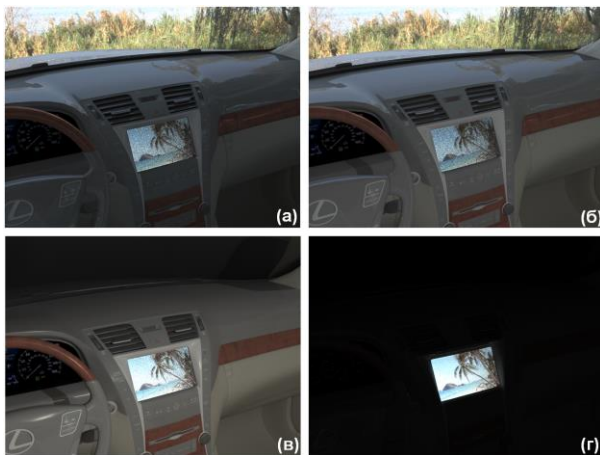


Рис 5: Результаты моделирования при различных условиях внешнего освещения.

На рис. 5(а) использовалось дневное освещение, заданное в HDR1 файле, на рис. 5(б) было добавлено салонное освещение, рис. 5(в) был создан для одного салонного освещения, а рис. 5(г) не имел никакого внешнего освещения.

Шум на изображении экрана дисплея объясняется спецификой работы алгоритма и частично связан с относительно малой мощностью (по отношению к внешнему освещению) светодиодных источников света жидкокристаллического дисплея.

## 6. ЗАКЛЮЧЕНИЕ

В рамках программного комплекса INSPIRER2 был реализован объектно-ориентированный алгоритм двунаправленной стохастической трассировки лучей. Данный алгоритм базируется на физически корректных моделях сцены и на уровне базовых интерфейсов позволяет осуществить физически корректный и эффективный рендеринг оптически и геометрически сложных сцен. Введенная концепция оптического элемента расширяет объектно-ориентированную модель рендеринга специальными объектами, которые не могут быть представлены в рамках базовой модели сцены.

Объектно-ориентированная модель рендеринга была реализована в спектральном пространстве. Поляризационные эффекты были поддержаны для прямой трассировки лучей, что обеспечивает физически корректный результат наблюдения в неполяризованном пространстве.

## 7. БЛАГОДАРНОСТИ

Работа поддержана грантами РФФИ № 12-01-00560 и 13-01-00454, а также компанией Integra Inc.

Авторы выражают свою признательность В.Г. Соколову и А.А. Гарбулю за предоставленные примеры.

## 8. ССЫЛКИ

- [1] Henrik Wann Jensen, Per Christensen. High quality rendering using ray tracing and photon mapping /SIGGRAPH '07.
- [2] Matt Pharr, Greg Humphreys. Physically Based Rendering - From Theory to Implementation // Morgan Kaufmann, 2004.
- [3] Toshiya Nishisaka and Henrik Wann Jensen. Stochastic progressive photon mapping. ACM Trans. Graph., 28(5):1-8, 2009.
- [4] Kajiya, J. T. The rendering equation / Siggraph 1986: 143.
- [5] <http://www.integra.jp/en/index.html>
- [6] Волобой А.Г., Галактионов В.А., Ершов С.В., Жданов Д.Д. Оптические элементы как средство расширения функциональности программ оптического моделирования // Труды 16-ой межд. конф., ГрафиКон' 2006, Россия, Новосибирск, июль 1-5, 2006, с. 182-191.
- [7] Ingo Wald, Carsten Benthin, and Philipp Slusallek. A Simple and Practical Method for Interactive Ray Tracing of Dynamic Scenes / report, Saarland University, 2002

## Abstract

The article is devoted to design of the object-oriented model of physically accurate rendering of the geometrically and optically complex scenes. The article proposes classification of the scene objects and base program interfaces for each kind of scene object which is sufficient for physically accurate light simulations and rendering. Moreover the article proposes the hierarchical scene organization which allows on the base of stochastic ray tracing creating very complex models from viewpoint of geometrical and optical scene organization. As an example the article is illustrated with rendering results of LCD display of car dashboard under different conditions of external illumination.

**Keywords:** Scene, scene object, optical element, program interface, stochastic ray tracing, physically accurate rendering.

## Построение реалистичных изображений в системах автоматизированного проектирования

Б.Х. Барладян, А.Г. Волобой, Л.З. Шапиро

Институт прикладной математики им. М.В.Келдыша РАН, Москва

### Аннотация

Рассмотрены различные подходы к интеграции синтеза реалистичных изображений и оптического моделирования в системы автоматизированного проектирования. Предложен подход, обеспечивающий эффективную интеграцию существующих систем синтеза реалистичных изображений в системы автоматизированного проектирования. Рассмотрен пример интеграции такой системы Inspire2 в систему автоматизированного проектирования CATIA.

*Ключевые слова:* САПР, автоматизация проектирования, реалистичные изображения, оптическое моделирование.

### 1. ВВЕДЕНИЕ

Обеспечение построения реалистичных изображений и оптического моделирования в системах автоматизированного проектирования является трудной задачей. Известны попытки ее решения, разработанные компаниями Discreet (Lightscape), Mental Images и Chaos Group (VRay) [1] для 3DS Max. В системе КОМПАС автоматизировано конструирование узла оптического прибора [2], но отсутствует моделирование. Компания OPTIS выпустила встроенный модуль моделирования освещенности для системы CATIA - SPEOS CAA V5 Based и для системы SolidWorks [3, 4]. Компания Mental Images интегрировала свою программу моделирования освещенности в Autodesk 3DS MAX, Maya и CATIA(PhotoStudio).

Специфический вариант интеграции системы Inspire2 в CATIA был предложен в работе [5]. По существу это конвертор из внутреннего формата CATIA во входной формат системы Inspire2. Используя дополнительную, специально сохраненную информацию, он позволяет обновлять геометрию в системе Inspire2 после ее модификации в CATIA. Этот подход позволяет существенно облегчить работу пользователя при модификации геометрии, но не решает главной задачи интеграции – пользователь вынужден работать в двух системах. Часть данных хранится в CATIA (геометрия), другая часть (атрибуты, источники света и ряд вспомогательных параметров) – в Inspire2.

Прямой путь полной интеграции требует, как минимум, полного перепрограммирования всего пользовательского интерфейса данной системы с использованием соответствующего набора инструментальных средств разработки программного обеспечения, поставляемой вместе с соответствующей системой. Для CATIA V5 это Rapid Application Development Environment (RADE). На самом деле такой прямой подход требует фактически и некоторого перепрограммирования ядра системы, поскольку RADE накладывает определенные ограничения на объекты в системе, на правила их взаимодействия, хранения данных и т.д. Разумеется, такой подход неэффективен, поскольку

требует слишком большого времени на реализацию. Поэтому была поставлена задача разработать более эффективную, компромиссную схему интеграции. Основные требования при создании интегрированной системы можно сформулировать следующим образом:

1. Максимально полное использование модулей уже существующей и отлаженной системы синтеза реалистичных изображений. Кроме ядра системы желательно также использовать некоторые специфические модули пользовательского интерфейса. Это позволит сэкономить время разработки, а также сохранить привычный интерфейс для пользователей базовой системы синтеза изображений и оптического моделирования.
2. Интегрированная система должна ощущаться пользователем как единая система – т.е. все основные данные для моделирования, такие как геометрия, атрибуты, должны храниться в системе автоматизированного проектирования.

Именно выполнение обоих этих требований позволит создать в реальные сроки интегрированную систему достаточно привлекательную для пользователей.

### 2. СИСТЕМА РЕАЛИСТИЧНОЙ КОМПЬЮТЕРНОЙ ГРАФИКИ

В настоящее время системы синтеза реалистичных изображений и оптического моделирования достигли высокого уровня развития. Так, например, система Inspire2 [6], разработанная в Институте прикладной математики им. М.В. Келдыша РАН, основанная на физически корректном моделировании распространения света, обеспечивает синтез реалистичных изображений с учетом различных сложных физических эффектов. Построенные изображения близки по восприятию к фотографиям реальных объектов. Такие изображения используются в проектах архитектурного проектирования, градостроительстве, ландшафтном дизайне, автомобильной и авиационной промышленности. Система позволяет учитывать такие сложные оптические эффекты, как рассеивание света на стохастических микронеровностях (шероховатостях) поверхностей и на регулярных микрорельефах поверхностей (десятки и сотни миллионов микро-призм заданной формы, размера и распределения), объемное рассеивание света на микрочастицах внутри материала и ряд других подобных явлений. Система поддерживает различные модели источников света, в частности, с определенной диаграммой излучения (гониограммой), в формате RaySet [7]. Также возможно задание стандартной CIE модели естественного дневного освещения и изображения с большим динамическим диапазоном в качестве источника света. Система обеспечивает моделирование как в RGB, так и в спектральном цветовом пространстве, моделирование распространения

света во флуоресцентных средах, моделирование прямой и не прямой освещенности методами Монте-Карло трассировки лучей. Обеспечивается также получение различных оптических характеристик сцены, таких, как распределение освещенности и яркости (угловое и пространственное) на различных реальных или виртуальных поверхностях.

Система удовлетворяет всем основным требованиям к современным системам синтеза реалистичных изображений и оптического моделирования – использует физически аккуратные методы оптического моделирования, позволяет учитывать сложные физические эффекты и измеренные параметры источников света, объемные и поверхностные свойства материалов. Основным недостатком системы Inspire2 является ее изолированность от систем автоматизированного проектирования, в которых в настоящее время ведется разработка и сопровождение в эксплуатации сложных технических изделий.

Так называемые технологии управления жизненным циклом изделий (Product Lifecycle Management (PLM)) обеспечивают управление всей информацией об изделии и связанных с ним процессах на протяжении всего его жизненного цикла, начиная с проектирования и производства до снятия с эксплуатации. Информация об объекте, содержащаяся в PLM системе является цифровым макетом этого объекта. Геометрическая модель объекта в таких системах создается, как правило, с помощью специализированных систем автоматизированного проектирования (САПР). Синтез реалистичных изображений для объектов, разрабатываемых в этих системах, а также получение их различных оптических характеристик таких, как распределение освещенности и яркости на различных реальных или виртуальных поверхностях, является важной частью общего процесса проектирования. Для обеспечения эффективного использования подсистемы оптического моделирования в проектировании необходимо обеспечить максимально гладкое ее взаимодействие с системой автоматизированного проектирования, где создается геометрия объектов, и хранятся их атрибуты. Разработчик изделия в системе автоматизированного проектирования должен использовать оптическую подсистему просто как одну из подсистем, которая автоматически воспринимает геометрию, созданную в системе, и те атрибуты, которые должны быть учтены в моделировании. Соответственно оптическая подсистема должна учитывать все те изменения в геометрии или атрибутах, которые могут вноситься на различных этапах проектирования изделий. Без такой эффективной интеграции разработчики, скорее всего, не будут пользоваться подсистемой синтеза реалистичных изображений.

Для трехмерных геометрических объектов «де факто» стандартом такой системы является CATIA [8]. По крайней мере, ее используют большинство авиационных, кораблестроительных и автомобильных компаний. Именно по этой причине было принято решение об интеграции системы Inspire2 в CATIA. Разработанный подход может быть использован и для других систем автоматизированного проектирования.

### 3. СХЕМА ИНТЕГРАЦИИ INSPIRER CATIA.

Основные и вспомогательные модули интегрированной системы Inspire CATIA представлены на рис. 1.

## Основные блоки Inspire CATIA и их взаимодействие



Рис. 1. Основные компоненты системы Inspire CATIA.

Пользовательский интерфейс системы CATIA написан с использованием библиотеки MFC, в то время как пользовательский интерфейс остальных компонент использует QT SDK. Совместное использование этих двух библиотек в одном выполняемом модуле невозможно, поэтому каждая из компонент на рис. 1 выполнена в виде отдельного выполняемого модуля.

**CATIA** – это базовая система автоматизированного проектирования. С точки зрения системы моделирования освещенности она обеспечивает создание и модификацию геометрии объектов сцены и хранение данных. Для целей моделирования освещенности она была расширена пользовательскими интерфейсами для описания и редактирования оптических атрибутов поверхностей, источников света и виртуальных измерительных объектов. На базе Inspire2 SDK была также обеспечена возможность расчета не прямой освещенности поверхностей, измерения оптических характеристик сцены с помощью виртуальных измерительных объектов и получение реалистичных изображений с помощью трассировки лучей. Эти базовые функциональности реализованы непосредственно в CATIA.

Пользовательские интерфейсы для задания источников света и оптических атрибутов поверхностей реализованы как дополнительные закладки в соответствующих собственных интерфейсах системы CATIA с использованием части собственных атрибутов. Такой подход облегчает освоение системы пользователями CATIA, поскольку они должны вводить только атрибуты, специфические для оптического моделирования. Виртуальные измерительные объекты были реализованы как собственные объекты CATIA. Они привязаны к геометрическим объектам и визуализируются непосредственно в окнах системы CATIA.

**I2 Server**, реализованный на базе Inspire2 SDK, обеспечивает дополнительные специфические функциональности системы Inspire2. В настоящее время этот модуль содержит такие пользовательские интерфейсы как, например, описание параметров среды распространения света, доступ к Inspire2 библиотекам описания сред, редактирование сложных критериев при визуализации лучей, трассировку лучей методом Монте-Карло, ряд вспомогательных пользовательских и программных интерфейсов.

**HDRIPostprocessor** позволяет производить обработку полученных реалистичных изображений и анализ результатов

измерений полученных на виртуальных измерительных приборах. В частности он позволяет:

- Преобразовывать полученные изображения в физических величинах (изображения с высоким динамическим диапазоном) в изображения в RGB пространстве графического монитора, используя различные операторы сжатия динамического диапазона яркостей.
- Отображать результаты моделирования с использованием тоновой заливки (искусственные цвета).
- Анализировать результаты моделирования, полученные в RGB и спектральных цветовых пространствах.
- Анализировать результаты моделирования, полученные на различных видах виртуальных сенсоров – угловое распределение яркости, распределение яркости или освещенности на прямоугольнике виртуального сенсора и др..

Система также содержит такие вспомогательные компоненты как редактор гониометрических диаграмм и редактор функции дисторсии объектива. Взаимодействие и синхронизация компонент осуществляется с помощью сообщений и событий операционной системы Windows. Данные между компонентами передаются с использованием общей памяти разделяемой процессами (shared memory). I2 Server может быть относительно легко расширен для поддержки в Inspirer2. Добавление, например, новой модели среды распространения света вообще происходит автоматически, поскольку и соответствующий пользовательский интерфейс, объект в котором сохраняются данные и само моделирование (Монте-Карло трассировка) реализованы в компонентах (DLL) общих для I2 Server и Inspirer2. Если новая функциональность требует передачи дополнительных данных между блоками, то это требует реализации лишь относительно небольших программных надстроек в соответствующих блоках.

#### 4. СИНХРОНИЗАЦИЯ КОМПОНЕНТ В СИСТЕМЕ INSPIRER CATIA.

Основное требование к синхронизации компонент это обеспечение максимально гладкого взаимодействия между блоками. Необходимо было обеспечить пользователю достаточно комфортную среду для работы с системой, чтобы она воспринималась бы как единое целое. Взаимодействие между блоками должно быть максимально скрыто от него.

I2 Server был реализован как обычное Windows приложение. Это приложение стартует одновременно с CATIA. Его главное окно скрыто. Его главная оконная процедура заменена на обработку сообщений, посылаемых CATIA. Специальные Windows события используются для синхронизации CATIA и I2 Server в процессе вычислений и перерисовки окна CATIA.

Рассмотрим синхронизацию компонент более детально на примере визуализации распространения света в оптической системе в виде трассируемых лучей. На первом шаге пользователь выбирает источники света, геометрические объекты и виртуальные сенсоры, которые будут участвовать в данном моделировании. Это, естественно, делается в диалоге, реализованном непосредственно в CATIA. Пользователь выбирает объекты непосредственно в окне визуализации или

в дереве сцены. Имена выбранных объектов отображаются в соответствующих окнах. В этом же диалоге можно установить ограничение времени моделирования, количество трассируемых лучей и другие необходимые параметры. После завершения выбора параметров моделирования соответствующий элемент появляется в поддереве моделирования. Теперь моделирование может быть запущено с помощью выпадающего меню.

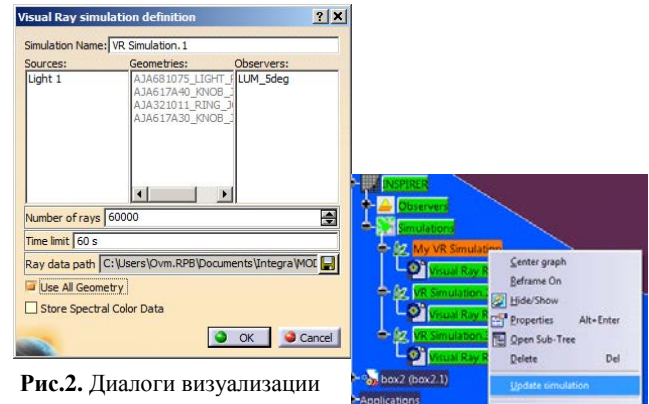


Рис.2. Диалоги визуализации трассируемых лучей.

Непосредственно моделирование выполняется в I2 Server. CATIA формирует описание сцены в формате IOF (бинарный формат системы Inspirer2), сохраняет его на диске и посылает I2 Server команду IDM\_VR\_SCENE\_LOAD (загрузки сцены), а за тем IDM\_VR\_CALCULATE (моделирование визуализации лучей). Дополнительные данные (путь на сохраненное описание сцены, ограничение по времени, количество лучей и т.д.) передается при этом с использованием общей памяти разделяемой процессами (shared memory).

Выполнение каждой команды в I2 Server синхронизируется с CATIA по общей схеме.

1. CATIA сбрасывает первое синхронизирующее событие (*event1*)
2. CATIA посылает команду и ждет взведения *event1*.
3. I2 Server извлекает данные из общей памяти, помещает дескриптор окна в общую память.
4. I2 Server сбрасывает событие *event2* и взводит *event1*. Ждет взведения события *event2*.
5. CATIA извлекает из общей памяти дескриптор окна, сбрасывает *event1* и взводит *event2*.
6. I2 Server дождавшись взведения *event2*, выполняет команду моделирования и, после завершения, взводит *event1*. Во время моделирования индикатор показывает степень выполнения команды: отношение количества отмоделированных лучей к тому, которое заказано в процентах. Результаты моделирования I2 Server записывает на диск в специальном формате.
7. CATIA, ожидая взведения *event1*, каждые 100 миллисекунд выполняет перерисовку своего окна и делает окно I2 Server активным.

По этой же схеме осуществляется синхронизация работы CATIA с HDRIPostprocessor, редакторами гониометрической диаграммы и функции дисторсии объектива.

## 5. ЗАКЛЮЧЕНИЕ

Предложенный в работе подход был использован для интеграции в систему САТИА основных функциональностей системы Inspire2, таких как вычисление прямой и не прямой освещенности, визуализация трассировки лучей и поддержки сложных оптических атрибутов и источников света. Подход позволил существенно ускорить разработку интегрированной системы и повысить ее надежность, поскольку наиболее сложные модули уже прошли отладку и тестирование в автономной системе. В результате время реализации новой функциональности в САТИА, уже существующей в Inspire2, сокращается многократно. Это особенно важно, так как стоимость среды программирования в САТИА (стоимость лицензии) довольно высока и само программирование требует квалифицированных специалистов со значительным опытом работы в этой среде.

На рис.3. приведены результаты визуализации автомобиля в Inspire САТИА при естественном освещении и использовании различных красок, задаваемых двунаправленной функцией отражения:



Рис3. Примеры визуализации в Inspire САТИА.

Работа поддержана грантами РФФИ № 11-01-00870, 12-01-00560, а также фирмой Integra Inc. (Япония).

## 6. СПИСОК ЛИТЕРАТУРЫ

- [1] А. Осинев. V-Ray — цените свое время. "САПР и графика", № 9, 2010, с. 88-89.
- [2] А. Колпаков, Н. Толстоба. Автоматизация конструирования узла оптического прибора. "САПР и графика", № 12, 2012, с. 68-70.
- [3] П. Гольдовский, А. Кокова. Моделирование оптических явлений и свойств задаваемых изделий. "САПР и графика", № 8, 2004, с. 46-47.
- [4] А. Алямовский. SolidWorks/OptisWorks — интегрированная среда анализа и синтеза в светотехнике и оптике. Оптический анализ и структура программного комплекса. "САПР и графика", № 4, 2006, с. 73-79.
- [5] Б.Х. Барладян, А.Г. Волобой, Л.З. Шапиро. Интеграция моделирования освещенности методом трассировки лучей в системы автоматизированного проектирования Труды 16-ой международной конференции по компьютерной графике и ее приложениям - ГрафиКон'2006, Россия, Новосибирск, 2006, с. 275-278.
- [6] Д.Д. Жданов, И.С. Потемин, В.А. Галактионов, Б.Х. Барладян, К.А. Востряков, Л.З. Шапиро. Спектральная трассировка лучей в задачах построения фотореалистичных изображений "Программирование", № 5, 2011, с. 13-26.
- [7] Radiant Imaging (Radiant ZEMAX, LLC), company website, Light Source Analysis and Ray Set Generation, <http://www.radiantzemax.com/prosource/>
- [8] Dassault Systems, company site, САТИА — цифровая продукция, <http://www.3ds.com/ru/products/catia/>

## Abstract

Various approaches to the integration of system of realistic rendering and optical modeling in computer-aided design system are considered. An approach that provides effective integration of existing realistic rendering systems in CAD system is proposed. An example of such integration of the Inspire2 system in CAD САТИА is considered.

**Keywords:** CAD, optics design, realistic rendering, lighting simulation

## Authors:

Борис Хаимович Барладян, к.т.н., с.н.с., ИПМ им. М.В. Келдыша РАН. E-mail: obb@gin.keldysh.ru

Алексей Геннадьевич Волобой, д.ф.-м.н., с.н.с., ИПМ им. М.В. Келдыша РАН. E-mail: voloboy@gin.keldysh.ru

Лев Залманович Шапиро, к.ф.-м.н., с.н.с., ИПМ им. М.В. Келдыша РАН. E-mail: pls@gin.keldysh.ru

## Программно-аппаратный комплекс интерактивных мультимедийных презентаций

Б.С. Мазурок, Б.С. Долговесов, Е.И. Коростелев, Т.Н. Артиков, А.Н. Артиков  
Институт Автоматики и Электротметрии СО РАН, Новосибирск, Россия  
kore3d@gmail.com

### Аннотация

В данной работе представлены особенности реализации современного программно-аппаратного комплекса интерактивных презентаций. Разработанная презентационная система предоставляет широкие возможности для подготовки и демонстрации различных мультимедийных материалов. Обеспечение интерактивного совмещения виртуальных трехмерных объектов и мультимедийных данных (видео, презентации, текст) с изображениями реальных персонажей позволяет существенно улучшить качество подачи материала.

**Ключевые слова:** виртуальный презентер, мультимедиа, визуализация реального времени.

### 1. ВВЕДЕНИЕ

В последнее десятилетие происходит развитие и адаптация систем виртуальной реальности для подготовки демонстрационных и образовательных материалов: учебные демонстрации, демонстрация товаров, визуализация бизнес процессов. Разрабатываемые системы строятся на базе технологии интегрированной виртуальной реальности. Данная технология предполагает интерактивное объединение в виртуальной среде моделей пространственных объектов, явлений и процессов с изображением реального персонажа (лектора, экскурсовода). Такое решение способствует существенно лучшему восприятию и пониманию излагаемого материала.

Авторами работы ранее разработана и представлена соответствующая система [1,3], успешно применяемая в образовательных целях (подготовка видеопособий, учебный процесс в специализированных школах). Основываясь на многолетнем опыте ее разработки и применения, а также современных требованиях, разработан программно-аппаратный комплекс виртуального презентера.

Разработанное решение обеспечивает формирование и визуализацию изображений (включая стереоизображения) виртуального окружения различной тематической направленности в комбинации с видеоизображением докладчика, интерактивно взаимодействующего с моделями демонстрируемых образцов. Интерфейс пользователя (рис. 1) ориентирован на использование сенсорного экрана и позволяет в несколько касаний выбрать демонстрируемый медиа-контент. Таким образом, докладчик, без привлечения сторонних специалистов, легко управляет ходом презентации, а также может переключать режимы микширования, медиа-источники и использовать вещание в сеть.



Рис. 1: Интерфейс управления презентацией

### 2. СТРУКТУРА КОМПЛЕКСА

Программно-аппаратный комплекс включает следующие аппаратные компоненты:

**Система визуализации**, выполненная на базе современного ПК – компьютера с высокопроизводительным графическим процессором.

**Система отображения** результирующего изображения (виртуальной среды, совмещенной с видеоизображением докладчика) на сенсорный монитор, проектор.

**Интерактивные средства управления презентацией:** джойстик, манипулятор «мышь», планшетный ПК.

**Съемочное оборудование** для записи видеоизображения докладчика, специализированный одноцветный фон (синий или зеленый) система освещения и звукозаписи.

Программное обеспечение комплекса также состоит из набора компонентов. Приведем некоторые из них:

**Менеджер ресурсов** обеспечивает экспорт и загрузку ресурсов сцены, управление интерактивными средствами.

**Подсистема визуализации** осуществляет отображение трехмерных сцен, текстурирование поверхностей трехмерных моделей, выполнение шейдерных операций.

**Менеджер сцены** осуществляет управление содержимым сцены в процессе отображения.

В следующем разделе рассмотрены особенности и специфичные для решаемых задач функции подсистемы визуализации. Их производительность существенно влияет на производительность всей системы, а набор реализованных в нём функций определяет применимость разработанной системы визуализации.

При разработке основы программных компонент существенное внимание уделялось гибкости системы: расширения системы новыми компонентами и конфигурирование всех параметров системы.



Расширяемость современных систем визуализации обеспечивается благодаря использованию компонентной архитектуры. В разработанной системе был расширен данный подход и предложена унифицированная структура хранения и обработки данных. В работе используется разработанный формат XQL, иерархическая база данных и алгоритмы ее функционирования [4]. Особенностью выбранного подхода является: межмодульное взаимодействие через базу данных (текстовый тегируемый протокол обмена), хранение данных произвольного типа (в том числе последовательности команд). Таким образом, интерфейсы модулей отделены друг от друга и зависимости между ними определяются только данными. При таком подходе модули более ранних версий интерпретируют только известные им данные, а неизвестные (добавленные для более поздних версий) просто игнорируют.

База данных используется для доступа к произвольным свойствам системы. Допускается частичное дублирование данных в программных модулях для более эффективного использования аппаратных ресурсов (отображение в видеопамять, предварительная компиляция эффектов, кэширование крупных структур). Для того, чтобы обеспечить обновление этих данных, применяется система обработки событий – каждый модуль регистрируется на уведомления о чтении и записи определенного набора структур в базу данных.

### 3. ПОДСИСТЕМА ВИЗУАЛИЗАЦИИ

Современные системы визуализации поддерживают десятки визуальных спецэффектов [2,6]. Кроме этого, используемые в сцене изображения могут обрабатываться различными фильтрами (например, фильтром гамма-коррекции, выделения актёра из монохромного фона). А на последнем этапе построения кадра используется последовательность спецэффектов (глубина резкости, свечение и другие).

Спецификой визуализации разработанного комплекса является способность формировать изображение интегрированной виртуальной среды (схема представлена на рис. 2). Изображение реального персонажа (лектора), взаимодействующего с изучаемыми виртуальными моделями, совмещается с тематическим виртуальным фоном и виртуальной моделью изучаемого объекта.

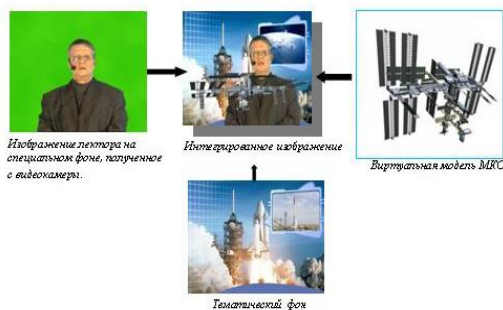


Рис. 2: Принцип формирования интегрированной виртуальной среды

В следующих подразделах рассматриваются основные компоненты подсистемы. Предлагаемый метод выделения актёра вынесен в соответствующий раздел.

### 3.1 Виртуальная сцена

Описание трехмерной сцены включает: описание трехмерных объектов, последовательностей их изменения во времени, настройки эффектов (параметры шейдеров), свойства отображения (фиксированное разрешение, аспект) и другие параметры. Все эти данные располагаются в иерархическом виде в разработанной базе данных.

В начале подготовки нового кадра производится обход графа сцены (иерархических связей), в процессе которого определяются видимые объекты (отсечение по пирамиде видимости с использованием ограничивающих сфер), строится список объектов для растеризации, производится их сортировка по материалам и в зависимости от удаленности от текущей виртуальной камеры. В узлах графа сцены располагаются виртуальные камеры, источники света и объекты. При этом виртуальный объект сцены может быть, либо обычной трехмерной геометрией (сеткой из треугольников), либо плоскостью, ориентированной на наблюдателя, либо ортогонально проецируемой поверхностью (визуализация элементов пользовательского интерфейса либо режима «оверлей») с текстурованием «тексель-в-пиксель». Все объекты сцены могут быть заданы в Autodesk 3ds Max и загружены в систему визуализации по необходимости.

Любые объекты сцены могут быть анимированы. Так, с течением времени могут меняться свойства объекта, траектории движения и позиция на траектории. Изменение анимационного времени, в зависимости от требований, рассчитывается по среднему времени кадра (для сохранения плавности анимации при буферизации кадров), либо синхронизировано с видеопотоком.

### 3.2 Система материалов

Материал - совокупность визуальных свойств поверхности объекта. Изображение вычисляется на основе описания его геометрии, материала, свойств сцены (например, освещение и туман). В ходе вычислений блоки графического процессора исполняют специальный код (шейдер), определяющий методы обработки поверхности объекта. Для всех трёхмерных объектов может быть указан универсальный шейдер, но это неэффективно, поскольку приводит к существенным потерям производительности. Вместо этого реализована автоматизированная генерация кода шейдера, в зависимости от свойств материала и ситуации на сцене.

В разработанной системе вводятся новые типы шейдеров: шейдер поверхности, освещения. Кроме этого, при их сборке в шейдер для графического конвейера (пиксельный шейдер) используется условная компиляции. Такой подход позволяет описывать поверхность объекта (цвет, нормаль) и модель расчета освещения независимо друг от друга и комбинировать их произвольным образом. При этом за счет условной компиляции из кода шейдера исключаются лишние инструкции, что улучшает производительность,

При разработке системы освещения основной задачей было повторение локальной модели освещения Autodesk 3ds Max (модель освещения Фонга). Также реализованы более сложные модели – например, освещение по Куку-Торренсу.

Для визуализации теней используются несколько методов (в зависимости от требований): статические (предвычисленные тени) и динамические - метод карт теней и тени в экранном пространстве (Screen Space Ambient Occlusion). Список объектов, отбрасывающих и принимающих тень, может быть задан в программе трехмерного моделирования.

### 3.3 Конвейер обработки изображений

Реализованная подсистема визуализации поддерживает возможность конфигурирования последовательной обработки изображений – набора последовательно применяемых к изображениям на графическом процессоре эффектов. Описание задается в виде графа фильтров, обрабатывающих изображения [2]. При этом входных и выходных изображений у фильтра может быть несколько. Для каждого изображения могут быть указаны различные параметры: абсолютные или относительные размеры, цветовое пространство, формат пикселей. В качестве входных изображений могут использоваться изображения с предыдущих кадров (временная фильтрация). На рисунке (Рис.3) представлен эффект глубины резкости.

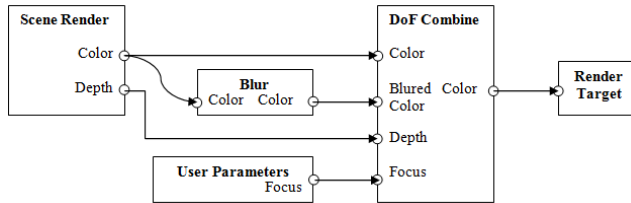


Рис. 3: Граф эффекта глубины резкости

Последовательная обработка текстур применяется как до визуализации (преобразование форматов, выделение актёра), так и после: цветовая коррекция, имитация расфокусировки камеры, туман, эффект смазывания при движении.

### 3.4 Воспроизведение HD-видео

Обработка видеоданных высокой четкости требует серьезных вычислительных ресурсов – необходима буферизация и декодирование большого объема данных (например, из YUV2 в RGB формат). Для проигрывания видеопотоков реализована асинхронная обработка потока данных (декодирование средствами DirectShow, передача в видеопамять).

Для каждого видеопотока выделяется отдельная нить исполнения и в базе данных размещаются: описание, команды управления, идентификатор очереди кадров. Идентификатор очереди кадров используется для запроса дескрипторов разделяемой видеопамети и информирования о состоянии кадров (их готовности). Таким образом, модули-источники видеоданных (с камер, диска) пишут кадры без дополнительных копирований через системную память.

В настоящий момент широко распространены источники чересстрочного сигнала – профессиональные и любительские видеокамеры. Для правильного отображения на устройствах с прогрессивной разверткой в разработанной системе используется метод устранения чересстрочности YADIF [7]. Метод адаптирован для исполнения на графическом процессоре и используется в варианте для формата YUV2.

Любой видеопоток может быть назначен в качестве «живого фона». При наличии в видеопотоке информации о глубине, может быть задано произвольное расположения актёра и элементов фона относительно друг друга. В обоих случаях применяется точная синхронизация кадров динамического фона со всей анимацией виртуальной сцены, поскольку без нее возникают неприемлемые визуальные дефекты (например, при движении виртуальной камеры).

### 3.5 Рисование на 3D объектах

Одним из поддерживаемых средств интерактивного взаимодействия является возможность рисовать поверх произвольного объекта сцены. Автоматически определяется какому объекту принадлежит выбранная пользователем точка

и какие текстурные координаты ей соответствуют. В итоге, создается дополнительная динамическая текстура, используемая как слой во время визуализации объекта.

Для определения выбранного в сцене объекта используется отдельный проход визуализации трехмерной сцены. Каждый объект рисуется со специальным шейдером, записывающим в цвет пикселя код: номер рисуемого объекта (либо группы треугольников) и текстурные координаты. Результат этого прохода считывается перед началом прохода визуализации следующего кадра. После этого информация передается в подсистему генерации динамического рисунка пользователя.

## 4. ВЫДЕЛЕНИЕ АКТЁРА

Для построения маски прозрачности по видео-изображению актёра был разработан оригинальный алгоритм, ориентированный на упрощенные условия съёмки. Для выделения актёра из фона необходимо для каждого пикселя определить какая его часть ( $\alpha$  – вес пикселя) должна попасть в результирующее изображение [5]. Предполагается, что актер снимается на одноцветном фоне (допустимы дефекты освещения, неровности, вкрапления) и цветовой тон фона отсутствует на актёре. Таким образом, цвет фона представляют собой некоторую область в цветовом пространстве YUV, размер которой в плоскости UV соответствует неоднородности цвета, а в направлении Y (яркости) – неоднородности освещения. Так как неоднородность освещения даёт более значимый вклад, область цветов фона получается вытянутой вдоль оси Y, поэтому для удобства расчётов эту область можно ограничить эллипсоидом, вытянутым по оси Y.

Для разделения всех трёх областей (фона, актёра и перехода актёр-фон) используются два эллипсоида, оси которых совпадают. Внутренний эллипсоид ограничивает цвета фона, внешний цвета актёра, пространство между ними – переход актёр-фон. Таким образом, в области цветов фона каждому пикселю будет поставлена в соответствие  $\alpha = 0$ , в области актёра  $\alpha = 1$ , а для цветов не принадлежащих двум этим множествам  $\alpha$  лежит в пределах от 0 до 1, в зависимости от того, к какой области цветов ближе находится цвет пикселя.

Пусть  $C_{и}$  – цвет исходного пикселя,  $C_{а}$  – цвет актёра,  $C_{ф}$  – цвет фона, то можно представить каждый пиксель входного изображения в виде:

$$C_{и} = C_{а} * \alpha + C_{ф} * (1 - \alpha)$$

Для нахождения  $\alpha$  находим расстояние L от цвета фона до цвета текущего пикселя (длина вектора в кубе YUV). Используем два параметра  $r_0$  и  $r_1$  (радиусы), где  $0 < r_0 < r_1$  и  $r < r_0$  задает область фона, а  $r > r_1$  – область актёра. Параметры задаем вручную либо автоматически так, чтобы для цветов, не принадлежащих ни актёру, ни фону, L лежало в пределах от  $r_0$  до  $r_1$ . В итоге вычисляем  $\alpha$ :

$$\alpha = \min\left(1, \max\left(0, \frac{L - r_0}{r_1 - r_0}\right)\right)$$

Тогда цвет актёра (без фона) вычисляется по формуле:

$$C_{р} = (C_{и} - C_{ф} * (1 - \alpha)) / \alpha$$

А цвет на этапе визуализации:

$$C_{вых} = (C_{и} - C_{ф} * (1 - \alpha)) + C_{сцена} * (1 - \alpha)$$

Для улучшения качества к  $\alpha$ -маске применяются фильтры обработки изображения (выделение границ, размытие), а также для цвета применяется цветокоррекция.

В совокупности эти методы, реализованные для графического процессора, позволяют в реальном времени совмещать

изображения реальных объектов с виртуальным без видимых дефектов в различных съёмочных условиях (рис. 4).



Рис. 4: Отображение актёра на произвольном фоне

## 5. ВЫВОД ИЗОБРАЖЕНИЯ

В разработанной системе реализована поддержка вывода результирующего видеопотока на внешние устройства, вещание в сеть, запись на диск, а также отображение на мониторы (выход видеокарты). Для обеспечения вертикальной синхронизации кадры буферизируются и переключаются в отдельных потоках исполнения. При этом между потоками передаются только идентификаторы на ресурсы графической видеокарты (используются средства видеодрайвера для разделения видеопамати между потоками). В качестве устройств для организации вещания поддерживаются платы SoftLab-NSK (семейство плат FD, [http://www.softlab-nsk.com/rus/forward/hardware\\_all.html](http://www.softlab-nsk.com/rus/forward/hardware_all.html)).

Различные форматы имеют разное разрешение и соотношение сторон пиксела (аспект). Растеризуемое изображение может иметь фиксированный размер и тогда к нему применяются корректирующие методы (вписывание, обрезка, масштабирование). Так, при необходимости, к изображению могут быть добавлены чёрные поля. Например, при несовпадении разрешений (либо аспектов) видеовыхода (монитор, плата и т.д.) с окном превью, изображение в окне «превью» автоматически масштабируется и обрамляется чёрными полями.

## 6. ЗАКЛЮЧЕНИЕ

В ходе разработки системы выделены ключевые особенности презентационных систем реального времени. Разработаны и реализованы решения для соответствующих вопросов, и, как результат, разработано программно-аппаратное решение для создания презентаций различной сложности. Применяемая архитектура системы позволяет расширять и дополнять ее новыми функциональными возможностями, необходимыми широкому кругу пользователей.

Сферой применения комплекса являются: всевозможные презентации для выставок, лекционных залов, ситуационных центров, музеев, обучение персонала управлению сложными технологическими процессами и техническими системами, дистанционное обучение и телеконференции.

## 7. БИБЛИОГРАФИЯ

- [1] Ванданов В.Г. и др. Программно-аппаратный комплекс 3d презентаций на основе виртуальной студии и виртуального окружения // Труды конференции «Трёхмерная визуализация научной, технической и социальной реальности. Кластерные технологии моделирования». 2009. Том 1. С. 73-77.
- [2] Городилов М.А. и др. Управление конфигурациями конвейера обработки изображений и спецэффектов систем визуализации реального времени // Сборник трудов «Технологии Microsoft в теории и практике программирования». 2011. – С. 85-87.
- [3] Долговесов Б.С. и др. Система обучения и презентаций с использованием технологии «погружения» лектора в интерактивную виртуальную среду // Сборник трудов «Online EDUCA 2007, Moscow». 2007.
- [4] Долговесов Б.С. и др. Объектно-ориентированная база данных в интерактивных системах 3D визуализации // Вестник НГУ. Серия: Физика (том 6, выпуск 3) . 2011. – С.59-63
- [5] Ковальков М.А. и др. Разработка и реализация алгоритмов рирпроекции на базе современного графического акселератора // Труды конференции Графикон-2006. 2006. – С. 360-362
- [6] Akenine-Moeller T. et. Real-time rendering, 3ed, 2008.
- [7] Balakhnin A., Port of Yet Another DeInterlacing Filter, // AviSynth's Documentation. 2007 (<http://avisynth.org.ru/>)

## Integrated solution for interactive multimedia presentations

### Abstract

This paper presents features of the modern hardware/software integrated solution for interactive presentations. Developed presentation system provides wide opportunities for training and demonstration of various multimedia data. Also the paper describes implementation details and presents useful techniques for providing an interactive integration both images of a lecturer and media (video, presentations, text) into virtual scene. Integrated multimedia content improves the quality of presentations.

**Keywords:** virtual presenter, multimedia, real-time rendering.

### About the author

Boris S. Mazurok is a scientific researcher of Synthesizing Visualization Systems Laboratory at Institute of Automation and Electrometry SB RAS. His contact email is [boris@albatros.iae.nsk.su](mailto:boris@albatros.iae.nsk.su)

Boris S. Dolgovesov (Ph.D.) is a head of Synthesizing Visualization Systems Laboratory at Institute of Automation and Electrometry SB RAS. His contact email is [bsd@iae.nsk.su](mailto:bsd@iae.nsk.su)

Evgeny I. Korostelev is a Ph.D. student at Institute of Automation and Electrometry SB RAS. His contact email is [kore3d@gmail.com](mailto:kore3d@gmail.com)

Timur N. Artikov is a student of Faculty of Information Technologies at Novosibirsk State University, an engineer-programmer of Synthesizing Visualization Systems Laboratory at

Institute of Automation and Electrometry SB RAS. His contact email is [t.artikov@gmail.com](mailto:t.artikov@gmail.com)

Artur N. Artikov is a student of Faculty of Information Technologies at Novosibirsk State University, an engineer-programmer of Synthesizing Visualization Systems Laboratory at Institute of Automation and Electrometry SB RAS.. His contact email is [a\\_artikov@mail.ru](mailto:a_artikov@mail.ru)

# Интерпретационный метод визуализации графовых алгоритмов

Гордеев Дмитрий Станиславович, gds@iis.nsk.su, ИСИ СО РАН

## Аннотация

В статье описывается подход к визуализации графовых алгоритмов, позволяющий задание алгоритма в качестве параметра. Основными преимуществами данного подхода являются возможность задания алгоритма в текстовой форме в качестве параметра и гибко настраивать результирующую визуализацию. Визуализация алгоритмов осуществляется с помощью множества настраиваемых эффектов. В качестве входных графов рассматривается класс иерархических графов. Использование иерархических графов позволяет облегчить визуализацию дополнительных структур данных, так как графовое представление структуры данных можно включить в иерархию графовой модели. Описываемый подход может использоваться как для создания образовательных систем, так и для исследования графовых алгоритмов.

**Ключевые слова:** граф, иерархический граф, алгоритм, визуализация, визуальный эффект.

## 1. ВВЕДЕНИЕ

В настоящее время методы визуализации графов имеют множество применений. Графы и графовые модели используются для задач, оперирующих информацией, представимой в виде объектов и связей между ними. Например, синтаксические деревья в трансляторах, раскраска графов в конструировании электрических схем, поиск кратчайшего пути в компьютерных играх или геоинформационных системах и т.д. [1]. Целью рисования графов является получение изображений графов, простых для рассмотрения и понимания. Рисование графов используется во многих приложениях для проектирования и анализа коммуникационных сетей, связанных документов, а также статических и динамических структур программ. Часто подобные системы подвергаются внутренним или внешним воздействиям, например, изменяющим связи между объектами. В случае, если такие процессы можно формализовать и представить в виде некоторого алгоритма возникает необходимость построить графическое представление процесса. Существуют методы позволяющие представлять такие алгоритмы, как в динамической форме, так и в статической форме. К статическим формам можно отнести блок-схемы. К динамическим формам можно отнести методы, генерирующие последовательности рисунков, изображающих промежуточные состояния графовых моделей. На приведенном ниже рисунке представлен пример динамической системы визуализации алгоритмов [4,6].

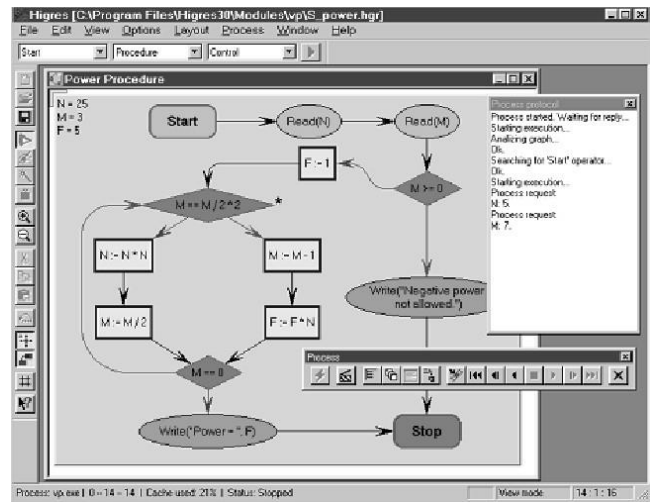


Рис. 1: Выполнение алгоритма, реализованного в виде внешнего модуля к системе Higes.

Существование подобных методов позволяет изучать графовые алгоритмы как в целом, так и в приложении к связанным системам. Большинство работ по визуализации алгоритмов сконцентрировано на построении примеров визуализации [5]. Основной характерной особенностью таких работ можно назвать узкую специализацию, в том смысле, что для каждого нового алгоритма требуется создавать новый визуализатор.

## 2. МЕТОДЫ ВИЗУАЛИЗАЦИИ

В данной работе описывается метод построения визуализаций основанный на визуальных эффектах, генерируемых по входному алгоритму. Обычно элементам графа сопоставляются некоторые графические примитивы, что дает возможность построить его изображение. Например, вершины отображаются окружностями, а дуги прямыми, ломаными или гладкими кривыми. Среди применяющихся методов визуализации алгоритмов можно выделить два метода: метод интересующих событий и метод, основанный на данных [2]. Первый метод основан на выделении событий, которые происходят во время исполнения алгоритма. Например, сравнение величин атрибутов некоторых элементов графа, добавление вершины или удаление дуги. Суть этого метода в том, чтобы для каждого такого события реализовать визуальный эффект. Второй метод, основан на изменении данных. Во время работы изменяется состояние памяти, например, значения переменных. Далее эти изменения некоторым образом визуализируются. В простейшем случае для этого используется отображение значений переменных в таблице. Такой метод применяется в отладчиках интегрированных систем разработки программного обеспечения.

У существующих визуализаторов алгоритмов есть ряд недостатков. Если есть необходимость построить визуализацию алгоритма, текстовое представление которого

отличается от текста оригинального алгоритма, потребуется заново изготовить визуализатор. Также визуализаторы обычно не отображают соответствие между инструкциями алгоритма и генерируемыми визуальными эффектами. Часто визуализаторы не позволяют перенастроить визуальные эффекты, соответствующие событиям. К недостаткам также можно отнести излишнюю перегруженность исходного текста алгоритма декларативными инструкциями. На рисунке 2 представлен кадр, полученный при визуализации алгоритма в системе Leonardo [3, 7]. Как видно на рисунке, в системе Leonardo используются директивы в специальном формате, `/** Not VisualUpdate **/`. Такие декларативные конструкции используются для группировки визуальных событий или же для непосредственного запуска визуальных эффектов.

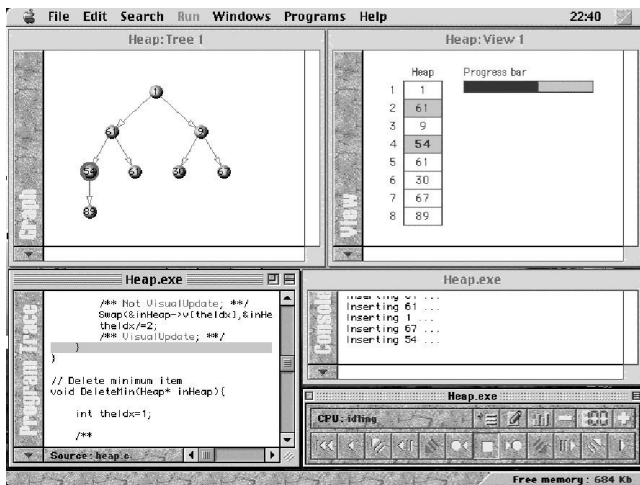


Рис. 2: Пример визуализации операций над binary heaps в системе Leonardo, A C Programming Environment for Reversible Execution and Software Visualization.

### 3. МОДЕЛЬ ИНТЕРАКТИВНОЙ ВИЗУАЛИЗАЦИИ

Для решения этих задач предложена модель визуализации графовых алгоритмов основанная на динамическом подходе. Суть данного подхода заключается в том, что заданный алгоритм формулируется на языке программирования, допускающем использование графовых структур данных, а также с возможностью исполнить программу, полученную из текста алгоритма после предварительной обработки. Во время исполнения полученной программы генерируется информация, которая используется для визуализации оригинального алгоритма. Примером конструкции, использующей графовые структуры данных, является операция добавления дуги или изменение атрибута вершины графа. В приведенном ниже примере представлен алгоритм обхода в ширину для неориентированного графа. В данном случае используются конструкции `Get`(идентификатор вершины, имя атрибута) и `Set`(идентификатор вершины, имя атрибута, значение) для чтения и изменения значения атрибута вершины соответственно. Для построения визуализации алгоритма обхода в ширину вершинам входного графа добавляется атрибут `state`, по значению которого можно определить была ли посещена вершина в процессе обхода графа или нет.

```
VertexQueue.Enqueue(1, Graph.Vertices[0]);
while (WhileCondition(2, VertexQueue.Count > 0))
{
    Vertex v = VertexQueue.Dequeue(4);
    Set(5, v, "state", "visited");
    foreach(Vertex n in ForeachCollection(6, v.Incidents))
    {
        string c = Get(8, n, "state");
        if(IfCondition(9, c != "visited"))
        {
            VertexQueue.Enqueue(11, st);
        }
    }
}
```

Для решения задачи визуального выделения текущей исполняемой инструкции в кадре, сгенерированном во время исполнения данной инструкции, используется следующий подход. Оригинальный текст алгоритма преобразуется так, чтобы добавить в каждую значимую строку текста номер этой строки. В представленном выше примере добавленные конструкции выделены жирным шрифтом. Во время исполнения преобразованной программы каждая инструкция алгоритма генерирует информацию, описывающую номер исполняемой строки, идентификатор обрабатываемого элемента графа, название атрибута графового элемента, предыдущее значение атрибута, новое значение атрибута, а также временную метку.

Данная информация о каждой исполненной инструкции позволяет получить историю исполняемых операций, содержащий подробную информацию о состоянии графовой модели во время исполнения алгоритма. Далее полученная история операций и исходный граф используется для построения визуализации алгоритма. Каждой записи истории соответствует некоторый графический эффект. Простейший пример такого графического эффекта это цветное выделение исполняемой строки алгоритма.

### 4. СИСТЕМА ВИЗУАЛИЗАЦИИ АЛГОРИТМОВ

С использованием предложенной модели реализован прототип системы визуализации алгоритмов. Система визуализации графовых алгоритмов содержит несколько компонентов. Это модуль исполнения алгоритма, графовый редактор и визуализатор графовых алгоритмов. Назначением модуля исполнения является запуск программы, полученной из текста алгоритма-параметра и порождение истории исполненных операций. Таким образом, следует заметить, что исполнение алгоритма-параметра отделено от визуализации. Это позволяет исполнить алгоритм однажды, и затем строить и уточнять визуализацию без повторных запусков. Это может быть полезно при визуализации вычислительно-емких алгоритмов, когда повторное исполнение алгоритма требует существенного времени.

Чтобы обеспечить корректное функционирование модуля исполнения, требуется выполнение важных условий. Во-первых, для реализации данного модуля подходит любой существующий компилятор или интерпретатор. Соответственно, входной алгоритм должен быть сформулирован с использованием языка выбранного компилятора или интерпретатора. Этот пункт не делает существенных ограничения на язык описания входных

алгоритмов, так как многие языки программирования допускают использование графовых конструкций. Во-вторых, запрещается использование специальных инструкций более одного раза в одной строке. Нарушение этого условия приводит к генерации нескольких визуальных эффектов при том, что выделяться цветом будет только одна строка из текста алгоритма.

Вторым основным компонентом является визуализатор. Он получает на вход текст алгоритма, история операций, граф и дополнительные графические настройки. Каждая запись истории операций содержит информацию об исполняемой инструкции алгоритма. Специальные инструкции являются функциями: Set(...), Get(...), IfCondition(...), WhileCondition(...) and ForeachCollection(...). Первый аргумент этих функций является номером строки в тексте алгоритма. Функции IfCondition(...) and WhileCondition(...) не производят никаких изменений в состоянии графовой модели, но позволяет генерировать эффект выделения строки текста с условным оператором if и оператором цикла while. ForeachCollection(...) инструкция используется для генерации визуального выделения множества вершин перед тем, когда множество будет обрабатываться в цикле foreach. Чтобы вставить эти функции в подходящие места текста алгоритма достаточно использовать контекстную замену.

Каждая запись истории может содержать информацию о текущем значении атрибута для некоторого элемента графа, будь то вершина, ребро или порт. Для вершины, инцидентной некоторой дуге, порт это точка, где дуга входит в вершину. Выделение порта в отдельную сущность обусловлено необходимостью изображать точки входа и выхода дуг из вершин. Часто, если из вершины выходит несколько дуг, необходимо построить изображение, в котором точки выхода необходимо различать. Выделенная сущность порта упрощает вычисление координат графических примитивов, представляющих дуги графа. При этом встает задача определения связи между информацией записи истории и соответствующим визуальным эффектом. В тексте алгоритма допускается использование произвольных имен атрибутов. В этом случае требуется вмешательство пользователя, чтобы установить интерпретацию именами атрибутов с помощью множества доступных визуальными эффектами.

На представленном ниже рисунке представлен пример визуализации алгоритма поиска в глубину. Рисунок представляет собой один из кадров, сгенерированный в процессе визуализации алгоритма. В левой части рисунка отображен текст алгоритма. Наличие атрибута цвета у вершины графа означает факт, что вершина уже была посещена в процессе обхода. Строка алгоритма может быть в одном из трех состояний: тонкий темный шрифт, светлый тонкий шрифт и светлый толстый шрифт. Первое состояние означает, что соответствующая строка алгоритма была исполнена хотя бы один раз. Второе состояние означает, что текущий визуальный эффект инициирован данной инструкцией. Последнее состояние означает, что инструкция еще не была исполнена. В правой части представлен иерархический помеченный граф с атрибутами. Наличие или отсутствие атрибута приводит к реализации соответствующих визуальных эффектов. В этом примере посещенные вершины получают установленный атрибут цвета. Такому

значению атрибута соответствует увеличение толщины линии, применяемой для рисования окружности. Вершины с тонким контуром еще не были посещены.

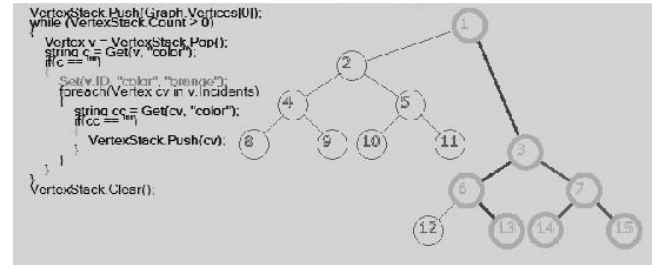


Рис. 3: Пример визуализации алгоритма поиска в глубину, один из промежуточных кадров.

При визуализации графовых алгоритмов существует возможность применить методы, повышающие наглядность визуализации. Если для некоторой записи истории определен контекст визуализации эффекта, то этот контекст используется для повышения наглядности работы алгоритма. Например, визуализации алгоритма обхода в глубину можно применять плавную визуализацию вдоль дуги соединяющей предыдущую и текущую вершины. В таком случае контекстом визуализации для текущей вершины будет предыдущая вершина. В случае, если предыдущая вершина не является смежной с текущей вершиной, для повышения наглядности визуализации применяется плавная визуализация вдоль дуг, принадлежащих кратчайшему пути, соединяющему текущую и предыдущую вершины. На рисунке 3 представлен пример использования визуализации с использованием контекста. В данном примере в процессе обхода, при продвижении от предыдущей вершины к текущей вершине изменяется толщина линий, применяемых для визуализации инцидентных дуг. Причем утолщенная светлая линия применяется для рисования дуг, принадлежащих пути от текущей вершины до корня дерева. Утолщенная темная линия применяется для рисования дуг, инцидентных уже посещенным вершинам.

Для повышения наглядности визуализации полезно отображать дополнительные структуры данных. например, при визуализации алгоритма обхода графа в ширину полезно отображать содержимое очереди. Содержимое очереди легко представляется в виде графа. Это, наряду с использованием иерархических графов, позволяет отобразить содержимое очереди без дополнительных усилий. Так дерево иерархии графа будет содержать корень и два листа. Первый лист соответствует фрагменту входного графа, а второй лист соответствует фрагменту графа, изображающего очередь.

## 5. ЗАКЛЮЧЕНИЕ

В данной статье описывается модель визуализации алгоритмов на графах, обеспечивающая построение визуализаций алгоритмов за счет использования алгоритма в качестве параметра и системы визуальных эффектов. Также описывается метод для повышения наглядности визуализации алгоритмов на графах, позволяющий использовать дополнительную информацию, генерируемую во время исполнения алгоритма. Разработана система

визуализации алгоритмов, обеспечивающая апробацию на практике предлагаемого подхода визуализации алгоритмов на графах. Результатом работы системы является последовательность изображений, соответствующих промежуточным состояниям графовой модели во время работы алгоритма. Интерактивность визуализации достигается с помощью возможности изменять текст алгоритма и заново генерировать визуализацию, а также с помощью возможности гибко настроить интерпретацию имен атрибутов, используемыми в тексте, с помощью множества визуальных эффектов.

## 6. REFERENCES

- [1] Касьянов В. Н., Евстигнеев В. А. *Графы в программировании: обработка, визуализация и применение*. - СПб. БНВ-Петербург, 2003. - 1104 с.. ул. ISBN 5-94157-184-4.
- [2] Demetrescu C., Finocchi I., Stasko J. T. *Specifying Algorithm Visualizations: Interesting Events or State Mapping?* // *In Proc. of Dagstuhl Seminar on Software Visualization - Lect. Notes in Comput. Sci.* - 2001. - P. 16-30.
- [3] Demetrescu C., Finocchi I. *A general-purpose logic-based visualization framework*, *Proceedings of the 7th International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media (WSCG'99)*, pp. 55-62, Plzen, Czech Republic, February 1999.
- [4] Lisitsyn I.A., Kasyanov V.N. *Higres - visualization system for clustered graphs and graph algorithms* // *Proc. of Graph Drawing 99*. - Berlin a.o.: Springer Verlag, 1999. - P. 82-89. - (*Lect. Notes in Comput. Sci.*; Vol. 1731).
- [5] Shaffer C. A., Cooper M. L., Alon A. J. D., Monika Akbar, Michael Stewart, Sean Ponce, S. H. Edwards. *Algorithm Visualization: The State of the Field* // *ACM Transactions on Computing Education TOCE*. -- ACM, 2010. -- v. 10, is. 3, pp. 1-22.
- [6] Higres. <http://pcosrv.iis.nsk.su/higres/>.
- [7] Leonardo. <http://www.dis.uniroma1.it/~demetres/Leonardo/>.



# Графический моделирующий комплекс для автономного необитаемого подводного аппарата

Михаил Морозов, Институт автоматизации и процессов управления ДВО РАН, morozov@iacp.dvo.ru

Игорь Туфанов, Институт проблем морских технологий ДВО РАН, ituphanov@gmail.com

## Аннотация

В работе рассматривается опыт создания и использования графического моделирующего комплекса автономного необитаемого подводного аппарата. Комплекс предназначен для отладки миссий аппарата в виртуальной трехмерной среде и позволяет выполнить часть необходимых испытаний в лабораторных условиях, что существенно снижает время и стоимость экспедиционных работ. Отладка миссии производится на основе формирования целевых реалистичных изображений окружающей среды для решения отдельных задач управления подводного аппарата.

**Ключевые слова:** Автономный необитаемый подводный аппарат, 3D моделирование, OpenGL, распознавание образов, машинное зрение.

## 1. ВВЕДЕНИЕ

При решении многих задач, связанных с маневрированием автономных необитаемых подводных аппаратов /АНПА/ вблизи подводных сооружений, требуется обработка видеоизображений для формирования надлежащего управления. В процессе подготовки миссии подводного аппарата обычно предварительно производится её моделирование с целью проверки и выявления ошибок, которые могут возникнуть в море. Этот процесс также называется верификацией миссии. Верификация особенно важна в миссиях с применением видео, поскольку такие миссии сложнее традиционных.

При использовании видео процесс моделирования усложняется: помимо движения аппарата и параметров водной среды, необходимо также иметь на выходе модели изображение, получаемое камерой. То есть, необходима модель трёхмерной среды, в которой аппарат выполняет перемещение, равно как и модель камеры, как одного из датчиков.

Настоящая работа посвящена разработке такой трёхмерной модели и схеме её использования для проверки миссии, являющейся переменной частью системы управления /СУ/ АНПА.

Существует множество систем моделирования для робототехники. В том числе и такие системы, в которых модельное изображение входит в цикл управления. Например, в работе [7] описан процесс моделирования, охватывающий не только программную, но и аппаратную части автономного летательного аппарата. Модельное изображение не передаётся модулю распознавания напрямую, а проецируется на экран и фиксируется камерой. Симулятор ÜberSim [6] используется в Carnegie Mellon University для моделирования действий группы роботов при игре в футбол. В области подводной робототехники существует модель, разработанная в рамках проекта CO<sup>3</sup>AUVs [4]. Она принимает на вход команды управления подводным аппаратом, а возвращает данные датчиков. В данной работе рассматривается опыт создания и использования

графического моделирующего комплекса автономного необитаемого подводного аппарата, предназначенного для АНПА МАРК [1].

## 2. ЗАДАЧИ АНПА, ТРЕБУЮЩИЕ УПРАВЛЕНИЯ, ОСНОВАННОГО НА ВИДЕО

К задачам, в которых требуется включение видео в цикл управления, относится задача следования вдоль протяжённого объекта [3]. Она может возникать в случае, если необходимо произвести инспекцию такого объекта (кабеля, трубы). Предполагается, что АНПА сначала совершает поисковый манёвр и начинает следовать вдоль искомого объекта, когда тот попадает в кадр. Предметом исследований здесь являются как алгоритмы движения, так и алгоритмы распознавания протяжённого объекта на кадре.

Другая задача, требующая обработки видео – это стыковка АНПА с донной станцией [1]. Предполагается, что для наведения на станцию в дальней зоне используется гидроакустический маяк, а в ближней зоне и во время стыковки АНПА использует изображение с камеры, чтобы распознать специальный маркер и выполнить точное позиционирование.

Ежегодно проходят различные соревнования по подводной робототехнике, в которых участникам необходимо создать АНПА, поведение которого основано на использовании видеoinформации. При этом перед АНПА участникам ставятся задачи, решения которых связаны с операциями вблизи визуальных объектов. Так, например, в соревнованиях RoboSub [9], проходящие под эгидой AUVSI Foundation, в бассейне в толще воды расположены различные объекты: буи, плоские дорожки, конструкции из труб ПВХ. Размеры и цвета объектов, а также их примерное расположение становится известным участникам приблизительно за полгода до соревнований. Аппарат участников должен перемещаться по бассейну, следуя оранжевым дорожкам, дотрагиваться корпусом до буёв определённого цвета, проходить сквозь рамки и т.д. Другие соревнования, Singapore AUV Challenge [10], проходят под эгидой IEEE. АНПА участников должны следовать по изображённым на дне бассейна дорожкам и попутно совершать некоторые дополнительные перемещения. Существуют также европейские соревнования, Student AUV Challenge – Europe [11]. Из трёх перечисленных соревнований последние два являются студенческими, а к участию в RoboSub допускаются также аспиранты и научные сотрудники. Участие в подобных соревнованиях преследует, помимо прочих, и технические цели: отладить СУ своего АНПА и сравнить её с аналогами. Одним из центральных элементов СУ для подобных соревнований является обработка видео и включение его в цикл управления.

## 3. СИСТЕМА УПРАВЛЕНИЯ

В настоящем разделе описывается СУ, которая применяется на АНПА «МАРК» [1] и была разработана для АНПА «Юниор»

для участия в соревнованиях RoboSub в 2012 и 2013 году. В настоящее время продолжается разработка и поддержка единой системы для двух аппаратов. Тем не менее, существуют части системы, специфичные для каждого аппарата и в настоящей работе рассматривается лишь та её часть, которая используется на АНПА «Юниор», поскольку именно он оснащён видеокерами.

СУ реализована в виде набора исполняемых модулей. Для передачи данных используется механизм IPC, основанный на сетевых сокетах. У каждого модуля имеется набор входящих и/или исходящих сообщений. Такая организация СУ позволяет произвольным образом распределять модули по компьютерам, если на борту АНПА компьютеров имеется несколько (это зависит также и от того, к каким компьютерам подключены бортовые приборы). Так, например, АНПА «Юниор» имеет два бортовых компьютера, на одном из которых запускаются программы обработки изображений и гидроакустических сигналов, а на другом – модули СУ, отвечающие за управление движением аппарата и исполнение программы-задания. На борту АНПА модули работают под управлением ОС Linux, но могут быть перекомпилированы и под другую ОС (например, Windows) в случае необходимости.

На рис 1 изображена часть схемы СУ. В диаграмму не вошли некоторые драйверы бортовых устройств и пульт оператора.

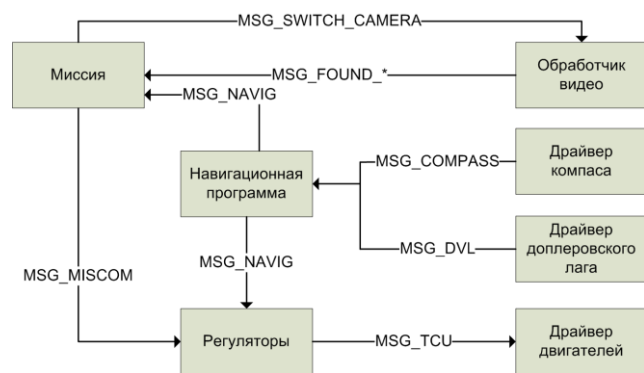


Рис 1: Основные модули СУ и сообщения, передаваемые между ними. Модули, находящиеся в правой колонке осуществляют обмен данными с бортовыми устройствами.

Опишем кратко некоторые модули и передаваемые между ними сообщения, поскольку это важно для дальнейшего изложения:

- **Миссия.** Основная программа, в которой действия необходимые для выполнения задачи разбиваются на простые команды движения. Эти команды передаются регуляторам в сообщениях MSG\_MISCOM.
- **Навигационная программа.** Собирает данные с навигационно-пилотажных датчиков в единый пакет (MSG\_NAVIG), на который подписаны регуляторы и миссия.
- **Регуляторы.** Принимают целевые параметры движения (MSG\_MISCOM) от миссии и текущие параметры (MSG\_NAVIG) от навигационной программы. На основе этого вычисляются необходимые параметры движения: тяги и угловые моменты.
- **Обработчик видео.** Модуль получает изображения от драйвера камеры (используется интерфейс v4l), осуществляет их предварительную обработку (цветокоррекция) и выполняет алгоритмы

распознавания. На выходе алгоритмов распознавания формируется описание найденных в кадре объектов: их цвета и геометрические параметры в координатах кадра. Сообщения с этим описанием (MSG\_FOUND\_\*) передаются миссии, которая использует их для принятия решений. Миссия может переключать камеры (менять активную камеру) и включать/выключать распознавание конкретных объектов, используя сообщение MSG\_SWITCH\_CAMERA.

- **Драйверы бортовых устройств.**

Таким образом, цикл управления, включающий обработку изображений, выглядит так: Миссия – Регуляторы – Драйвер двигателей – Двигатели – Среда – Камера – Обработчик видео – Миссия.

#### 4. ОБРАБОТКА ИЗОБРАЖЕНИЙ

Визуальные объекты, которые необходимо найти на изображениях, созданы искусственно и характеризуются резкими границами и яркими цветами. Однако при разработке алгоритмов поиска объектов на изображении информация о цвете используется лишь в непосредственной близости от объектов, поскольку под водой цвета искажаются.

Поиск объектов осуществляется с применением методов выделения границ, аппроксимации контуров полигонами, бинаризации по яркости или границам Н-канала в цветовом пространстве HSV. Для поиска прямолинейных объектов используется алгоритм на основе преобразования Хафа, учитывающий направления градиентов на изображении. При реализации широко использовалась библиотека OpenCV [5].

Примеры объектов и результатов их поиска на изображениях приведены на рис 2. Для поиска оранжевой дорожки (вверху) использовалась бинаризация по цвету в Н-канале. Для поиска шара (внизу) использовался алгоритм на основе преобразования Хафа.

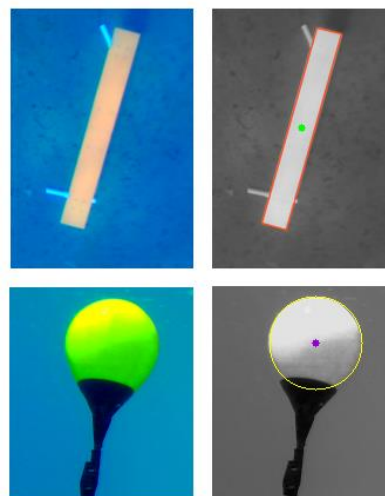


Рис 2: Примеры изображений искусственных объектов (слева) и результат их поиска (справа).

#### 5. МОДЕЛЬ СРЕДЫ

Для отладки задач управления на основе видео разработан графический комплекс моделирования внешней среды для АНПА. Комплекс позволяет моделировать получение сенсорной информации о внешней среде посредством фото-

видео камер и гидролокаторов. Комплекс дает возможность имитировать движение АНПА в водной среде и может служить визуальным инструментом для отладки различных алгоритмов выполнения миссий.

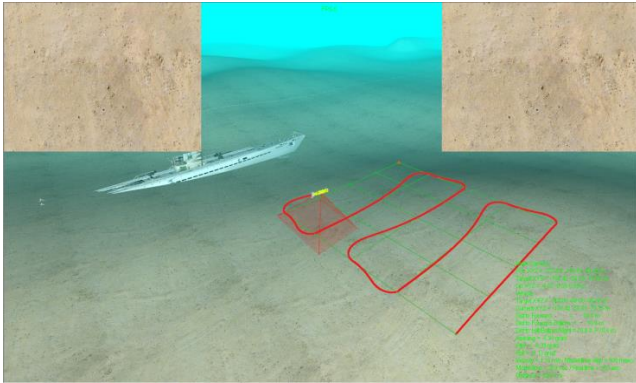


Рис 3: Моделирование движения АНПА в подводной среде. В отдельных окнах показаны изображения с бортовых камер, работающих в стерео-режиме. Камеры направлены на дно, поле зрения камер показано красной полупрозрачной пирамидой.

#### 4.1 Состав моделирующего комплекса среды

Моделирующий комплекс подводной обстановки сочетает в себе:

- 1.) Данные о подводной обстановке (рельеф, текстуры, статические и динамические объекты).
- 2.) Моделирование движения АНПА с учетом динамики реального аппарата.
- 3.) Средства отображения обстановки на экране.
- 4.) Возможность моделировать работу бортовых видеокамер и гидролокаторов АНПА в режиме реального времени.

#### 4.2 Данные о подводной обстановке

Рельеф и другие объекты задаются полигональными сетками с нанесенными текстурами, т.е. стандартными средствами представления сцен в трехмерной графике. На рельефе могут располагаться статические объекты, задающие различные детали, а над ним в водной среде движутся динамические анимированные объекты (рыбы). Сцена, описывающая подводную обстановку, загружается из файлов формата 3DS и таким образом может быть подготовлена в любом 3D-редакторе.

#### 4.3 Моделирование движения АНПА

Комплекс моделирования среды работает в двух основных режимах: автономном и в режиме внешнего управления. В первом случае расчет траектории движения АНПА производится самостоятельно, во втором случае траектория движения задается внешней программой. С внешней управляющей программой налажен двусторонний обмен посредством механизма разделения памяти (Shared Memory). От внешней программы МК среды получает координаты АНПА, обратно выдает изображения с бортовых камер.

#### 4.4 Рендеринг сцены

Визуализация сцены происходит посредством OpenGL с использованием аппаратного ускорения за счет хранения геометрии и текстур в видеопамати. Кроме этого используются методика разделения крупных объектов сцены на части и оптимизации за счет рендеринга только видимых частей. Достигнутая скорость рендеринга позволяет

моделировать движение и работу сенсоров АНПА в режиме реального времени. А в некоторых случаях (зависит от сложности сцены, конфигурации сенсоров и видеокарты), в режиме ускоренного моделирования миссии, т.е. с опережением реального времени.

#### 4.5 Моделирование сенсоров

В настоящее время таких сенсоров три: бортовая видео-фото камера, гидролокатор бокового обзора и однолучевой эхолот. Бортовые камеры моделируются стандартными средствами компьютерной графики (OpenGL). Каждая камера неподвижна относительно аппарата. Положение камеры задается точкой закрепления в локальной системе координат и вектором направления наблюдения. Также задается разрешение изображения, поле зрения и цветность RGB/Grayscale. Есть возможность имитировать на изображении размытие удаленных объектов за счет включения эффекта тумана. В настоящий момент комплекс позволяет моделировать работу 3х видеокамер, но при необходимости их количество может быть увеличено. Кадры с видеокамер могут формироваться или по запросу внешней управляющей программы или непрерывно с заданной частотой. Формирование гидролокационной картины происходит методом, описанным в работе [12]. Имитация работы гидролокатора бокового обзора была использована для разработки алгоритмов прослеживания протяженных объектов в работе [13].

### 6. МОДЕЛЬ В ЦЕЛОМ

Основной целью моделирования является верификация миссии. Возможна попутная отладка других модулей. При этом важно, чтобы исходный код модулей, подвергающихся проверке, не изменялся и был одним и тем же и в режиме моделирования и в реальном режиме работы.

При моделировании реальная физическая среда подменяется исполняемым модулем, который и является моделью. Модель перехватывает IPC-сообщения системы управления и подменяет сообщения от некоторых её модулей.

Трёхмерная модель среды является отдельной программой и работает под управлением ОС Windows. Передача данных между двумя частями модели осуществляется через общую память, являющуюся частью динамически присоединяемой библиотеки (DLL). Таким образом, процесс моделирования либо целиком осуществляется в ОС Windows, либо модель запускается на одном компьютере, а СУ – на другом (например, на бортовом компьютере АНПА), уже под управлением ОС Linux.

Тем не менее, моделирование с использованием двух компьютеров может быть затруднено, если предполагается передача несжатых изображений по сети. В этом случае изображения приходится предварительно сжимать таким образом, чтобы это не сказалось на качестве распознавания. В случае запуска всех модулей на одном компьютере подобных проблем не возникает.

Схема подключения модели изображена на рис 4. Приходится создать новое сообщение, MSG\_VIDEO\_FRAME, которое модель формирует для передачи обработчику видео. Это сообщение содержит изображение для обработки. Модель также формирует навигационный пакет, и принимает параметры движения аппарата от регуляторов. Основной цикл модели выглядит следующим образом:

1. Принять сообщение о параметрах движения.
2. Вычислить новое положение аппарата.
3. Передать новое положение аппарата модели среды.

4. Получить от модели среды изображения с камер.
5. Передать изображения с камер обработчику

Начальные параметры движения аппарата передаются модели через параметры командной строки.

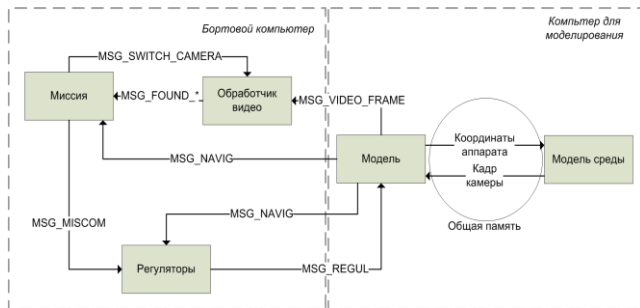


Рис 4: Схема подключения модели с использованием двух компьютеров. На бортовом компьютере работают основные модули СУ. На втором компьютере работают обе части модели, обменивающиеся данными через общую память.

## 7. РЕЗУЛЬТАТЫ

Далее будут описаны параметры, с которыми модель использовалась для подготовки к соревнованиям RoboSub 2012 и RoboSub 2013.

Запуск модели производился на одном компьютере под управлением ОС Windows. Модули СУ могут быть скомпилированы для этой ОС. То же самое касается и библиотек третьих сторон (IPC [8] используется для передачи сообщений и OpenCV [5] используется для обработки изображений).

Основной целью моделирования была верификация миссии, в то время как регуляторы являлись предметом отдельного направления работы. Поэтому при моделировании регуляторы не использовались, и модель напрямую принимала сообщение MSG\_MISCOM. Такая схема подключения представлена на рис 5.

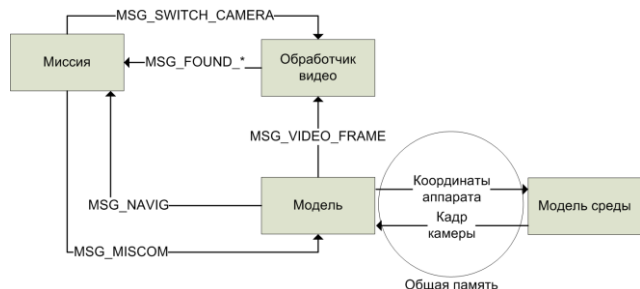


Рис 5: Схема моделирования миссии без использования регуляторов.

Опишем каждый шаг основного цикла моделирования более подробно:

1. Принять сообщение MSG\_MISCOM, если таковое имеется в очереди сообщений. Оно задаёт целевые параметры движения: координаты, глубину и курс.
2. Вычислить новое положение аппарата. Используется кинематическая модель, в которой крен и дифферент полагаются равными нулю. Линейные и угловые скорости аппарата ограничены. В случае если расстояние до целевой точки находится в пределах некоторой константы, аппарат совершает поворот в

сторону целевой точки и движется вперед. В противном случае аппарат смещается в сторону целевой точки и стабилизирует курс к целевому (движительная система аппарата позволяет это сделать).

3. Передать новое положение аппарата модели среды. На этом этапе необходимо выполнить перевод из системы координат, используемой в СУ ( $x, y$ , глубина, азимут курса), к координатам, используемым в среде ( $x, y, z$ , курсовой угол).
4. Получить от модели среды изображения с камер. Используются две камеры: передняя и нижняя. Вертикальный угол обзора каждой камеры составляет  $60^\circ$ . Размер изображения  $320 \times 240$ .
5. Передать изображения с камер обработчику. Изображения передаются на вход распознавателя в несжатом виде.

Использование модели значительно облегчило программирование миссии.

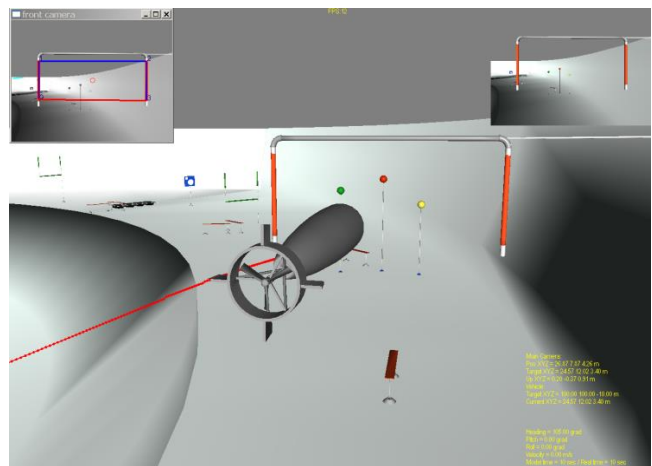


Рис 6: Часть окна модели среды. В модель загружена сцена, подготовленная для соревнований RoboSub. В левом верхнем углу находится отладочное окно модуля обработки видео. В нем отмечены специальным маркером распознанные «ворота».

## 8. ВЫВОДЫ

К факторам, которые не учитывает настоящая модель, но которые могут быть учтены, относятся:

- Динамика аппарата.
- Тяговые характеристики движителей (например, как распределена сила при вращении винта в обе стороны).
- Ошибки датчиков.
- Искажения изображений, возникающие из-за специфики водной среды.
- Искажения, вносимые оптикой видеокамер.

Таким образом, верификация миссии на модели означает лишь, что логика миссии верна. Программное обеспечение аппарата при этом всё еще может содержать ошибки или не учитывать немодельных факторов.

Тем не менее, созданная модель позволяет верифицировать алгоритм работы миссии и представляется полезной для предварительной отладки реальных морских работ.

## 9. ЛИТЕРАТУРА

1. БОРЕЙКО А. А., ВОРОНЦОВ А. В., КУШНЕРИК А. А., ЩЕРБАТЮК А. Ф. *Алгоритмы обработки видеоизображений для решения некоторых задач управления и навигации автономных необитаемых подводных аппаратов* // Подводные исследования и робототехника. – 2010. – №1. – С. 29–39.
2. ВАУЛИН Ю. В., ДУБРОВИН Ф. С., КУШНЕРИК А. А., ТУФАНОВ И. Е., ЩЕРБАТЮК А. Ф. *Малогобаритный необитаемый подводный аппарат МАРК нового поколения для выполнения групповых операций* // Мехатроника. Автоматизация. Управление. – 2012. – №6. – С. 59–65.
3. ИНЗАРЦЕВ А. В., ПАВИН А. М. *Интегрированная система технического зрения и управления АНПА для поиска и обследования протяжённых кабельных линий* // Подводные исследования и робототехника. – 2007. – №2. – С. 15–20.
4. ANTONELLI G., САПТИ А., CASALINO G., INDIVERI G., PASCOAL A., CAFFAZ A. *The CO<sup>3</sup>AUVs project: Overview and current progress* // Proceedings of the OCEANS 2011 MTS/IEEE Conference, June 6-9, 2011, Spain.
5. BRADSKI G., КАЕHLER A. *Learning OpenCV: computer vision with the OpenCV library* – 2008. – O'Reilly. – 555 p.
6. BROWNING B., TRYZELAAR E. *ÜberSim: A multi-robot simulator for robot soccer* // Proceedings of Autonomous Agents and Multi-Agent Systems, AAMAS'03. – 2003. – P. 948–949.
7. GANS N. R., DIXON W. E., LIND R., KURDILA A. *A hardware in the loop simulation platform for vision-based control of unmanned air vehicles* // Mechatronics. – 2009. – Vol 19. – Issue 7. – P. 1943–1056.
8. SIMMONS R., JAMES D., *Inter Process Communication: A Reference Manual* // <http://www.cs.cmu.edu>.
9. <http://www.auvsifoundation.org/foundation/competitions/robosub/>
10. <http://ewh.ieee.org/r10/singapore/oes/sauvc/>
11. <http://sauc-europe.org/>
12. БОБКОВ В.А., МОРОЗОВ М.А. *Моделирование работы гидролокционных устройств средствами машинной графики* // Подводные исследования и робототехника. – 2012. – №1. – С. 46–51.
13. БАГНИЦКИЙ А.В., ИНЗАРЦЕВ А.В., ПАВИН А.М., МЕЛЬМАН С.В., МОРОЗОВ М.А., *Модельное решение задачи автоматической инспекции подводных трубопроводов с помощью гидролокатора бокового обзора* // Подводные исследования и робототехника. – 2011. – №1. – С. 17–23.

### Об авторах

Морозов Михаил Александрович - ведущий инженер-программист лаборатории машинной графики Института автоматизации и процессов управления ДВО РАН.

E-mail: morozov@iacp.dvo.ru

Туфанов Игорь Евгеньевич - аспирант Института проблем морских технологий ДВО РАН.

E-mail: ituphanov@gmail.com

# Построение распределённой мультимедийной виртуальной среды с многоканальной визуализацией медиа-данных на графических акселераторах.

Б.Б. Морозов, Б.С. Долговесов, Б.С. Мазурок, М.А.Городилов  
Институт Автоматики и Электротметрии СО РАН, Новосибирск, Россия  
bsd@iae.nsk.su

## Аннотация

*Описывается подход к разработке технических решений для создания распределённой мультимедийной виртуальной среды реального времени. Основная задача среды – управление формированием и доставка мультимедийного контента от множественных источников множественным потребителям.*

**Ключевые слова:** мультимедийные среды, реальное время, коммутация, мониторинг.

## 1. ВВЕДЕНИЕ

Быстрый рост потребностей и научно-технологической базы в сфере приложения разнородных систем реального времени обуславливает постепенный переход от частных решений к универсальным подходам, в частности созданию специфических коммуникативных сред для многих предметных областей. Это требует развития как собственно предметного инструментария, так и общих подходов в обработке данных и коммуникации. Конкретными задачами для аудиовизуальных сред реального времени являются программно-аппаратные решения подготовки и ввода разнотипных многопоточковых данных в распределённую мультимедийную виртуальную среду (PMBC) с соответствующей дальнейшей обработкой и отображением. Насущная потребность в техническом переоснащении отчётливо проявляется в соответствующих предметных областях: системах управления и принятия решений, ситуационного моделирования, мониторинга и управления технологическими процессами; обучающих и тренажёрных системах. Мировая и отечественная практика показывает, что практически любое инновационное техническое решение в этих областях требует проработки новых подходов к построению систем, что ведёт к быстрой эволюции и устареванию стандартов. В частности, развитие современных средств обработки и передачи мультимедийных данных позволяет перейти к новым технологическим решениям для организации многоканального ввода и демонстрации аудиовизуальных данных с помощью специализированных систем управления, ввода, коммутации, мониторинга и вещания.

Суть предложенного здесь подхода в оригинальном способе организации высокопроизводительной цифровой среды передачи медиа-данных с иерархической сетевой архитектурой, распределённым управлением и функциональностью. Физическая среда строится на базе набора типовых устройств ввода, коммутации и визуализации медийных данных для широкого спектра форматов и источников. Такая распределённая среда обеспечивает гибкое

конфигурирование и возможность интерактивного управления процессами посредством подключения соответствующих контроллеров в любом доступном месте среды. Обеспечивается высокая надёжность за счёт распределённой архитектуры, сохраняющей общую работоспособность при выходе из строя отдельных частей.

## 2. СРАВНИТЕЛЬНОЕ ОПИСАНИЕ РЕШЕНИЙ

### 2.1 Типовой подход.

Существующие типовые решения для достаточно масштабных (например, многозальных) систем визуализации и мониторинга обычно строятся на основе матричных коммутаторов [1] некомпьютеризованных сигналов и существенно проигрывают предлагаемым новым решениям как по техническим характеристикам (набору ограничений), так и по сложности инсталляции и обслуживания. Кроме того, это обычно совсем не среда, а достаточно жёсткая система, решающая текущую конкретную задачу. С точки зрения технических средств проблемной функциональностью, в частности, является: переключения разноформатных источников без подрывов; генерация мультиоконного отображения (PIP); многоканальное превью видео для удалённого управления; полиэкранное отображение (в частности, видеостены) и т.п. Реализация подобного функционального набора потребует применения самых дорогих моделей матричных коммутаторов с преобразованием форматов (и масштабированием) на каждом канале в комбинации с рядом отдельных специализированных видеопроцессоров (многоканальных, полиэкранных...). Также большие технические и финансовые затраты необходимы для реализации системы и интерфейсов управления средой, как централизованных, так и распределённых. Например, организация центральной аппаратной (удалённой) требует установки крайне дорогостоящих масштабных матричных коммутаторов для обеспечения доступа ко всем источникам и потребителям аудиовизуального контента одновременно с соответствующей многоканальной кабельной разводкой, особенно затратной для аналоговых систем.

### 2.2 Основы предлагаемого подхода.

Предлагаемый альтернативный подход предполагает использование сжатых (компрессированных) цифровых данных для удалённой передачи в универсальных сетевых средах в комбинации с локальной коммутацией некомпьютеризованных (плоских) сигналов непосредственно на событийных площадках (залах, кабинах, операционных и т.п.). Такие решения позволяют строить уровни иерархии источников и потребителей (в т.ч. мультимплексирование, уплотнение), что позволяет значительно снизить коммуникационные нагрузки по сравнению со схемой

каждый-с-каждым, типовой для систем на матричных коммутаторах. Использование современных высокопроизводительных компьютерных платформ в сочетании с мощными графическими акселераторами позволяет производить в реальном времени произвольные преобразования форматов, микширование, наложения графики, генерацию из набора источников мультимедийных выходных потоков для полиэкранного отображения, в том числе отдельных превью-потоков для мониторинга и управления.

В качестве основной физической среды передачи данных предлагается использовать сеть Ethernet 1 или 10 Gb/s по меди или оптике, в зависимости от расчётных нагрузок и дистанций связи. Эти сети сейчас легко масштабируются без каких-либо переделок конечных источников и потребителей. Это решение кардинально уменьшает затраты на кабельную прокладку и её обслуживание по сравнению с матричной коммутацией.

Можно выделить три необходимых функциональных задачи обработки сигналов для системы коммутации и мониторинга в РМВС. Это:

- ввод медиа-данных в среду;
- преобразования и микширование (генерирование виртуальных источников), как для системы управления, так и для формирования собственно вещаемого медиа-контента;
- итоговое финальное отображение сформированных и выбранных медиа-данных на доступных устройствах вывода.

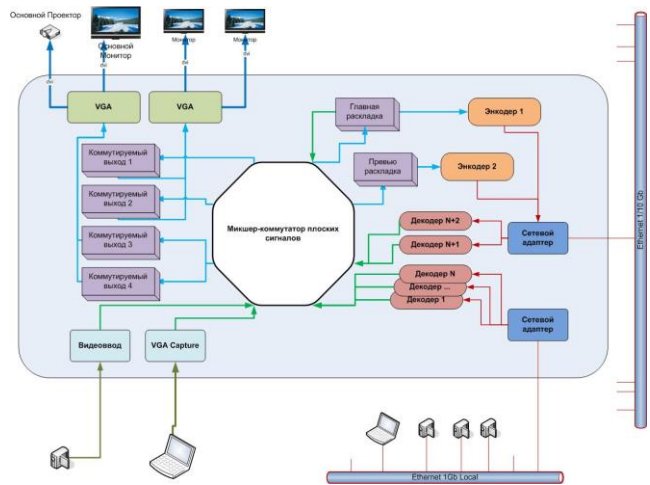
Собственно управление коммутацией и вообще средой в целом – отдельная задача специализированного ПО управления.

Если производительности современных сетевых решений достаточно для параллельной передачи нескольких десятков видеопотоков высокой чёткости (~15-150 в компрессированном виде [2]), то производительность современных компьютерных платформ ограничена примерно десятью потоками реального времени (обычно меньше десяти, зависит от конкретных условий и форматов данных по текущим оценкам). Это положение требует разработки специфических эффективных решений для организации полнофункциональной коммутации на типовых аппаратно-программных средствах.

Микширование и коммутация (выбор из существующих источников и файлов, наложение графики, составление раскладок или мозаик) в РМВС имеет целью генерацию виртуальных комбинированных источников для дальнейшего использования как для вещания, так и для управления. В качестве более сложного типа «микширования», например, возможно использование погружения различных медийных источников в трёхмерную виртуальную синтезируемую среду посредством применения специальных систем типа «Виртуальный презентер» [3-4]. Такие операции производятся наиболее эффективно с некомпрессированными (плоскими) сигналами, в этом случае также минимальны временные задержки на обработке и переключениях. Работа с компрессированными сигналами в данном случае приводит к потере качества изображения, высоким накладным расходам и дополнительным задержкам. Параллельно производится

коммутация и микширование соответствующих звуковых данных, как в автоматическом режиме, так и с интерактивной настройкой. Для большинства задач, связанных с обработкой звука, производительность современных систем обычно более чем достаточна. (Специализированные конференционные системы, например, в данном случае рассматриваются как отдельные внешние источники-приёмники медиа-данных, клиенты среды).

В рассматриваемом подходе узловым архитектурным решением является использование программно-аппаратных микшеров-коммутаторов разного уровня для организации иерархии. Типовой микшер-коммутатор (на базе персонального компьютера), предназначенный для использования на локальном уровне, может использоваться одновременно для коммутации некомпрессированных (плоских) сигналов на локальные устройства отображения (графические акселераторы с полиэкранным выводом) и для формирования виртуальных источников (в частности, «мозаик» для операторов) с последующей передачей во внешнюю среду (со сжатием). В качестве некомпрессированных сигналов для микширования и коммутации используются как декодированные сетевые мультимедийные потоки из внешней среды, так и сигналы с аппаратных устройств медиа-ввода (видеограбберов и т.п.).



**Рисунок 1 Внутренняя логика работы микшера-коммутатора.**

В принципе, на локальном уровне эти задачи (помимо кодирования/декодирования) можно решать и с помощью типового оборудования для матричной коммутации, но в этом случае теряется унификация оборудования и гибкость, присущая компьютерным решениям, усложняются протоколы управления.

В случае нехватки производительности для обработки всех необходимых сигналов в конкретном расположении количество локальных микшеров-коммутаторов можно наращивать (масштабировать), либо распределять функциональность по разным рабочим станциям (серверам). Например, возможно сконфигурировать только отображающие, вводные или микширующие сервера в зависимости от требований конкретной задачи. Характерным примером является оперативное конфигурируемое мультимедийное и полиэкранное отображение на больших высокоразрешающих мониторах (видеостенах) для центров управления и мониторинга. Управление всеми локальными

микшерами-коммутаторами осуществляется удалённо, посредством единой стандартизованной физической сетевой среды (Ethernet). Это позволяет легко организовывать удалённое распределённое и/или централизованное управление и контроль, гибко перестраивать конфигурации при любых изменениях требований.

На более высоком уровне иерархии используются коммутаторы потоковых сигналов (сжатых), энкодеры/декодеры, прокси-сервера, и прочее сетевое оборудование. Для управления всеми коммутациями и правами доступа используются управляющие рабочие станции со специализированным программным обеспечением (фактически это могут быть стандартные персональные компьютеры, в некоторых случаях возможно повышенной производительности), которые могут размещаться как локально в событийных/демонстрационных расположениях, так и в выделенных центрах управления. При этом затраты на организацию центров управления кардинально снижаются по сравнению с решениями на базе матричных коммутаторов, так как фактически всё управление строится на базе типовых рабочих станций или даже мобильных и планшетных компьютеров. Нет необходимости в отдельной многоканальной кабельной разводке для нежатимированных сигналов и в центральных матричных коммутаторах.

Отображение и мониторинг аудиовизуальной информации в организованной PMBC возможны на широком спектре цифровых устройств - от персональных компьютеров до телевизионных мониторов с сетевым подключением, сет-топ боксов и мобильных персональных устройств. Главным условием является наличие сетевого подключения и возможность декодирования сжатых медианых.

### 3. ПРИМЕР КОНФИГУРИРОВАНИЯ ОБОРУДОВАНИЯ

Минимальная система для оснащения одного из залов (комнат совещаний) с несколькими источниками видео сигналов высокой четкости (HD) потребует два типа модулей: интегрированный микшер-коммутатор и управляющий модуль.

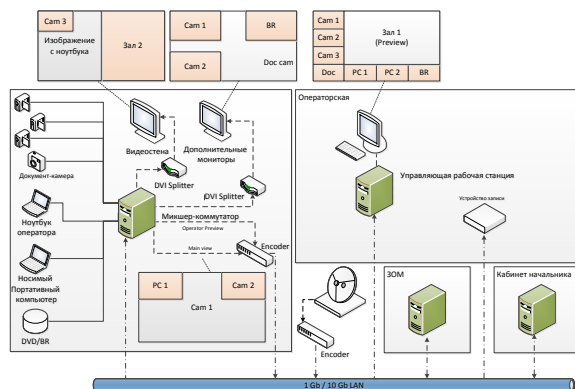


Рисунок 2 Пример конфигурации оборудования.

Управляющий модуль состоит из персонального компьютера с двумя мониторами (для интерфейса управления и многоканального мониторинга, соответственно) и программного обеспечения, которое позволяет осуществлять

контроль над отображаемыми источниками и настройкой коммутации.

Модуль микшера-коммутатора состоит из персонального компьютера с установленными устройствами видеозахвата соответствующего формата и специального ПО. К этому модулю подключаются видео-камеры и другие источники сигналов. Также подключение источников возможно через высокоскоростную локальную сеть. Задача этого модуля состоит в генерации нескольких выходных аудио-видео потоков в виде настраиваемых раскладок-мозаик для вывода их через HDMI/DVI или в сеть. Один из этих видео потоков предназначен для предпросмотра всех входных источников микшера-коммутатора для удалённой настройки режимов микширования. Второй поток предназначен для последующей записи и вещания в сеть. Остальные - для отображения на локальных устройствах (видеостенах, ЖК мониторах).

Количество входных источников обрабатываемых одновременно на одном микшере коммутаторе ограничено (обычно менее десяти потоков высокой четкости). Сгенерированные аудио-видео потоки можно просматривать практически на любом компьютере, подключённом к высокоскоростной локальной сети, с установленным программным обеспечением для воспроизведения сетевых аудио-видео потоков. В частности, это могут быть и мобильные устройства. При необходимости использования специфических форматов (Например, для беспроводного вещания на устройства типа iPad или iPhone) могут использоваться дополнительные прокси-сервера для преобразования форматов вещания.

При необходимости одновременного мониторинга и управления вещанием в нескольких залах возможна установка отдельного микшера-коммутатора в центре управления (операторской) сконфигурированного исключительно для отображения максимального числа источников на одном или нескольких больших мониторах высокого разрешения (режим видеостены).

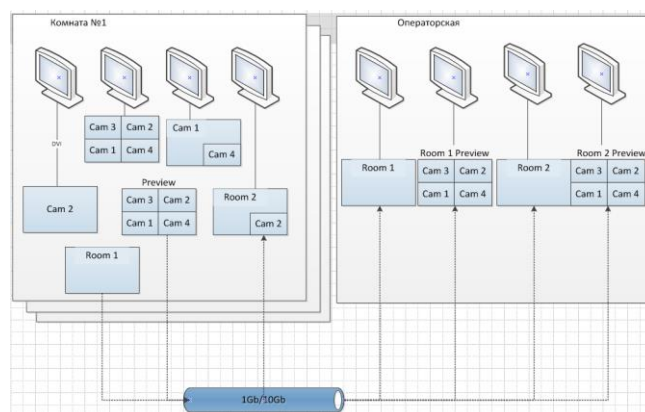


Рисунок 3 Пример возможных наборов раскладок для отображения и мониторинга

В графическом интерфейсе управления оператору могут визуализироваться не только «результатирующая» и «превью» раскладки микшера-коммутатора, но и текущие результирующие назначения всех выходных воспроизводящих устройств в зале (комнате). Коммутация может осуществляться в интерфейсе оператора простым перетаскиванием изображения доступного источника на



соответствующий выход (мышкой или с помощью сенсорного экрана).



Рисунок 4 Пример генерируемых раскладок (мозаик) в интерфейсе управляющего ПО

#### 4. ОСОБЕННОСТИ РЕАЛИЗАЦИИ

Особенности аппаратной и программной реализации технических решений для создания среды частично рассмотрены в отдельной статье (в настоящий момент не опубликована), посвящённой эффективным и надёжным методам работы с потоковыми данными средствами персонального компьютера.

Интересными представляются также работы по созданию новых стандартов и средств работы с мультимедийными данными в сетевой среде. В частности, стандарт рабочей группы 802.1 Audio Video Bridging (AVB) института IEEE [5]. Этот стандарт предназначен для передачи (и трансляции) в сетях Ethernet синхронных потоковых данных с малыми задержками. Основное предназначение – передача телевизионных сигналов без потерь аналогично HD-SDI, но в сетях Ethernet. Очевидно, что в рамках этого стандарта возможна и передача сжатых данных с гарантированными временными характеристиками, что может быть ключевым требованием для части возможных приложений PMBC, как в пределах локальных расположений, так и среды в целом. Проблема использования на практике заключается в том, что в настоящий момент доступное оборудование, поддерживающее стандарты AVB фактически отсутствует на рынке. Консорциумы производителей только объявляют о начале поддержки этого стандарта. Тем не менее, в предложенном подходе к построению PMBC переход на новое или совместное использования оборудования текущих и новых стандартов не должно вызывать существенных трудностей за счёт гибкости и универсальности средств управления средой.

#### 5. ЗАКЛЮЧЕНИЕ.

Одним из существенных преимуществ описываемого подхода является возможность постепенного и поэтапного наращивания и совершенствования как системы в целом, так и её частей. Так как аппаратно архитектура базируется на стандартном модульном оборудовании широкого профиля,

которое сейчас стремительно развивается и функционально, и технически при этом дешевле, то есть уверенность в возможности решения разнообразных возникающих задач. Также, распределённость архитектуры приводит к большей её надёжности в целом, так как фактически отсутствуют ключевые узлы, выход из строя которых может привести к полной неработоспособности системы; высокая взаимозаменяемость оборудования, ремонтпригодность и модифицируемость даже «своими силами» (без необходимости вызова сторонних специалистов).

Предложенный подход к построению PMBC демонстрирует новизну разработки в части средств управления средой в сочетании с практической реализуемостью. Видны существенные преимущества подхода для практических целей. Потенциально развитие подобных подходов должно привести к появлению стандартов на технические решения для PMBC, в том числе промышленных.

#### 6. ЛИТЕРАТУРА

- [1] Ерохин А.В., Фрадкин Б.Г., Левин И.И., Рыжих О.А. Матричный Коммутатор // Патент на изобретение № 2103729 Официальный бюллетень Роспатента «Изобретения. Полезные модели» 27.01.1998
- [2] Виктор Дворкович, Михаил Чобану. Проблемы и перспективы развития систем кодирования динамических изображений ([HTTP://MEDIAVISION-MAG.RU/UPLOADS/02%202011/55-57%2002\\_2011.PDF](http://mediavision-mag.ru/uploads/02%202011/55-57%2002_2011.pdf))
- [3] Долговесов Б. С., Морозов Б. Б., Тарасовский А. Н., Пархоменко И. Н. Система обучения и презентаций с использованием технологии «погружения» лектора в интерактивную виртуальную среду // Труды международной конференции VEonPC'2006 (Алтай 6-10 июля, Протвино 17-18 ноября 2006) и третьей научной конференции СИМ'2006 (Москва, июнь-июль 2006), ИФТИ, Москва, 2006, с.99–102.
- [4] Долговесов Б.С., Лаврентьев М. М., Морозов Б. Б. Интерактивный обучающий комплекс на основе технологии интегрированной виртуальной среды // Труды XVIII Всероссийской научно-методической конференции Телематика'2011 (Санкт-Петербург, Россия, СПб ГУИТМО, 20–23 июня 2011). СПб ГУИТМО, т. 1. С. 138–139.
- [5] IEEE 802.1 Audio/Video Bridging Task Group Audio Video Bridging ([HTTP://www.ieee802.org/1/pages/avbridges.html](http://www.ieee802.org/1/pages/avbridges.html))

### Development of Distributed Multimedia Virtual Environment with multichannel visualization of multimedia data by graphic accelerators.

#### Abstract

An approach to the development of technical solutions for the creating of a real-time Distributed Multimedia Virtual Environment is described. The main objective of the environment - control the formation and delivery of multimedia content from multiple sources to multiple recipients.

**Keywords:** *multimedia environment, real-time, commutation/switching, monitoring.*

### **About the author**

Boris B. Morozov is a scientific researcher of Software Systems for Computer Graphics Laboratory at Institute of Automation and Electrometry SB RAS. His contact email is [bbm@sl.iae.nsk.su](mailto:bbm@sl.iae.nsk.su)

Boris S. Dolgovesov (Ph.D.) is a head of Synthesizing Visualization Systems Laboratory at Institute of Automation and Electrometry SB RAS. His contact email is [bsd@iae.nsk.su](mailto:bsd@iae.nsk.su)

Boris S. Mazurok is a scientific researcher of Synthesizing Visualization Systems Laboratory at Institute of Automation and Electrometry SB RAS. His contact email is [boris@albatros.iae.nsk.su](mailto:boris@albatros.iae.nsk.su)

Mikhail A. Gorodilov is a Ph.D. student at Institute of Automation and Electrometry SB RAS. His contact email is [gorodilovm@gmail.com](mailto:gorodilovm@gmail.com)

# Интерактивный выбор объектов трёхмерной сцены

Е.И. Коростелев, М.А. Городилов, Б.С. Долговесов  
Институт Автоматики и Электротехники СО РАН, Новосибирск, Россия  
kore3d@gmail.com

## Аннотация

В статье предлагается метод выбора трёхмерных объектов виртуальной сцены для систем визуализации реального времени. Метод использует один бит буфера трафарета и позволяет определять порядковый номер объекта, которому принадлежит некоторый пиксель итогового изображения. Таким образом, он позволяет определять объекты под курсором мыши, что позволяет интерактивно взаимодействовать с ними. Для этого не требуется дополнительных проходов визуализации, и отсутствуют ограничения на используемые при визуализации материалы (процедурные сетки, полупрозрачность, альфа-тест).

**Ключевые слова:** выбор объектов, буфер трафарета, визуализация реального времени.

## 1. ВВЕДЕНИЕ

Большинство систем виртуальной реальности предоставляют возможность интерактивного взаимодействия с рендеримыми виртуальными объектами. Как правило, взаимодействие осуществляется путём выбора конкретного объекта курсором мыши, а выбранный объект выделяется специальным образом. Кроме этого, в презентационных и обучающих системах [1] существует необходимость рисовать поверх объектов, для чего используется аналогичный механизм выбора.

В системах визуализации, где число доступных для выбора объектов незначительно, как правило, применяется метод выбора объектов путём нахождения ближайшего пересечения луча с набором доступных объектов [2]. Этот алгоритм реализуется на центральном процессоре и не позволяет точно выделять динамически изменяющиеся объекты (например, процедурно-генерируемые сетки).

Одним из методов определения порядкового номера (порядок визуализации) объекта под курсором является: отдельный проход рендеризации всех видимых объектов сцены с упрощёнными материалами и запись в буфер цвета (как правило, одноканальный 32-битный) идентификатора объекта [3]. Этот метод использует графический процессор, но имеет ряд недостатков. Во-первых, он требует повторную рендеризацию всех объектов (списка батчей), что накладно в случае динамически изменяющихся сеток (например, скелетная анимация). Во-вторых, в «простом» материале, используемой на отдельной стадии, необходимо учитывать свойства материалов, которые отбраковывают часть поверхности (например, инструкция discard в шейдере) или дают альфа-канал (альфа-тест).

Авторами статьи предлагается метод, лишенный изложенных выше недостатков и обладающий определёнными преимуществами: простота интеграции в существующую систему, производительность, возможность реализации для видеокарт уровня DirectX 9. Несущественными недостатками

предлагаемого метода является необходимость использовать 1 бит буфера трафарета [4] (stencil buffer), теста трафарета (stencil test) и дополнительной видеопамати (до 32 МБ на разрешение сверхвысокой чёткости 4K UHD TV).



Рис 1: Выбор одного из 1000 виртуальных объектов

Пример использования разработанного метода приводится на рисунке 1. На изображении представлен результат визуализации тысячи трёхмерных объектов (разноцветных сфер), поверхность которых определяется специальной текстурой (отбраковка части поверхности), а цвет задан случайно. Объект, находящийся под курсором, выделен специальным цветом.

## 2. АЛГОРИТМ

Основой метода является использование буфера и теста трафарета. Буфер трафарета используется для перезаписи 1 бита после визуализации каждого объекта (вызов отрисовки). Поскольку требуется перезаписывать только 1 бит, данный метод может быть скомбинирован с другими алгоритмами, использующими трафарет, либо применяться без ограничений. Большинство алгоритмов визуализации не используют тест трафарета и ограничиваются использованием буфера и теста глубины (буфера формата D24SX, где D24 – 24 бита для записи глубины). Таким образом, достаточно использовать буфер формата D24S8.

Дополнительно, для хранения идентификаторов рендеримых объектов, потребуется один буфер, соответствующий размеру кадра (обозначим его *PickBuffer*). Для него достаточно формата R16G16. В красный канал (R16) будет записываться идентификатор объекта.

При использовании предлагаемого метода визуализация всех объектов производится как обычно, без ограничений. Алгоритм поддерживает буферы с несколькими «семплами» и не влияет на поддержку алгоритмов устранения ступенчатости.

Рассмотрим общую схему работы алгоритма:

1. Начало кадра: инициализируем буфер трафарета значением 0.
2. Визуализация объектов сцены: включаем запись в буфер трафарета и пишем 1 бит, если пиксель прошёл тест глубины (либо тест выключен)
- 2.1 После каждого вызова отрисовки временно переключаем буфер кадра на *PickBuffer* и с включенным тестом трафарета на проверку бита с шага 1 рисуем прямоугольник в 1 пиксель (размер области, в которой необходимо определить ближайший объект). При этом, во время проверки сбрасываем этот бит.
3. Конец кадра: пересылаем содержимое *PickBuffer* в системную память.
4. Декодирование области *PickBuffer* (распаковка идентификатора) – получение номера объекта

На шаге 2.1 может быть выбрана область произвольного размера, но её существенное увеличение приведет к падению производительности, поскольку потребуются обновлять больше значений в видеопамяти. Каждая видеокарта имеет ограничение скорости заполнения пикселями (филлрейт).

Содержимое буфера идентификаторов (*PickBuffer*) для полного кадра тестовой сцены представлено на рисунке 2. На изображении для наглядности заполнены все пиксели, но, в случае использования однопиксельной области, в буфере обновляется только 1 пиксель. Кроме этого, для лучшего визуального различия содержимого буфера цвета были переназначены и использован красный цвет для подсветки текущего выбранного объекта.

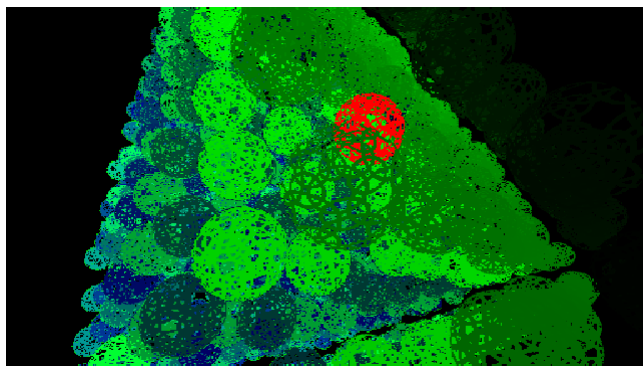


Рис 2: Буфер идентификаторов 3d-объектов полного кадра

При использовании мультисемплирования буфер идентификаторов содержит несколько субпикселей на пиксель. Перед передачей его в системную память производится преобразование субпикселей в пиксель. Данный механизм поддерживается аппаратно, но плохо конфигурируется и, при настройках по умолчанию, даёт усредненные значения по границам объектов, что нельзя интерпретировать как идентификаторы. На большинстве видеокарт 8-летней давности поддерживается возможность задать маску семплирования, что позволяет исключить преобразование значений. В случае отсутствия возможности маскировать какой-то один субпиксель, допускается использование области из 9 пикселей на шаге 2 и отбраковке значений, если в окрестности не найдено 2-3 совпадения.

Копирование *PickBuffer* в системную память (третий шаг алгоритма) следует выполнять с задержкой, как минимум, в 1 кадр, т.е. на текущем кадре считывать предыдущий. При

несоблюдении этого условия возникает точка синхронизации между графическим и центральным процессором, что отрицательно сказывается на производительности. Причина в том, что центральный и графический процессор работают асинхронно и, к моменту выполнения шага 3 центральным процессором, данные будут еще не готовы. Кроме этого, все команды графическому процессору буферизируются и могут задерживаться на несколько кадров.

Небольшое запаздывание выбора объекта под курсором мыши (из-за задержки считывания *PickBuffer*) устраняется путём прогнозирования перемещения курсора. Для этого используются значения ускорения и скорости движения мыши за несколько предыдущих кадров, а для текущего кадра берется прогнозируемая позиция.

### 3. РЕЗУЛЬТАТЫ

Реализация алгоритма тестировалась на различном аппаратном обеспечении. При этом типовым случаем считалось наличие большого набора объектов и использование алгоритмов устранения ступенчатости (мультисемплирование). Приводимые далее результаты были получены в реализации алгоритма на Direct3D11 и исполнение его на видеокarte с аппаратной поддержкой DirectX10 (функциональный профиль 10.0 устройства Direct3D11). Кроме этого, работоспособность алгоритма проверена в рамках функций Direct3D9 (и функционального профиля 9.3 устройства Direct3D11). Результаты замера производительности для реализации Direct3D11 приведены в таблице 1.

Видеокарта \ N	512	1000	2197
NVIDIA 8800 GTX	3 мс	4 мс	5 мс
AMD FirePro W8000	5 мс	13 мс	26 мс

Табл. 1: Производительность при выборе среди N объектов

Как видно, из таблицы наиболее эффективно метод работает на видеокартах NVIDIA. На результат тестирования при этом влияет, как версия драйвера, так и поколение видеочипа. Поскольку в большинстве случаев не требуется выбирать объект из десятков сотен других, производительности достаточно для решения большинства задач, где необходимо интерактивное взаимодействие с трёхмерными объектами.

Разработанный метод также позволяет выбирать объекты последовательно, т.е. выбирать объекты не прошедшие растеризацию или последовательно прятать ближайшие. Такой механизм удобен при необходимости выбирать объекты, находящиеся позади полупрозрачных. Данная функция реализуется посредством составления «чёрного» списка объектов, для которых игнорируется запись в буфер идентификаторов.

Возможность рисования поверх поверхности объекта после выбора некоторого объекта обеспечивается путём дополнительной растеризации этого объекта в конце кадра. Это позволяет определить текстурные координаты выбранной поверхности и реализуется с использованием уже заполненного на этапе подготовки кадра буфера глубины.

## 4. ЗАКЛЮЧЕНИЕ

В статье предложен эффективный алгоритм выбора объектов, использующий аппаратные возможности современных и старых поколений графических процессоров. Разработанная технология применяется в интерактивной системе визуализации реального времени, что позволяет предоставить пользователю возможность выбирать объекты, менять их свойства, а также рисовать на поверхности виртуальных объектов.

## 5. БИБЛИОГРАФИЯ

- [1] Долговесов Б.С. Системы виртуальной реальности для космических тренажерных комплексов и интерактивных презентаций // Сборник материалов конференции «Пилотируемые полеты в космос», ИМБП РАН, 2011. С. 66.
- [2] Brian Hook, Mouse Ray Picking Explained, <http://bookofhook.com>
- [3] D. Cantor et al., WebGL Beginner's Guide, 2012, pp.257-285
- [4] J. Zink et al. Practical Rendering and Computation With Direct3D 11, 2011, pp.241-258

# Interactive object picking by stencil buffer technique

## Abstract

A new approach for fast object picking is presented in the article. Described technique uses a stencil test. No additional rendering passes are needed. So this technique allow to select any meshes (procedural, alpha blended, etc) and works fine with multisample render targets. And also it's implementable for d3d9-capable hardware.

*Keywords: object picking, stencil buffer, real-time rendering.*

## About the author

Evgeny I. Korostelev is a Ph.D. student at Institute of Automation and Electrometry SB RAS. His contact email is [kore3d@gmail.com](mailto:kore3d@gmail.com)

Mikhail A. Gorodilov is a Ph.D. student at Institute of Automation and Electrometry SB RAS. His contact email is [gorodilovm@gmail.com](mailto:gorodilovm@gmail.com)

Boris S. Dolgovesov (Ph.D.) is a head of Synthesizing Visualization Systems Laboratory at Institute of Automation and Electrometry SB RAS. His contact email is [bsd@iae.nsk.su](mailto:bsd@iae.nsk.su)

## О подсистеме истории в среде научной визуализации SharpEye

П.А.Васёв<sup>1</sup>, С.С.Кумков<sup>1,2</sup>, Е.Ю.Шмаков<sup>2</sup>

<sup>1</sup> Институт математики и механики им. Н.Н.Красовского, УрО РАН, Екатеринбург

<sup>2</sup> Уральский Федеральный университет, Екатеринбург

{pavel.vasev, sskumk, schmak.ekb}@gmail.com

### Аннотация

В докладе рассматривается разрабатываемая авторами модульная среда научной визуализации SharpEye. Обсуждаются пути реализации подсистемы редактируемой истории, в возможности которой должны входить функции отката и повтора манипуляций, проделанных пользователем, сохранение и восстановлению подобранного вида сцены.

**Ключевые слова:** научная визуализация, система визуализации, подключаемые внешние модули, редактируемая история, откат/повтор действий

### 1. ВВЕДЕНИЕ

В течение последних лет авторы разрабатывают среду SharpEye – конструктор систем научной визуализации [3,4]. В основу среды положен модульный принцип: сама среда не содержит никаких алгоритмов восстановления геометрических объектов по данным пользователя, а лишь предоставляет программный интерфейс (API) для наполнения сцены и средства манипулирования ею. При этом подразумевается, что для каждой конкретной задачи или класса задач пользователь самостоятельно создаёт процедуру восстановления геометрического образа визуализируемых данных и оформляет её в виде внешнего модуля, подключаемого системой при запуске и используемого при загрузке соответствующих файлов данных. Такой подход основывается на идее об уникальности визуализируемых особенностей для каждого класса задач и, как следствие, специфичности процедур восстановления геометрических образов.

Существующие системы визуализации (например, ParaView) зачастую всё-таки идут по пути включения в свой состав более или менее широкого набора готовых процедур, а на случай обработки нестандартных данных пользователю предоставляется возможность формирования сцены в файле оговоренного формата. Подход, предложенный авторами, представляется более гибким, так как помимо собственно процедуры восстановления программный модуль, разработанный пользователем, может содержать и другие процедуры, например, анимации созданных объектов в случае визуализации динамических процессов или специфических элементов управления.

Основной целью пользователя при работе с системой визуализации является подбор такой точки зрения (расположения и ориентации объектов перед камерой, их визуальных атрибутов, режима освещённости и т. д.), чтобы наиболее чётко были видны визуализируемые особенности. Разрабатываемая система предоставляет соответствующий функционал управления сценой и объектами как через пользовательский интерфейс (GUI), так и через программный.

В процессе ручного подбора точки зрения зачастую оказывается, что в результате действия или набора действий, проделанных пользователем, наглядность получаемой картинки ухудшилась. Поэтому удобно было бы отменить проделанные действия и попробовать другие направления улучшения наглядности изображения. Соответственно, система должна предоставлять пользователю функционал редактируемой истории [1,2].

Но реализация редактируемой истории достаточно нетривиальна. Не слишком сложно сохранять и восстанавливать стандартные параметры объектов и сцены (текущая камера, её пространственное и угловое положение; параметры и положения источников света; цвет, прозрачность, видимость отдельных объектов в сцене). Однако система позволяет пользователю задавать собственные свойства объектов и управлять ими. При этом их тип и методика работы с ними системе априори неизвестны — соответствующий функционал полностью заключён в пользовательских модулях. Это и объясняет сложность реализации данной подсистемы. Кроме того, непросто спроектировать взаимодействие подсистемы истории и подсистемы анимации, поскольку действие последней заключается в значительном количестве изменений параметров сцены и/или объектов, осуществляемых через программный, а не пользовательский интерфейс.

Также запись действий пользователя может быть сохранена в файл и в дальнейшем повторена, что даёт возможность сохранения удачно подобранной точки зрения не в виде статичной растровой картинки, а именно в виде состояния сцены.

В статье обсуждается реализация подсистемы истории, преодолевающей указанные проблемы.

### 2. ЭЛЕМЕНТЫ ИСТОРИИ

При разработке системы авторы рассмотрели два подхода к реализации подсистемы истории.

Первый подход подразумевает, что перед выполнением любой операции, которая может повлиять на состояние сцены, система подаёт команду менеджеру истории на сохранение полного снимка сцены (*независимый объект истории*). В данный снимок входит информация о местоположении источников света и их свойствах, информация о расположении и направленности камер, а также значения, которые принимают стандартные и пользовательские свойства моделей.

Преимуществом данного подхода является простота реализации операций отмены и повтора: необходимо лишь восстановить сохранённые в снимке объекты и установить их свойства в соответствии со снимком. При этом сам процесс сохранения снимка не зависит от типа операции. Однако

снимок системы может быть достаточно большим, если сцена содержит много объектов. Как следствие, это может вызвать проблемы как с хранением снимков в оперативной памяти, так и с сохранением их на диске. Кроме того стоит заметить, что если мы храним историю в виде набора снимков, то большая часть информации в каждом из них избыточна, поскольку каждая из операций, как правило, изменяет лишь малую часть параметров сцены, что приводит к тому, что соседние в истории снимки отличаются незначительно. Данное свойство является недостатком первого подхода и приводит к мысли о том, что можно хранить только изменения сцены (*инкрементальный объект истории*).

Во втором варианте реализации подсистемы истории для каждой операции сохраняются только те параметры сцены, которые были изменены. При этом удобно сопоставить с каждой операцией отдельный класс для сохранения изменяемых данных. Классы элементов истории для разных операций реализуют общий интерфейс, декларирующий всего два метода: Undo и Redo. Первый метод позволяет отменить операцию, а второй повторно ее применить, если она была отменена ранее. Например, для операции «поворот камеры» можно создать класс, имеющий несколько внутренних полей: «имя камеры», «угол поворота» и «ось вращения». Тогда метод Redo данного класса будет поворачивать заданную камеру относительно текущего положения на заданный угол, а метод Undo — на угол, ему противоположный.

Преимущества данного варианта очевидны: сохраняется минимум данных, необходимых для отката операции; каждый класс элемента истории связан с конкретным преобразованием сцены, что позволяет отображать в пользовательском интерфейсе текст элемента в виде, понятном пользователю, например, для приведённого выше примера текст может выглядеть так «Поворот камеры 'имя камеры' на угол 'угол поворота'». Однако данный подход требует очень аккуратной реализации. Например, такой метод хранения поворотов камеры будет накапливать погрешность: если произвести несколько поворотов камеры, а потом их откатить, то камера может не вернуться в исходное состояние, а быть повернутой на некоторый небольшой угол, который возникнет из-за погрешности вычислений чисел с плавающей запятой. Чтобы избежать данной проблемы можно сохранять не «угол поворота» и «ось вращения», а начальное и конечное положение вектора направления камеры и вектора верха камеры. При таком подходе выполнение отката или повтора операции не повлечёт никаких операций с числами с плавающей запятой, кроме присваивания, таким образом, ценой небольшого увеличения сохраняемых данных проблема накопления погрешности может быть решена.

Существенным недостатком данного подхода является то, что для каждой новой операции над сценой приходится писать код, позволяющий сохранять данные, относящиеся к данному преобразованию, в отличие от первого подхода, где всегда сохраняются одни и те же данные вне зависимости от операции.

Общим у описанных подходов является то, что при произведении действия система сохраняет некий объект, так или иначе поддерживающий действия Undo и Redo. При этом Undo подразумевает, что система находится в состоянии, имевшемся сразу после совершения действия, и переводит её в состояние, имевшееся непосредственно перед

совершением этого действия. Redo производит прямой перевод.

Для удобства использования разумно потребовать от подсистемы истории, чтобы она группировала несколько последовательных однотипных действий в одно суммарное. Например, несколько последовательных поворотов сцены могут быть заменены одним итоговым поворотом. Или несколько последовательных изменений цвета одного и того же объекта могут быть сгруппированы в изменение исходного цвета на конечный. Для этого объект элемента истории может поддерживать метод Merge, который, возможно, соединяет его с другим элементом истории, передаваемым в этот метод.

Конкретная реализация элемента истории для того или иного действия обуславливается сущностью этого действия. В частности, если пользователь в рамках модуля работы со своими объектами предусматривает введение нестандартных свойств и работу с ними, то соответствующие обработчики могут взаимодействовать с менеджером истории, поставляя ему объекты истории работы с этими свойствами.

### 3. ТРАНЗАКЦИИ В ИСТОРИИ

Некоторые действия пользователя могут разворачиваться в значительные наборы простейших операций. Например, загрузка файла данных реально приводит к созданию модели, наполнению её геометрии, установке свойств цвета, прозрачности, видимости; тут же возможно создание, редактирование и/или удаление камер и/или источников света и т. д. Соответственно, такие наборы простейших действий должны откатываться и повторно выполняться не по одному, а в совокупности.

Это размышление приводит к необходимости создания *транзакций*, специальных объектов истории, задачей которых является объединение набора тривиальных действий в одно составное. При этом методы Undo и Redo транзакции вызывают в соответствующем порядке методы Undo и Redo объектов, объединённых в транзакцию.

### 4. СОХРАНЕНИЕ И ЗАГРУЗКА СЕССИИ

Объект истории может автоматизированно поддерживать сохранение себя в указанный поток и чтение себя из указанного потока. Соответственно, для этого предусмотрены два метода: `Serialize` и `Deserialize`. В случае, когда запись объектом поддерживается, решается несколько задач.

Во-первых, сохранение на диске удачно выбранного состояния сцены с целью его дальнейшего быстрого восстановления. Во-вторых, на этом принципе может быть построена подсистема удалённой визуализации: на стороне пользователя интерфейсная часть программы генерирует действия и передаёт их в канал связи с серверной частью. На второй стороне канала серверная часть производит считывание объектов истории и применяет их к сцене, высылая пользователю получаемую картинку.

Данная возможность может быть полезна, как при личном использовании системы, так и при коллективном. При личном использовании сохранение истории изменений позволяет продолжить сеанс работы с того места, на котором он был прерван. При коллективном же использовании данная возможность становится удобным инструментом для

публикации результатов, поскольку результат представляется не в виде статичного изображения, а поставляется вместе с системой визуализации, позволяющей рассмотреть его во всех подробностях.

## 5. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

Система SharpEye реализована на языке C# для среды исполнения Microsoft .Net 4.0 и выше. Для доступа к своему API среда SharpEye предоставляет библиотеку Contracts, в которой определены интерфейсы всех классов, доступ к которым может быть полезен для организации работы пользовательского модуля со средой. Авторами разработана документация к этой библиотеке (и, фактически, справочник для программиста).

В частности, в этой библиотеке определён интерфейс IHistoryElement, в котором определены методы Undo, Redo, Merge, Serialize, Deserialize и свойство IsIndependent. Последнее свойство позволяет менеджеру истории оптимизировать процессы отката и повтора пользовательских действий: если откатывается или повторяется указанный пользователем набор действий, то можно не производить часть или все промежуточные инкрементальные изменения, а сразу установить состояние сцены из независимого объекта (если таковой присутствует в этом наборе) и продолжить применение инкрементальных поправок от этого состояния.

Внутри самой системы SharpEye наличествует достаточное количество классов, производных от интерфейса IHistoryElement, которые обслуживают запись элементов истории для стандартных действий.

Следует отметить, что доступ в система SharpEye может быть получен не только в рамках программ, написанных для среды .Net, но и в скриптах, написанных на языке Ruby. В систему интегрирован интерпретатор IronRuby. Поэтому при реализации пользовательского объекта типа IHistoryElement методы Redo и Merge могут представлять собой скрипты на языке Ruby, в том числе, генерируемые системой.

Программный доступ к сцене обеспечивается через ссылку на объект типа IStorage, реализующий интерфейс хранилища сцены. (Такая ссылка передается модулю пользователя при его инициализации.) Этот объект, в частности, имеет свойство, возвращающее ссылку на объект, реализующий интерфейс IHistoryManager. Объект такого типа предоставляет средства для работы с историей:

- UndoLast — откат последнего совершённого действия;
- RedoLast — повтор последнего отменённого действия;
- UndoUpTo, RedoUpTo — откат и повтор группы действий до указанного элемента или до конца записанной истории;
- AddElement — добавление элемента истории в конец имеющейся записи; если имелась запись отменённых действий, она теряется;
- OpenTransaction, CloseTransaction — начать/закончить транзакцию; при этом все элементы истории, добавляемые в процессе

формирования транзакции, автоматически добавляются в элемент транзакции;

- SaveSession, LoadSession — запись и чтение истории в/из заданного потока.

В объекте хранилища IStorage предусмотрены методы MakeRestorationPoint и ApplyRestorationPoint. Первый из них является открытым и создаёт снимок сцены, возвращая его в независимом объекте истории; второй, закрытый — принимает независимый объект истории и переводит сцену в соответствующее состояние.

## 6. ВЗАИМОДЕЙСТВИЕ ПОДСИСТЕМ ИСТОРИИ И АНИМАЦИИ

Проблема с анимацией заключается в том, что для «оживления» изображения в сцене производится множество изменений: либо перемещается камера, либо меняется какое-либо свойство того или иного объекта, либо меняются свойства у большого количества объектов (например, одни объекты делаются невидимыми, а другие «включаются»). Если делать эти изменения инкрементальными, то существует опасность накопления ошибки, отмеченная в разделе 2. Поэтому объекты системы, ответственные за управление анимацией, делают снимки системы, соответствующие началу и концу анимации для быстрого перевода системы в эти состояния. Перевод системы в состояния, соответствующие промежуточным моментам анимации, вообще говоря, возможен только при проигрывании анимации с начала (если пользователь специально не организует управление своей анимацией иным образом).

Однако в процессе проигрывания анимации пользователь может дополнительно преобразовывать сцену (поворачивать, перемещать и т.д.) для того, чтобы более чётко рассматривать динамику изменения тех или иных частей сцены. В настоящее время система SharpEye реализована таким образом, что пользовательские действия, производимые во время анимации, в историю не записываются и теряются при перематке анимации в начало или в конец. При этом необходимый режим функционирования менеджера истории обеспечивается вызовом его соответствующих методов в начале и конце проигрывания анимации.

## 7. ЗАКЛЮЧЕНИЕ

В рамках среды – конструктора систем визуализации SharpEye, разрабатываемой авторами доклада, создана подсистема записи, отката и повтора действий пользователя. Данная система позволяет отменять неудачные изменения точки зрения, а также сохранять и восстанавливать сессию работы с системой для быстрого восстановления полученного вида отображения. В рамках технологии сохранения и восстановления сцены также может быть реализована удалённая визуализация.

При проектировании и разработке подсистемы истории авторам пришлось решить ряд проблем идеологического и программного характера (в том числе, для оптимизации работы данной подсистемы). В настоящее время подсистема истории полностью реализована и функциональна в соответствии с изложенными требованиями, позволяющими



удобно использовать её как через пользовательский, так и через программный интерфейс.

## **8. СПИСОК ЛИТЕРАТУРЫ**

- [1] Cypher A. (Ed.), Watch What I Do. Programming by Demonstration. MIT Press, Cambridge, MA. 1993.
- [2] Авербух В.Л., Визуализация программного обеспечения. Конспект лекций. Екатеринбург: мат.-мех. ф-т, УрГУ. 1995.
- [3] Васёв П.А., Кумков С.С., Шмаков Е.Ю., Конструктор специализированных систем визуализации // Научная визуализация, 2012, Т. 4, № 2, С. 64–77. МИФИ. ISSN 2079-3537. <http://sv-journal.com/2012-2/05.php?lang=ru>.
- [4] Васёв П.А., Кумков С.С., Шмаков Е.Ю., О создании среды разработки систем научной визуализации // Труды XIII Международного семинара «Супервычисления и математическое моделирование» (3–7 октября 2011 г.) под редакцией Р.М. Шагалиева. — ИПК ФГУП «РФЯЦ-ВНИИЭФ», г. Саров, 2012. С. 131–140.

# Алгоритм реализации скользящего стерео-курсора

Александра Афанасьева, Алексей Игнатенко  
Факультет вычислительной математики и кибернетики  
Московский Государственный Университет имени М.В.Ломоносова, Москва, Россия  
{afedorova, ignatenko}@graphics.cs.msu.ru

## Аннотация

Быстро развивающиеся технологии работы со стерео-изображениями предполагают использование новых инструментов для просмотра и обработки стереоизображений. В настоящей работе рассматривается алгоритм реализации скользящего стерео-курсора в реальном времени.

Будучи встроенным в систему работы со стерео-изображениями, скользящий курсор может быть использован как для обозначения границ объекта стереопары (например, при совместной работе с изображениями или при построении геометрической модели объекта вручную). Алгоритм также может быть использован в качестве базы для построения карт глубины.

Скользящий стерео-курсор может быть полезен, если мы имеем дело со сложными полупрозрачными объектами на стереопаре, когда установить глубину объекта бывает порой затруднительно. В данном случае использование стереомонитора и соответствующего интерфейса позволит улучшить восприятие и понимание пользователем взаимного расположения объектов.

Технологии, использующие алгоритмы позиционирования стерео-курсора, могут применяться в некоторых областях виртуальной реальности ([4]).

**Ключевые слова:** стерео-курсор, скользящий курсор, корреляция, диспаратет, карта диспаратетов, карта глубины, карта ошибок.

## 1. ВВЕДЕНИЕ

При работе со стерео-интерфейсом важны такие факторы, как форма курсора, его размер, прозрачность, а также алгоритм позиционирования.

Значительную роль при работе со стерео играет форма курсора. В работе [6] проводится моделирование наклона курсора в зависимости от того, в каком положении находится виртуальная «кисть» в руке пользователя. При работе с объектами виртуальной реальности и визуальном отделении их от окружающего пространства возможно использование объемного полупрозрачного курсора (см. [8]).

В ряде задач (например, при построении геометрической модели объекта вручную) требуется вручную выставлять глубину объектов стереопары. При работе со сложными изображениями полупрозрачных объектов восприятие глубины объектов ухудшается. Для упрощения работы необходим инструмент, который при работе с объектом выставлял бы стерео-курсор на корректную глубину (глубину, равную глубине поверхности размечаемого объекта). Таким образом, в настоящей работе мы будем рассматривать алгоритм позиционирования стерео-курсора на поверхность объектов стереоизображения.

*Скользящий стерео курсор – это стерео курсор, который визуально устанавливается на поверхность объекта изображения с текущими экранными координатами при перемещении мыши в плоскости экрана.*

Изображение курсора складывается из двух изображений курсора на каждом из изображений стереопары (соответственно, на каждом изображении курсор имеет свои координаты).

В силу особенностей зрения человека для позиционирования курсора на поверхность объекта нам потребуется вычислять стерео-соответствие в некоторой области вокруг курсора. Один из курсоров будет считаться *ведущим*. Ведущий курсор может быть как правым, так и левым, в зависимости от восприятия результата пользователем. Положение ведущего курсора берется как заведомо правильное и соответствует текущим координатам курсора на изображении, курсор на втором изображении стереопары устанавливается в точку, соответствующую точке объекта, на которую указывает ведущий курсор.

## 2. СУЩЕСТВУЮЩИЕ ПОДХОДЫ

Существует несколько подходов к вычислению стерео-соответствия:

- 1) способы, основанные на подборе окна корреляции,
- 2) способы, связанные с поиском, локализацией и фиксацией взаимного расположения локальных особенностей на изображении,
- 3) способы, базирующиеся на построении карт диспаратетов и карт глубины для всего изображения.

(Подобную классификацию можно обнаружить в работе [2].) Первая группа способов основывается на рассмотрении некоторой области изображения вокруг точки, которой ищется соответствие. Данный подход прост в реализации и дает высокую скорость работы алгоритма.

Вторая группа способов основана на поиске локальных особенностей на изображении и на использовании геометрии взаимного расположения найденных особенностей. Использование исключительно данного подхода может быть ненадежным на ряде изображений, на которых локальные особенности могут встречаться редко или вовсе отсутствовать.

Наибольшую надежность дает подход, при котором мы строим карту стерео-соответствий для всей стереопары. В таком случае последующая постобработка полученной карты позволит уменьшить скачки и выбросы диспаратета. В данной работе этот подход использовался для верификации реализованного алгоритма и для определения основных ошибок с целью последующего развития.

### 3. АЛГОРИТМ ПОИСКА СТЕРЕО-СООТВЕТСТВИЯ

#### 3.1 Метод корреляции

##### 3.1.1 Метод простой корреляции

В работе реализован алгоритм поиска стерео-соответствий на изображениях на основе метода корреляционного окна. Корреляционным окном называют прямоугольную область на изображениях с фиксированными размерами. Путем сравнения данных областей на изображениях устанавливается стерео-соответствие. В настоящей реализации параметры корреляционного окна выбираются, исходя из размеров исходного изображения.

Если брать положение ведущего курсора на одном из изображений как  $(x, y)$ , то положение соответствующей ему точки на втором изображении будет  $(x+dx, y)$  (в силу того, что мы видим объект с обоих глаз на одной и той же высоте), и задача сводится к нахождению диспаратета  $dx$ , который может быть как положительным, так и отрицательным.

Сравнение фрагментов изображений осуществляется по метрике L2. Количество сравнений сильно сокращается за счет поиска по горизонтали.

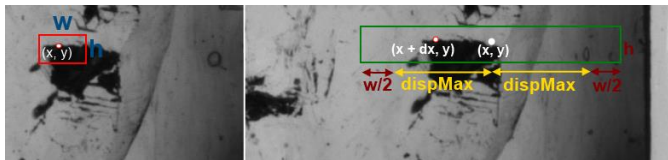


Рисунок 1: Корреляционное окно. Максимальный разброс диспаратета.

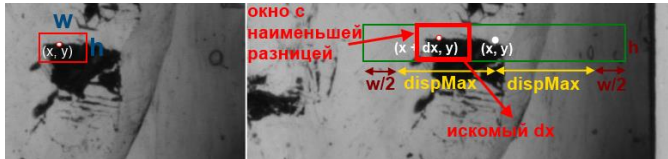


Рисунок 2: Наиболее подходящее для соответствия корреляционное окно и соответствующий ему диспаратет.

В текущей реализации в качестве дополнительных входных параметров могут задаваться размер корреляционного окна и максимально возможный разброс диспаратета.

##### 3.1.2 Предварительный поиск по уменьшенному изображению

С целью улучшить качество результата и увеличить скорость работы диспаратет сначала вычисляется для уменьшенных копий изображений, а затем уточняется на оригинальном изображении. Поскольку окно на уменьшенном изображении содержит больше информации, чем на исходном, значительно сокращаются ошибочные скачки диспаратета.

#### 3.2 Оценка надежности

В процессе сравнения окон строится график корреляции, по которому можно судить о достоверности результата.

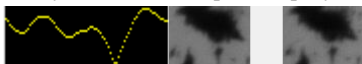


Рисунок 3: Пример достоверного графика корреляции (имеется ярко выраженный минимум).

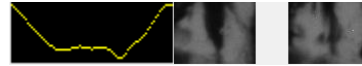


Рисунок 4: Пример недостоверного графика корреляции (в силу того, что объект выглядит по-разному на изображениях стереопары, ярко выраженный минимум отсутствует).

Диспаратет также вычисляется не только в текущем пикселе, но и в соседних пикселях (в настоящей реализации они располагаются на расстоянии одной десятой части от размера окна). Для каждого пикселя-соседа строится график корреляции и находятся локальные минимумы, после чего выбирается минимум, присутствующий на графиках чаще остальных.

Каждый график корреляции домножается на весовую функцию (линейную либо экспоненциальную), что позволяет сократить разброс диспаратета и повысить вклад в конечный результат локальных минимумов на графике, соответствующих небольшому значению диспаратета.

Для оценки надежности результата работы алгоритма была введена функция надежности, показывающая «уверенность» работы алгоритма в конкретной точке изображения. При вычислении данной функции использовалось значение абсолютного и всех локальных минимумов графика корреляции:

$$\mu = 1 - \frac{abs\_min}{\sum_i local\_min_i}$$

$\mu$  - функция «уверенности» алгоритма в точке.

### 4. РЕЗУЛЬТАТЫ

На рисунках 5, 6 и 7 приведены результаты работы алгоритма на различных входных данных. Красным отмечены соответствия на изображениях.

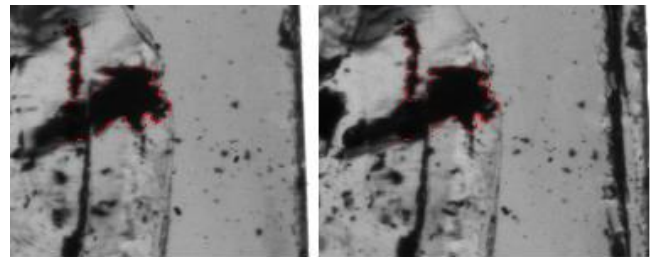


Рисунок 5: Пример работы алгоритма на четких контурах внутри полупрозрачного объекта.



Рисунок 6: Пример работы алгоритма на четких контурах непрозрачного объекта.

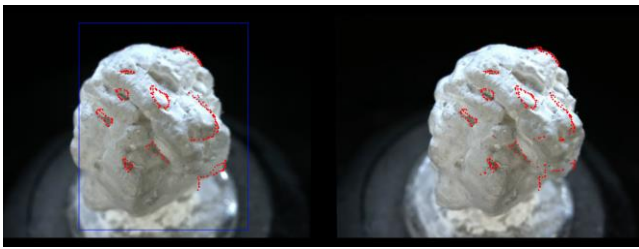


Рисунок 7: Пример работы алгоритма на рельефе равномерно освещенного объекта.

## 5. ВЕРИФИКАЦИЯ

### 5.1 Реконструкция одного из изображений стереопары

«Складывая» карту диспаратетов с правым изображением стереопары, можно реконструировать левое изображение стереопары. (Под «сложением» в данном контексте подразумевается сдвиг каждого пикселя правого изображения на вычисленный в данной точке диспаратет.)

Сравнивая его с исходным, мы получаем области с ошибочно посчитанным диспаратетом.

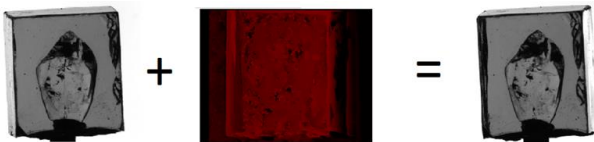


Рисунок 8: Реконструкция левого изображения стереопары путем сдвига пикселей правого изображения стереопары соответственно карте диспаратетов.

### 5.2 Верификация на основе построения карт глубины

Помимо ручного тестирования, в ходе работы была проведена верификация алгоритма на синтезированных данных с заранее известной геометрией сцены.

На начальном этапе было проведено построение карт диспаратета на наборе входных данных. По данным картам, с учетом известной геометрии сцены и характеристик камеры, были восстановлены карты глубины. Полученные карты сравнивались с заведомо верными картами, синтезированными в процессе трассировки лучей.

В результате проведенного сравнения было выяснено, что в областях изображения, соответствующего четким границам непрозрачных объектов, разница между восстановленной и корректной картами глубины была минимальна, что свидетельствует о корректности работы алгоритма в данных точках.

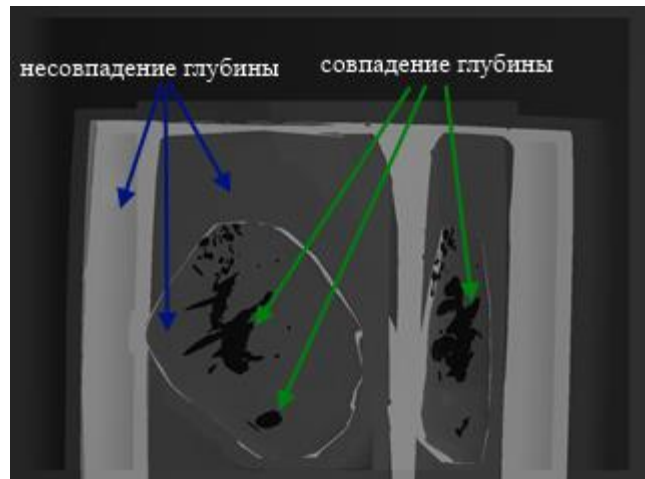


Рисунок 9: Разность между настоящей и восстановленной картами глубины.

По результатам верификации можно сделать вывод о том, насколько приемлимыми являются результаты работы алгоритма, а также построить базу фрагментов изображений, для которых соответствие находится неверно.

## 6. ВЫВОДЫ

Алгоритм дает мало ошибок на четких границах объектов. Основной выигрыш в качестве работы достигается за счет предварительного поиска на уменьшенном изображении, а также за счет домножения графика корреляции на весовую функцию. Учет значений диспаратета в соседних пикселях позволяет в несколько раз сократить число скачков диспаратета, которые сильно снижают удобство работы со скользящим курсором.

Ошибки возможны в случае, если на изображении присутствует ряд регулярных особенностей (решетки, узоры, сетка), а также в случае различного взаимного расположения объектов на изображении, сравнимых по размеру с частью корреляционного окна.

## 7. ИСТОЧНИКИ

- [1] T. Grossman and R. Balakrishnan. *Pointing at trivariate targets in 3D environments*. CHI 2004, pp. 447-454.
- [2] Kuhn, M., Electr. Eng.; Moser, S.; Isler, O.; Gurkaynak, F.K.; Burg, A.; Felber, N.; Kaeslin, H.; Fichtner, Wolfgang. *Efficient ASIC implementation of a real-time depth mapping stereo vision system*. Circuits and Systems, 2003 IEEE 46th Midwest Symposium on. 27-30 Dec. 2003. Page(s): 1478 – 1481 Vol. 3.
- [3] Robert J. Teather, Wolfgang Stuerzlinger. *Cursors for 3D Pointing*. CHI'12, May 5–10, 2012, Austin, Texas, USA.
- [4] Robert J. Teather, Wolfgang Stuerzlinger. *Investigating One-Eyed and Stereo Cursors for 3D Pointing Tasks*. Dept. of Computer Science & Engineering, York University.
- [5] Robert J. Teather, Wolfgang Stuerzlinger. *Pointing at 3D Target Projections with One-Eyed and Stereo Cursors*.
- [6] Feng Tian, Xiang Ao, Hongan Wang, Vidya Setlur, Guozhong Dai. *The tilt cursor: enhancing stimulus-response compatibility by providing 3d orientation cue of pen*. ACM

New York, NY, USA ©2007. Proceeding CHI '07  
Proceedings of the SIGCHI Conference on Human Factors in  
Computing Systems. Pages 303-306.

- [7] C. Ware and K. Lowther. *Selection using a one-eyed cursor in a fish tank VR environment*. ACM TOCHI, 4 (4), 1997, pp. 309-322.
- [8] S. Zhai, W. Buxton, and P. Milgram. *The "silk cursor": Investigating transparency for 3D target acquisition*. CHI 1994, pp. 459-464.

## **Об авторах**

Александра Афанасьева – аспирантка ВМК МГУ имени М.В. Ломоносова. E-mail: [afedorova@graphics.cs.msu.ru](mailto:afedorova@graphics.cs.msu.ru).  
Алексей Викторович Игнатенко – доцент ВМК МГУ имени М.В. Ломоносова. E-mail: [jgnatenko@graphics.cs.msu.ru](mailto:jgnatenko@graphics.cs.msu.ru).

# Получение оверлеев векторных данных большого объёма

И.А.Матвеев<sup>1</sup>, И.А.Юдин<sup>2</sup><sup>1</sup>Вычислительный центр им. А.А. Дородницына Российской академии наук, Москва, Россия<sup>2</sup>Научно-исследовательский институт аэрокосмического мониторинга «АЭРОКОСМОС», Москва, Россия  
matveev@ccas.ru

## Аннотация

Рассмотрена задача построения оверлеев (пересечения, объединения, разности) векторных данных, содержащих большое число контуров простой структуры. С целью решения этой задачи изучены представленные в литературе методы. Как оказалось, лишь три метода из множества имеющихся могут претендовать на решение этой задачи за разумное время. При более детальной изучении и тестировании имеющихся реализаций этих методов лишь одна из них (Vatti/GPC) оказалась пригодной, причём только после внесения существенных доработок. Итоговая улучшенная реализация алгоритма апробирована на массиве данных по пожарам и пригодна для решения практически важных задач.

**Ключевые слова:** оверлеи многоугольников, скан-линейный алгоритм, шейпфайл.

## 1. ВВЕДЕНИЕ

В различных приложениях, связанных с обработкой данных дистанционного зондирования Земли (ДЗЗ), появляется задача вычисления объединения, пересечения, разности множеств, представляющих собой площади земной поверхности, заданные в векторном виде, а именно как набор замкнутых многоугольников, описанных последовательно-стями вершин [1]. Типичными задачами являются: объединение площадей, полученных при обработке данных последовательных пролётов спутников над заданным участком; получение пересечения такой объединённой площади (имеющей обычно весьма сложную форму) с границами административных образований или областями определённого типа растительности; вычисление разности двух областей с целью определения добавленных участков.

## 2. ПОСТАНОВКА ЗАДАЧИ

Области (называемые также *операндами*) состоят каждая из многих (до десятков тысяч) контуров:

$$A = \bigcup_{i=1}^N A_i, B = \bigcup_{j=1}^M B_j, N, M \in [1; 10000).$$

Каждый контур является многоугольником (полигоном) и представлен последовательностью своих вершин, число вершин обычно невелико (порядка десяти), но может достигать нескольких тысяч:

$$A_i = \{a_{ik}\}_{k=1}^{N_i} = \{(x_{ik}, y_{ik})^T\}_{k=1}^{N_i}, N_i \in [3; 1000).$$

Контуров невырожденные, не имеют самопересечений, однако контуры одного аргумента могут взаимно пересекаться.

Общее число точек во всех контурах может достигать сотен тысяч:

$$\sum_{i=1}^N N_i \in [3; 100000).$$

Требуется построить алгоритмы, вычисляющие объединение  $A \cup B$ , пересечение  $A \cap B$ , разность  $A \setminus B$  таких множеств.

Подобные задачи решались многими исследователями при помощи различных методов. В [4] проводится обзор алгоритмов, предложенных Sutherland, O'Rourke, Weiler, Леонов, Schutte, Holwerda, Margalit и сравнение их по ряду таких параметров, как вычислительная устойчивость, ресурсоёмкость, скорость работы, простота программной реализации. Кроме этого, авторами рассматривались также алгоритмы Liang-Barsky, Maillot, Vatti, Greiner-Horman, рекомендованные в [5]. Однако, несмотря на кажущееся обилие методов, большая часть из них оказалась непригодной из-за большой размерности задачи. Многие из перечисленных методов рассчитаны на работу с очень сложными, но малочисленными контурами, в то время как здесь имеется прямо противоположная ситуация: большое число относительно простых контуров.

Алгоритмы Сазерленда, О'Рурка, Шутте, Лианга-Барского и Maillot признаны непригодными, так как требуют отсутствия "дыр" и/или выпуклости хотя бы одного из операндов. Алгоритм Вейлера-Азертонна пригоден для операции лишь с двумя (пусть и очень сложными) контурами, а алгоритм Маргалита-Кнотта пригоден лишь в случае когда один из операндов представлен единственным многоугольником. Поскольку число многоугольников может составлять десятки тысяч (хоть и очень простых), для операций в нашем случае потребуются десятки тысяч вызовов этих алгоритмов с последующим объединением результатов; причём оптимальное объединение результатов само по себе является непростой задачей. Алгоритм Холверда является предшественником более быстрого и стабильного алгоритма Ватти, а алгоритм Леонова - модификацией алгоритма Грейнера-Хормана. Таким образом, более подробному разбору, в том числе экспериментальной проверке, подверглись три алгоритма: триангуляции [3], Леонова [2] и Ватти [9].

## 3. АНАЛИЗ ТРЁХ АЛГОРИТМОВ

*Триангуляцией* многоугольника называется его представление в виде полного набора взаимно не пересекающихся треугольников. Основная идея алгоритма построения оверлеев (т. е. пересечения, объединения или разности) многоугольников с помощью триангуляции заключается в построении совместной триангуляции исходных многоугольников (при которой структурными линиями

служат стороны обоих аргументов), а затем выбора из множества полученных треугольников некоторых, согласно выполняемой операции и принадлежности каждого из треугольников исходным множествам-аргументам. Избранные треугольники можно наконец объединить в многоугольники, хотя для рассматриваемой задачи это не обязательно.

*Алгоритм Леонова* (Грейнера-Хормана) состоит в поиске и маркировке всех пар пересекающихся рёбер, после чего проводится трассировка, начинающаяся с любой вершины одного из контуров, выделяющая все минимальные ограничивающие контуры областей, полученных наложением аргументов. Каждый из контуров маркируется согласно выполняемой операции и принадлежности аргументам, после чего проводится сборка этих контуров в результирующее множество.

*Алгоритм Ватти* также называется скан-линейным. Его основная идея состоит в том, что все вершины аргументов упорядочиваются по значению их ординат, и каждая из них порождает горизонтальную скан-линию. Далее при последовательном просмотре скан-линий производится анализ их пересечений с точками и рёбрами аргументов, поддерживается список контуров, в который вносятся изменения, соответствующие характеру и последовательности пересечения очередной скан-линии с вершинами и сторонами.

Поскольку все эти алгоритмы достаточно сложны, в качестве основы были взяты программные реализации, разработанные ранее. Для построения триангуляции и для выполнения операций над треугольниками были использованы алгоритмы из библиотеки Computational Geometry Algorithms Library [6]. Алгоритм Леонова был взят из [7]. За основу реализации алгоритма Ватти был взят код [8].

Для первого этапа тестов были использованы данные о пожарах, полученные в 2011 году. Было использовано 3906 векторных файлов, каждый из которых содержит от одного до нескольких тысяч полигонов простой формы (обычно это четырёхугольники), являющихся фактически векторным представлением пикселей растрового изображения, на которых детектирован пожар. Файлы были разбиты на группы по месяцам. Для тестов алгоритмов на первом этапе была использована *подгруппа* файлов, имеющих общий размер 100 килобайт. Также была сформирована группа, включающая в себя все файлы всех месяцев. В Таблице 1 приводятся характеристики этих групп.

Таблица 1. Характеристики тестовых групп шейпфайлов

Месяц	Число файлов	Общий размер файлов, МБ	Общее число контуров	Общее число вершин
Апрель	470	3,1	20634	102389
Май	1231	11,2	69560	405416
Июнь	1322	11,6	62292	472804
Июль	397	2,2	16193	64772
Сентябрь	268	1,8	13298	53192
Октябрь	218	1,5	8938	35752
<i>Подгруппа</i>	8	0.1	608	3899

Все месяцы	3906	31,1	190915	1134325
------------	------	------	--------	---------

Тесты всех трёх алгоритмов проводились только для "Подгруппы", результаты представлены в Таблице 2.

Таблица 2. Результаты обработки тестовой подгруппы файлов тремя wybranymi алгоритмами

Алгоритм	Время исполнения, с	Число контуров	Число вершин
Триангул.	0,038	1314	3942
Леонова	0,11	596	3880
Ватти	0,017	598	3883

Имеющаяся реализация алгоритма Леонова не прошла по временным ограничениям, кроме того, в этой реализации при вычислениях производится перевод координат из чисел с плавающей точкой в целочисленные 64-битные значения и обратно. По этой причине координаты точек не сохраняются строго, что может вызвать проблемы при дальнейшей обработке. По этим причинам было принято решение отказаться от алгоритма Леонова.

Что касается триангуляционного алгоритма, то при переходе к расчётам на реальных данных (после вполне благополучных тестов на "Подгруппе") обнаружился его существенный недостаток. На реальных данных один и тот же пожар может быть детектирован десятки раз. Соответственно, десятки файлов содержат контур пожара в одном и том же месте. На Рис.1. приводится пример объединения двух прямоугольных контуров (каждый из которых может быть представлен как два треугольника), в результате которого при использовании метода триангуляции получается 10 треугольников, и объединение трёх контуров, в результате которого получается уже 31 треугольник.

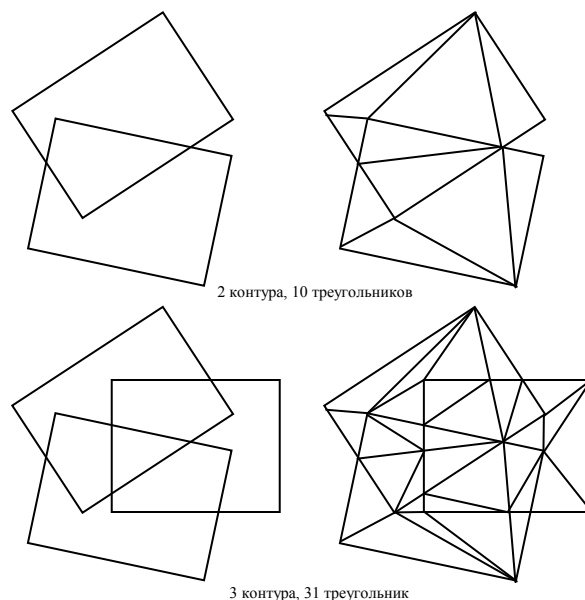


Рис 1: Пример нарастания сложности триангуляции.

В общем случае при объединении наложенных контуров количество треугольников растёт квадратично от числа объединений. Таким образом, на реальных данных, содержащих области с многократным перекрытием число треугольников резко возрастает, а сами они уменьшаются до

ничтожных площадей. Обратное объединение треугольников в некоторую фигуру само по себе является трудной задачей, требующей существенной доработки алгоритма триангуляции. По этой причине было принято решение отказаться от метода триангуляции.

#### 4. ДОРАБОТКИ АЛГОРИТМА ВАТТИ

За основу реализации алгоритма Ватти был взята библиотека оверлеев [GPC]. Оригинальный код этой библиотеки оказался непригоден для решения в полном объёме задач, поставленных в проекте, и был переработан. были сделаны следующие существенные доработки: исключена рекурсия при перемещении по спискам (деревьям) вершин, активных рёбер, фигур; исключено задание в явном виде таблицы попарных пересечений многогранников; учтена возможность пересечения фигур, находящихся в одном файле. Опишем каждую из этих доработок более подробно.

##### 4.1 Рекурсия при перемещении по спискам

Алгоритм Ватти оперирует с несколькими списками объектов, это список вершин (составляемый из вершин исходных фигур, пополняется в процессе работы пересечениями рёбер), список активных рёбер, списки активных фигур и построенных фигур. В версии GPC эти списки реализованы как бинарные деревья, для оптимизации поиска. Однако, перемещение по спискам в исходном коде было реализовано в виде рекурсивного вызова функций обработки элемента. При работе с большими объёмами данных степени самых удалённых от корня вершин деревьев возрастают до нескольких десятков тысяч. При использовании рекурсивных вызовов для перемещения по дереву это означает вызов функции самой из себя несколько десятков тысяч раз. По этой причине при обработке больших групп файлов (например "Май" и "Июнь") происходило исчерпание стека вызовов и аварийное завершение алгоритма и самой программы.

Были сделаны доработки с целью исключить рекурсию. В переработанной версии алгоритма глубина вызовов функций не превышает 5, рекурсия исключена.

##### 4.2 Таблица попарных пересечений

В исходной версии алгоритма для минимизации вычислений алгоритм применён простой метод уменьшения количества контуров, для которых используется скан-линейный метод. Идея заключается в том, что каждый из аргументов  $A$  и  $B$  представлен множеством отдельных контуров, но при этом далеко не каждый контур  $A$  пересекает хотя бы один контур  $B$  и наоборот. Таким образом каждый операнд можно разбить на два подмножества, *активное* и *пассивное*:

$$A = A_{active} + A_{passive}, \quad B = B_{active} + B_{passive}.$$

Пассивные части операндов исключаются из выполнения скан-линейной процедуры и могут быть добавлены уже потом к результату:

$$A \cap B = A_{active} \cap B_{active}$$

$$A \setminus B = (A_{active} \setminus B_{active}) + A_{passive}$$

$$A \cup B = (A_{active} \cup B_{active}) + A_{passive} + B_{passive}$$

Однако, реализация этого подхода выполнена, очевидно, с расчётом на небольшое количество контуров. В частности, для получения списков пассивных и активных контуров используется массив чисел, имеющий размер, равный произведению числа контуров двух операндов. Для решаемых в рамках проекта задач оба эти числа могут превышать 100000, что приводит к невозможности размещения в оперативной памяти такого массива.

Была произведена доработка алгоритма, в результате которой необходимый размер массива стал равен не произведению, а сумме числа контуров в операндах.

##### 4.3 Пересечение фигур в одном шейпфайле

Проверка условий включения каждого полученного многогранника в результат осуществляется подсчётом числа пересечений скан-линии с рёбрами многоугольников  $A$  и  $B$ . С этим связана особенность алгоритма, которая потребовала дополнительных усилий для его использования, а именно, при наличии пересекающихся фигур в каком-либо операнде (например  $A$ , т.е.  $A_i \cap A_j \neq \emptyset$ ) алгоритм исключает их пересечение из результата. Пример на рисунке ниже.

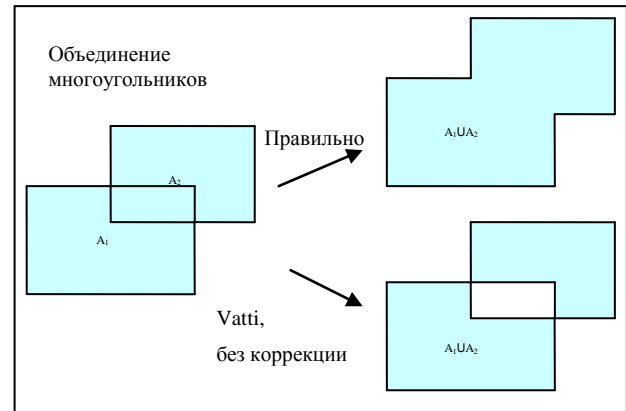


Рис 2: Пример работы алгоритма Vatti при условии наличия пересекающихся фигур в одном операнде

Была сделана надстройка над алгоритмом. Каждый из операндов (пусть это будет  $A$ ) предварительно разбивался на группы контуров (далее такая группа называется *слоем*), такие, что внутри каждой такой группы контуры не

пересекались: 
$$A = \sum_{g=1}^G A^{(g)}, \quad A^{(g)} = \sum_{s=1}^{S_g} A_s^{(g)},$$

$$\forall g : A_s^{(g)} \cap A_t^{(g)} = \emptyset \Leftrightarrow s \neq t. \quad \text{Далее слои}$$

$A^{(g)}, g = G \dots 1$  последовательно объединяются.

Для разбиения на слои разработан специальный алгоритм, использующий проецирование фигур (точнее, их ограничивающих прямоугольников) на вспомогательный растр. Работа алгоритма представлена блок-схемой на Рис.4. В результате исполнения алгоритма получается несколько слоёв, в каждом из которых содержатся взаимно непересекающиеся контуры. Как правило, слои с большим порядковым номером содержат меньшее количество контуров. Поэтому слои объединяются начиная с меньших



номеров, с целью уменьшения количества вычислений алгоритмом Ватти.

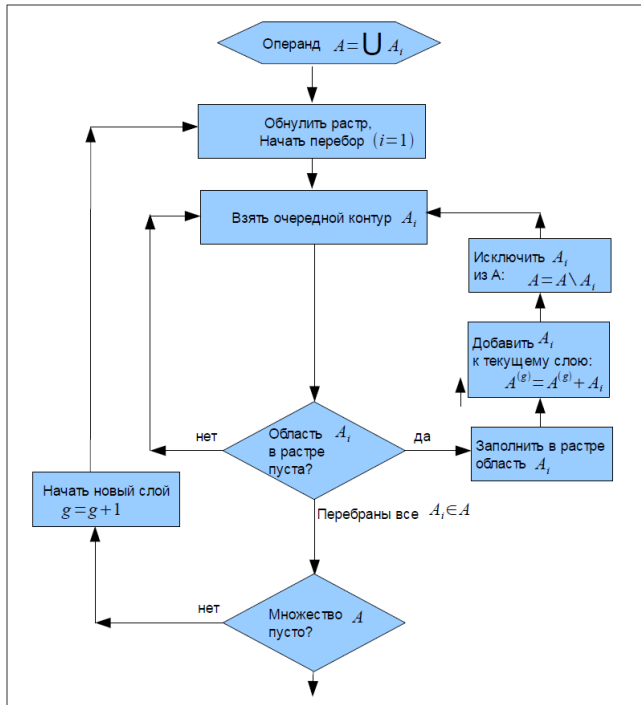


Рис 3: Алгоритм разбиения на слой

## 5. РЕЗУЛЬТАТЫ ТЕСТОВ

Результаты тестов алгоритма на множествах Таблицы 1 приведены в Таблице 3. Проводилась операция объединения всех контуров всех файлов в группе в один набор контуров.

Таблица 3. Результаты тестов алгоритма Ватти

Группа	Время исполнен., с	Число слоёв	Число контуров результата	Число вершин результата
Апрель	6	43	12495	91783
Май	87	134	23103	256568
Июнь	143	168	31353	321899
Июль	7	92	6462	45523
Сентябрь	4	66	6754	41970
Октябрь	4	38	5394	30344
Все месяцы	751	419	78234	730128

## 6. ЗАКЛЮЧЕНИЕ

В целях автоматизации построения оверлеев при обработке данных дистанционного зондирования Земли были изучены имеющиеся методы и их программные реализации. Лишь один из имеющихся методов оказался по сути пригоден для решения поставленных задач, причём только после значительных доработок. Сделано несколько доработок программной реализации алгоритма, позволивших существенно расширить область его применимости и размерность обрабатываемых им данных. Сделана надстройка

над алгоритмом, позволяющая обойти один из его принципиальных недостатков, некорректную обработку аргументов со взаимно пересекающимися контурами. Полученный в итоге алгоритм пригоден для решения практических задач и используется в этом качестве.

В процессе работы выяснились следующие недостатки:

Имеющаяся реализация алгоритма Ватти обладает алгоритмической сложностью  $O(n\sqrt{n})$ , где  $n$  - сумма числа точек аргументов. Алгоритм Ватти может быть доработан для оптимизации поиска пересечений рёбер таким образом, что его сложность станет  $O(n \log n)$ .

## 7. AKNOLEDGMENTS

Работа выполнена при финансовой поддержке РФФИ (проект №12-01-91162).

## 8. REFERENCES

- [1] Бондур В.Г. *Актуальность и необходимость космического мониторинга природных пожаров* // Вестник ОНЗ РАН. 2010. Том 2. NZ11001.
- [2] Леонов М.В., Никитин А.Г. *Эффективный алгоритм, реализующий замкнутый набор булевых операций над множествами многоугольников на плоскости* / Препринт Института систем информатики СО РАН № 46. 1997. 20 с. // <http://www.iis.nsk.su/files/preprints/046.pdf>
- [3] Скворцов А.В. *Триангуляция Делоне и ее применение*. Томск: Изд-во Томского ун-та, 2002.-128 с.
- [4] Ченцов О.В., Скворцов А.В. *Обзор алгоритмов построения оверлеев многоугольников* // [www.ict.edu.ru/ft/004461/46.pdf](http://www.ict.edu.ru/ft/004461/46.pdf)
- [5] Agoston M.K. *Computer Graphics and Geometric Modeling* // Springer. 2004. 920 p.
- [6] Computational Geometry Algorithms Library // <http://www.cgal.org/>
- [7] Leonov M. *PolyBoolean* // <http://www.complex-a5.ru/polyboolean/comp.html>
- [8] Murta A. *General Polygon Clipper Library* // <http://www.cs.man.ac.uk/~toby/gpc/>
- [9] Vatti B.R. *A generic solution to polygon clipping* // Communications of the ACM. 1992. V.35. N.7. P.56–63.

## Информация об авторах

Иван Алексеевич Матвеев, зав. сектором ВЦ РАН.

Юдин Илья Антонович, н.с. НИИ аэрокосмического мониторинга «АЭРОКОСМОС», [yudinilya@gmail.com](mailto:yudinilya@gmail.com)

# Компьютерная графика как эффективный инструмент развития современных технологий

А.Г. Волобой, В.А. Галактионов  
Институт прикладной математики им. М.В.Келдыша РАН, Москва

## Аннотация

Компьютерная графика является не только интересной областью исследований и разработок алгоритмов, но и реально применяется в современном высокотехнологичном производстве. Приведены примеры практических задач, которые стало возможно решать именно с помощью методов и алгоритмов реалистичной компьютерной графики.

**Ключевые слова:** системы реалистичной визуализации, моделирование оптических систем, Монте-Карло трассировка лучей

## 1. ВВЕДЕНИЕ

Компьютерная графика изначально зародилась как эффективное и мощное средство связи между человеком и вычислительной машиной. Использование графической формы представления информации, организация диалога между человеком и компьютером с использованием визуальных образов позволили существенно увеличить скорость обработки информации человеком, что привело к повышению эффективности исследований и разработок в самых различных областях науки и техники.

Однако с появлением методов синтеза реалистичных изображений, основанных на физически корректном моделировании распространения света, область применения компьютерной графики существенно расширилась. Созданные алгоритмы и программные средства стали представлять интерес для использования в архитектуре, градостроительстве, проектировании систем освещения, в автомобильной и авиационной промышленности и др.

В докладе приведено несколько примеров практических задач, связанных с построением физически корректных реалистичных изображений и моделированием распространения света в различных средах, с которыми авторы сталкивались в своей работе. Все приведенные ниже изображения были сгенерированы с помощью программных систем компьютерной графики, разработанных в ИПМ им. М.В.Келдыша РАН [1].

## 2. ПРИМЕНЕНИЕ РЕАЛИСТИЧНЫХ ИЗОБРАЖЕНИЙ

**Архитектура и ландшафтный дизайн.** Архитектура была одной из первых областей применения генерации реалистичных изображений виртуальных сцен. Это связано с относительной простотой виртуального представления архитектурных объектов в памяти, что в свою очередь снижает требования к ресурсам компьютера. При сильном упрощении здания могут быть представлены параллелепипедами, стены, этажные перекрытия – плоскостями. Практически отсутствуют криволинейные поверхности.

С помощью компьютерного моделирования заранее возможно определить и согласовать с заказчиком архитектурного проекта не только эстетичность интерьеров помещений и внешнего вида здания, что стало уже традиционным, но и следующие факторы:

- 1) В последние годы важным аспектом строительства любого здания является экономичность его системы освещения. Основываясь на точных расчетах освещения помещений дневным и солнечным светом, архитектор может варьировать положение и ориентацию здания относительно сторон горизонта и окружающих строений, расположение и размеры оконных проемов, лоджий и т.д. Основной целью при таком дизайне является максимальное использование естественного света (создаваемого солнцем и небом) для достижения комфортного освещения помещения при минимальных энергетических затратах на его искусственное освещение.
- 2) При проектировании рабочих и офисных помещений освещенности рабочих мест должны соответствовать существующим стандартам, в которых указаны не только минимально допустимые величины освещенности, но и такие параметры, как, например, доля прямого света от источника в поле видимости человека («комфортность» освещения). Для эффективной проверки соответствия стандартам рассчитанная освещенность может быть визуализирована в виде световой карты. Световая карта – это такое же реалистичное изображение виртуальной сцены, но только физические значения освещенности представлены в ней заданными цветами («псевдо-цветами»). Правильное задание соответствия значений освещения определенному цвету позволяет сразу увидеть соответствие разрабатываемого проекта стандартам. Пример такой световой карты представлен на рис. 1.

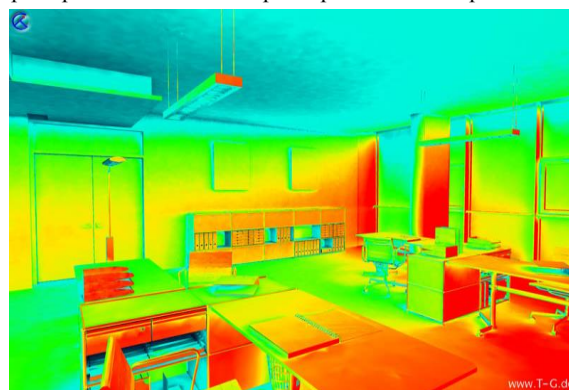


Рис. 1. Распределение освещенности, представленное световой картой.

**Автомобильная промышленность.** Алгоритмы реалистичной визуализации, используемые в автомобильной промышленности, значительно сложнее тех, что сначала были разработаны для архитектурных приложений. Основными дополнительными факторами здесь являются

- наличие большого числа криволинейных поверхностей, обычно представляемых в виде сплайнов в системах САПР;

- сложные оптические свойства поверхностей; для примера достаточно сравнить внешний вид автомобильной краски «металлик», цвет которой меняется в зависимости от направлений освещения и наблюдения, и стен здания, которые в большинстве случаев являются практически диффузными.

Однако и класс задач, в решении которых применяются алгоритмы компьютерной графики, в автомобильной промышленности значительно шире. Приведем некоторые из них.

1) Задача расчета освещения салона автомобиля является схожей с задачей расчета освещения интерьеров помещений. Здесь также важными факторами являются как эффективность освещения, так и соответствие освещенности принятым стандартам.

2) Важным направлением, получившим развитие в последние годы, стало моделирование и визуализация новых оптически сложных материалов, таких как многослойные краски со сложной микроструктурой (типа перламутровых и «металлик»). Разработанные программные средства позволяют моделировать и визуализировать краски, состоящие из плоскопараллельных однородных слоев, каждый из которых составлен из прозрачного лака и взвешенных в нем пигментных частиц и интерференционных чешуек.

На рис. 2 представлен автомобиль, окрашенный такой краской, при солнечном освещении. Цвет этих красок зависит от угла зрения и направления освещения. Он может резко меняться («переключаться») даже при плавном изменении этих углов. Светорассеивающие свойства таких красок не могут быть заданы скалярными коэффициентами, а требуют более сложного описания с помощью двунаправленных функций отражения света. Другим аспектом визуального восприятия краски является освещение, она выглядит различно утром и в середине дня, при солнечной или пасмурной погоде. Гораздо дешевле смоделировать на компьютере внешний вид краски и исследовать ее при различных условиях освещения (дневной, солнечный свет, искусственное освещение), чем произвести опытную партию краски и покрасить ею реальный автомобиль. В настоящее время такой подход к моделированию красок применяется не только в автомобильной промышленности, но и для окраски большого спектра других изделий (мобильные телефоны, бытовая техника и пр.)



Рис. 2. Автомобиль, окрашенный оптически сложной краской, при солнечном освещении.

Результаты моделирования красок оказались столь успешными, что позволили перейти к более сложной задаче моделирования и визуализации покрытий с высокой концентрацией пигментных частиц, например, при разработке новейших типов принтерных чернил. С точки зрения компьютерной графики задача визуализации принтерных чернил выглядит похожей на предыдущую. Но малые размеры частиц и их высокая концентрация полностью меняют вычислительные методы, применимые для данного случая. Вычислительная сложность связана не только с необходимостью использовать методы решения волновых уравнений для моделирования всего слоя, но и с последующим формированием внешнего вида чернил из полученного волнового решения.

3) Важным аспектом безопасности является отсутствие бликов и отражений элементов салона в стеклах автомобиля. На рис. 3 показан пример визуальной оценки нежелательных отражений, которые получаются при моделировании с заданными параметрами материалов и геометрии салона автомобиля.



Рис. 3. Визуальная оценка нежелательных отражений в боковом стекле автомобиля.

4) В современных условиях, когда автомобильный рынок перенасыщен, и каждая фирма вынуждена бороться за покупателя, модификация внешнего вида автомобиля становится практически ежегодной. Одним из наиболее выигрышных элементов являются фары. Поэтому использование инструментов компьютерной графики для разработки и оценки внешнего вида как включенных, так и выключенных фар, в дневное и ночное время позволяет существенно снизить затраты и ускорить разработку очередной модификации автомобиля. При моделировании фар также проверяется на соответствие стандартам направление и яркость получаемого светового луча. Аналогичным образом производится моделирование задних фонарей, стоп-сигналов и поворотных огней. При этом важным является правильная цветопередача красных и желтых фонарей или стекол лампочек.

**Авиационная промышленность.** Как и при визуализации автомобиля, в модели салона самолета мы сталкиваемся с большим числом криволинейных поверхностей, материалов и тканей со сложными оптическими свойствами, которые покрывают значительную часть элементов салона. Однако главной особенностью моделирования освещения салона самолета

является практически полное отсутствие прямого света. Правильно рассчитать освещение здесь возможно только с использованием алгоритмов глобального освещения.

По данным фирмы Boeing разработка и создание внутренней отделки пассажирского самолета составляют до 60% его стоимости. Во многих случаях фирма строит прототип салона в натуральную величину, показывает представителям авиакомпаний-заказчиков, и затем модифицирует в соответствии с замечаниями. И только после утверждения прототипа он будет реализован в воздушном лайнере. Компьютерное моделирование освещения и реалистичная визуализация салона позволяет существенно сократить затраты. На рис. 4 показано изображение салона самолета Airbus A-380, созданное дизайнерами еще во время его начальной разработки.



Рис. 4. Виртуальная модель компоновки и внутренней отделки салона самолета Airbus A-380.

Важным критерием проектирования кабины пилотов является хорошая читаемость приборов и устройств управления при различном освещении, которые можно проверить с помощью системы физически аккуратной, реалистичной визуализации.

Следует подчеркнуть, что все вышеперечисленные задачи можно решать, только если моделирование основано на физических законах распространения света и позволяет достичь приемлемой точности. Или, другими словами, если реалистичное изображение, сгенерированное компьютером по виртуальному описанию, будет соответствовать восприятию реальности человеком. Для решения этих задач мы использовали универсальную и достаточно эффективную технологию двунаправленной Монте-Карло трассировки лучей.

### 3. УЧАСТИЕ В ПРОЕКТЕ МКО ТС.3.33

МКО – Международная комиссия по освещению [2] – является наиболее авторитетной организацией, разрабатывающей и принимающей стандарты и технологии в области цвета и света. По заданию этой организации под руководством Ф. Маатаги были разработаны и в 2002 году опубликованы наборы аналитических тестов для программ моделирования освещения [3]. Всего было предложено 227 тестов.

Тесты были направлены на проверку того, насколько программы моделирования распространения света реально подчиняются законам оптики, в частности, для различных моделей естественного освещения. Тесты отдельно проверяли такие аспекты, как корректность задания источников света, корректное моделирование множественных переотражений и т.д. Особое внимание

было уделено сложной для корректного вычисления модели, когда внутреннее помещение освещается небесной полусферой через небольшое окно. Для этой модели корректные результаты в разумное время можно получить практически только двунаправленной Монте-Карло трассировкой лучей.

Программная система моделирования освещенности, разработанная в ИПМ, участвовала в проверке этими тестами вместе с такими известными продуктами как Dialux, Genelux, Lightscape, Radiance. Результаты независимого тестирования были опубликованы в [4], где наша программная система Inspiger показала в большинстве случаев результат не хуже или превосходящий конкурентов. Более половины тестов показали ошибку менее 1%, и только 2% тестов превысили «инженерно-приемлемую» ошибку 10%.

### 4. МОДЕЛИРОВАНИЕ ОПТИЧЕСКИХ СИСТЕМ

Созданная технология расчета распространения света оказалась столь эффективной и точной, что позволила перейти к моделированию и инженерному проектированию сложных оптических светопроводящих и осветительных систем, таких как жидкокристаллические мониторы, приборные панели автомобилей, светодиоды и осветительные системы на их основе.

Растущие требования, предъявляемые к эргономичности и эффективности современной техники, заставляют производителей искать новые решения при проектировании оптических систем. Так для того чтобы добиться максимально равномерного и комфортного освещения экрана ЖК монитора или приборной панели, создаются сложные системы подсветки, в производстве которых используются высокотехнологичные светопроводящие элементы и устройства со сложными оптическими свойствами. Принцип работы таких устройств основан на эффекте полного внутреннего отражения. Для их моделирования практически невозможно использовать обычные детерминистические методы лучевого расчета распространения света, применяемые для расчета линзовых оптических систем. Здесь также наиболее эффективна Монте-Карло трассировка лучей.

Рассмотрим основные принципы проектирования осветительной части ЖК монитора. Свет, излучаемый люминисцентной лампой или светодиодами, поступает через входную грань в *светопроводящую пластину* (СПП) - основной оптический элемент системы (рис. 5).

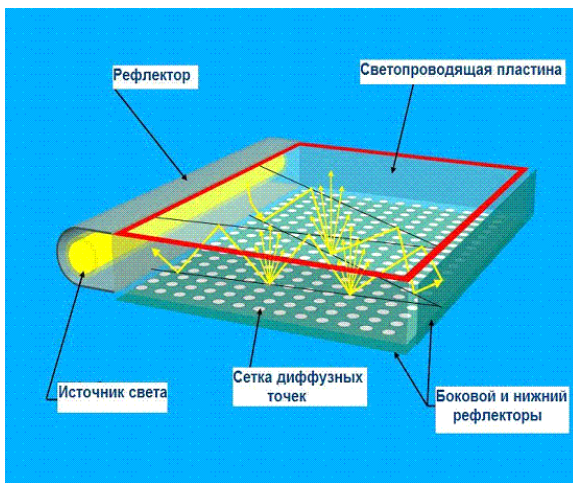


Рис. 5. Схема функционирования системы подсветки жидкокристаллического монитора.

Распространяясь внутри пластины, свет претерпевает многократные (сотни тысяч раз) полные внутренние отражения от ее поверхностей. На нижнюю сторону пластины нанесена сетка диффузных точек, либо микрорельеф, которые рассеивают падающий свет. Свет выходит из пластины через верхнюю выходную грань только в результате диффузного рассеяния от этих точек. Все остальные световые лучи падают на верхнюю поверхность пластины под углами, превышающими угол полного внутреннего отражения, что не дает возможности свету покинуть пластину. Таким образом, интенсивность выходящего светового потока в точке определяется плотностью распределения диффузной сетки (или микрорельефа) под ней и в некоторой ее окрестности.

Конечной целью моделирования подобных систем является расчет пространственного и углового распределений яркости света над выходной гранью устройства (помеченной красной рамкой на рис. 5). Задача заключается в том, чтобы сделать яркость на выходе максимально равномерной по всей выходной грани. Однако по мере удаления от входной грани пластины (у которой находятся лампа и рефлектор) световой поток ослабевает. Для компенсации этого эффекта плотность диффузной сетки (микрорельефа) должна постепенно возрастать слева направо.

Выше был описан простой пример моделирования системы подсветки. В реальности все обстоит гораздо сложнее. В частности, применение диффузных отражающих точек не эффективно за счет поглощения света на красителе. Типичный процесс проектирования подобных устройств включает в себя:

- проектирование источников света (светодиодов) и рефлекторов для получения максимального светового потока на входе в светопроводящую пластину (СПП);
- проектирование структуры микрогеометрии на нижней, рассеивающей, части СПП для получения пространственно равномерного распределения света над ее выходной (верхней) гранью;
- проектирование нескольких призматических слоев (BEF) для концентрации и усиления светового потока на выходе из СПП;
- проектирование диффузного и/или поляризационного фильтра (DBEF) для еще

большого усиления яркости экрана в определенном направлении.

Аналогичные принципы используются при проектировании таких устройств, как приборная доска автомобиля, кнопки с подсветкой и т.д. Формы СПП для разных устройств бывают достаточно сложными, и производится также моделирование их оптимальной формы. Дополнительно нами были успешно разработаны комплексы автоматического (без участия человека) дизайна таких оптических систем [5].

Светодиоды являются в последнее время наиболее эффективными и массовыми источниками света. Это порождает ряд задач их проектирования и использования. До сих пор актуальна задача создания белого светодиода, так как изначально он излучает свет в синем диапазоне. Одна из существующих технологий – использование флуоресцентного материала для коррекции цвета. При помощи моделирования можно добиться оптимальной концентрации флуоресцентного вещества. Пример такого дизайна приведен на рис. 6.

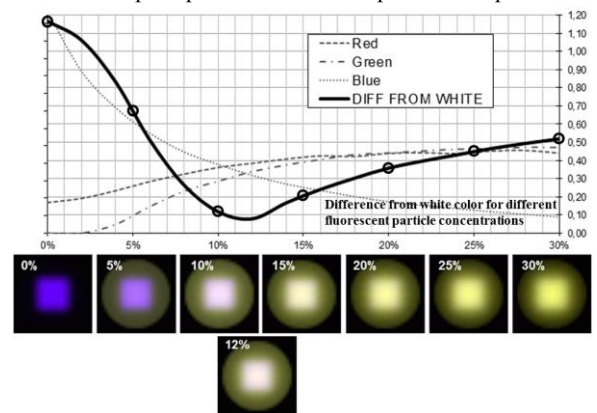


Рис. 6. Зависимость итогового цвета от концентрации флуоресцентного вещества.

При проектировании осветительных приборов на основе светодиодов дополнительной сложностью являются маленькая площадь и сила свечения одного светодиода. Для достижения требуемых значений освещенности часто необходимо использовать большие массивы светодиодов, вводить в проектируемые устройства такие оптические элементы, как концентрирующие линзы.

## 5. ЗАКЛЮЧЕНИЕ

Бурное развитие промышленных технологий, появление многочисленных новых материалов с уникальными свойствами, прогресс в вычислительной технике, наблюдаемые в последнее десятилетие, существенно изменили и дополнили классическое понимание компьютерной графики как эффективного средства взаимодействия между человеком и вычислительной машиной, построенного на основе визуальных образов. Технологии компьютерной графики, ранее применявшиеся при построении изображений, оказались широко востребованными в компьютерном моделировании и инженерном проектировании высокотехнологичных промышленных изделий, разработке энергоэффективных осветительных устройств, киноиндустрии и других областях. При этом требования к реалистичности изображений, их физической достоверности, скорости генерации, точности вычислений значительно возросли.

Работа поддержана грантами РФФИ № 11-01-00870, 12-01-00560, 13-01-00454, а также фирмой Integra Inc.

## 6. ЛИТЕРАТУРА

- [1] А.Г. Волобой, В.А. Галактионов. Машинная графика в задачах автоматизированного проектирования // «Информационные технологии в проектировании и производстве», № 1, 2006, с. 64-73.
- [2] CIE – International Commission on Illumination, <http://www.cie.co.at/cie/index.html>
- [3] F. Maamari. TC.3.33. List of proposed test cases // ENTPE – France, 2002.
- [4] F. Maamari, M. Fontoynt, M. Hirata, J. Koster, C. Marty, A. Transgrassoulis. Reliable Datasets for Lighting Programs Validation, Benchmark Results // Proceedings of CISBAT 2003, EPFL Lausanne, 2003, pp. 241-246.
- [5] D.Zhdanov, A. Garbul, V. Mayorov, V.Sokolov, I.Potemin, T.Hyodo, A.Voloboy, V.Galaktionov. Automatic Design of Illumination Systems // Optical Review, vol. 20, no.2, 2013, p.155-159.

## Abstract

The paper demonstrates that computer graphics is not only the area of research and development of algorithms but also useful and productive tool for high technology products development. The examples of practical tasks which were solved with the aid of methods and algorithms of realistic physically accurate computer graphics are presented.

**Keywords:** *realistic rendering software, simulation of optical systems, Monte Carlo ray tracing*

## Authors:

Алексей Геннадьевич Волобой, д.ф.-м.н., с.н.с.,  
E-mail: [voloboy@gin.keldysh.ru](mailto:voloboy@gin.keldysh.ru)

Владимир Александрович Галактионов, д.ф.-м.н., проф.,  
зав. отделом Компьютерной графики и вычислительной оптики, E-mail: [vlgal@gin.keldysh.ru](mailto:vlgal@gin.keldysh.ru)

# Физически корректный алгоритм рендеринга анизотропных поглощающих кристаллов

В.А. Дебелов, Д.С. Козлов

Институт вычислительной математики и математической геофизики СО РАН,  
Новосибирский государственный университет, Новосибирск, Россия  
{debelov, kozlov}@oapmg.ssc.ru

## Аннотация

Данная работа посвящена задаче фотореалистической визуализации (рендерингу) поглощающих оптически изотропных и анизотропных кристаллов (сред), а именно расчёту поглощения света, в процессе его распространения внутри среды.

В статье излагается локальная модель, позволяющая рассчитать изменение интенсивности луча, в процессе его распространения внутри изотропной, одноосной или двуосной среды. На основе предложенной модели поглощения света разработан алгоритм физически корректного рендеринга анизотропных поглощающих кристаллов.

Рассматриваемая модель предназначена для интеграции в алгоритмы лучевой трассировки, позволяя их авторам добавить физически корректный учёт явления плеохроизма.

**Ключевые слова:** Фотореалистический рендеринг, физически корректный рендеринг, кристаллы, двойное лучепреломление, анизотропное поглощение, плеохроизм, поляризация, дисперсия света.

## 1. ВВЕДЕНИЕ

Оптически анизотропные среды, в том числе кристаллы, являются достаточно сложными объектами для физически корректного рендеринга. Из-за анизотропии среды взаимодействие луча света с кристаллом зависит от его направления распространения в среде и его состояния поляризации. Более того, анизотропные среды изменяют состояние поляризации луча, как при преломлении (они являются естественными поляризаторами) и отражении от их границы, так и в процессе распространения внутри среды (оптическая активность). Законы преломления, отражения и распространения света в таких средах оказываются более сложными, чем в изотропных. Наблюдается ряд несвойственных<sup>1</sup> изотропным средам оптических эффектов: двойное лучепреломление, коническая рефракция, плеохроизм, подробнее, см., например, [9].

Анизотропные среды можно разделить на одноосные и двуосные по числу направлений распространения луча, по которым отсутствует двойное лучепреломление. Двуосные кристаллы – это кристаллы наименее симметричных кристаллических систем (ромбической, моноклинной и триклинной) [8], т.е. обладающие меньшей симметрией, чем одноосные, а, следовательно, взаимодействие с ними описывается более сложными законами.

В трудах конференций и журналах за последние несколько лет небольшое число работ [1–3, 5–7, 9] посвящено проблеме рендеринга анизотропных кристаллов. Авторы части из них [7, 3, 5, 9] рассматривают только задачу рендеринга одноос-

<sup>1</sup> Изотропные среды могут проявлять те же свойства, в присутствии внешних электрических полей или внешнем механическом воздействии, см. искусственная анизотропия [8].

ных прозрачных кристаллов, решая её лишь для некоторых частных случаев. Только в работе [1] задача решена в общем виде: представлена локальная модель расчёта взаимодействия луча с границей между любыми средами (изотропной, одноосной и двуосной). В работах [1] и [7] представлено сравнение фотографии с рассчитанным изображением.

Проблема рендеринга изотропных поглощающих сред решена относительно давно, см., например, [4], и многие современные пакеты трёхмерного моделирования имеют возможность рендеринга таких объектов. Анизотропное поглощение рассматривается лишь в работах [2] и [6]. В работе [2] рассматривается вопрос рендеринга одноосных поглощающих кристаллов, рассчитывается изображение кристалла турмалина (без сравнения с фотографией), однако статья больше посвящена ускорению процессу рендеринга, а не его физической корректности. В работе [6] теоретически рассматривается расчёт абсорбции в двуосных кристаллах, но алгоритма рендеринга и каких-либо расчётов не представлено.

Данная работа рассматривает общий случай поглощения света изотропными, одноосными и двуосными кристаллами, предлагается физически корректная вычислительная модель расчёта поглощения света в таких средах. На базе разработанной модели и модели взаимодействия света с прозрачными кристаллами из работы [1] построен алгоритм физически корректного рендеринга анизотропных поглощающих кристаллов и кристаллических агрегатов. Проведены численные эксперименты.

## 2. ПОГЛОЩЕНИЕ СВЕТА

Поглощением электромагнитного излучения (света) называется процесс потери энергии потоком электромагнитного излучения, в процессе распространения его через вещество.

Уменьшение интенсивности потока (луча) происходит экспоненциально от расстояния пройденного лучом и количественно описывается законом Бугера-Ламберта-Бера [8]:

$$I = I_0 \exp(-2\omega n \kappa l / c), \quad (1)$$

здесь  $I$  – интенсивность луча света с частотой  $\omega$  и начальной интенсивностью  $I_0$  после прохождения им расстояния  $l$  в среде с безразмерным коэффициентом поглощения  $\kappa$  и коэффициентом преломления  $n$ , а  $c$  – скорость света.

На практике (при расчётах и измерениях) используют это же выражение, но записанное в иной форме:

$$I = I_0 \exp(-\kappa_\lambda l), \quad (2)$$

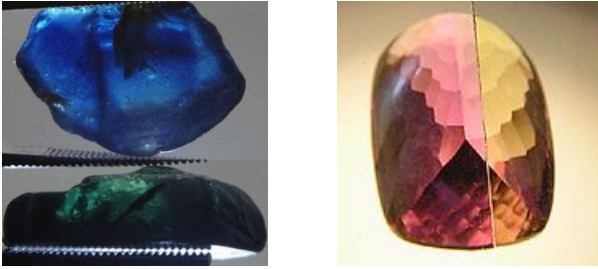
здесь  $\kappa_\lambda$  – это коэффициент поглощения, т.е. величина обратная расстоянию, при прохождении которого луч света ослабляется в  $e$  раз. Коэффициент поглощения  $\kappa_\lambda$  связан с безразмерным коэффициентом поглощения  $\kappa$  как:

$$\kappa_\lambda = 4\pi\kappa/\lambda = 4\pi n\kappa/\lambda_0, \quad (3)$$

где  $\lambda$  – длина волны в среде, а  $\lambda_0$  – длина волны в вакууме. Более частое использование формы (2) обуславливается тем, что при измерениях получается именно  $\kappa_\lambda$ , а получение коэффициента преломления требует дополнительных измерений, кроме того форма (2) более удобна в расчётах.

Зависимость коэффициента поглощения  $\kappa_\lambda$  от длины волны называется спектром поглощения и совместно со спектром источника света определяет цвет кристалла на просвет.

В анизотропных средах зависимость коэффициента поглощения от направления распространения луча приводит к тому, что одноосные кристаллы имеют два главных спектра поглощения, и, соответственно, две главные окраски – дихроизм, а двуосные – три спектра и три окраски – плеохроизм.



**Рис. 1.** Дихроизм сапфира при повороте кристалла (слева) и плеохроизм цоизита при просмотре через поляризатор.

Окраски могут проявляться при наблюдении кристалла с разных ракурсов или при повороте кристалла (рис. 1, слева), а также при наблюдении кристалла в свете различной поляризации или через поляризационный фильтр (рис. 1, справа).

### 3. МОДЕЛЬ ПОГЛОЩЕНИЯ СВЕТА

Для учёта поглощения света в алгоритме физически-корректного рендеринга кристаллов необходимо и достаточно уметь рассчитывать изменение (уменьшение) интенсивности луча, проходящего через кристаллическую среду от одной грани до другой. Из выражения (2) следует, что для этого необходимо вычислить коэффициент поглощения кристалла по направлению и состоянию поляризации луча.

Опуская подробности, из монографии [8] следует, что безразмерный коэффициент поглощения для анизотропной слабо поглощающей среды (т.е. верно для полупрозрачных кристаллов) ромбической и более симметричных кристаллических систем может быть вычислен из следующего соотношения (обозначения из [8]):

$$\kappa v_p^2 \left\{ \frac{s_x^2}{(v_p^2 - v_x^2)^2} + \frac{s_y^2}{(v_p^2 - v_y^2)^2} + \frac{s_z^2}{(v_p^2 - v_z^2)^2} \right\} = \frac{\kappa_x v_x^2 s_x^2}{(v_p^2 - v_x^2)^2} + \frac{\kappa_y v_y^2 s_y^2}{(v_p^2 - v_y^2)^2} + \frac{\kappa_z v_z^2 s_z^2}{(v_p^2 - v_z^2)^2}, \quad (4)$$

где  $v_p$  – фазовая скорость луча,  $v_k, k \in \{x, y, z\}$  – главные фазовые скорости, т.е.  $v_k = c / n_k$ , где  $n_k$  – главные коэффициенты преломления,  $\kappa_k$  – главные безразмерные коэффициенты поглощения,  $s = (s_x, s_y, s_z)$  – единичный вектор волновой нормали [1, 8].

Поскольку каждому направлению фазовой нормали соответствует две различные фазовые скорости, две волны распространяются по одному направлению, но взаимно перпендикулярно поляризованы и будут иметь различные коэффициенты

поглощения. Это объясняет изменение окраски при наблюдении кристалла через поляризатор (рис. 1, справа). При повороте кристалла изменяется множество наблюдаемых лучей, то есть изменяется направление  $s$  (рис. 1, слева).

Формула (4) верна для двуосных, одноосных и изотропных сред, но может быть несколько упрощена для последних двух [8]. Для необыкновенного луча в одноосной среде:

$$\kappa_{pe} v_{pe}^2 = \kappa_o v_o^2 \cos^2 \theta + \kappa_e v_e^2 \sin^2 \theta, \quad (5)$$

где  $v_{pe}$  – фазовая скорость необыкновенного луча,  $\kappa_o, \kappa_e$  – главные безразмерные коэффициенты поглощения обыкновенного и необыкновенного лучей,  $v_o, v_e$  – главные фазовые скорости,  $\theta$  – угол между вектором  $s$  и оптической осью.

Для обыкновенного луча в одноосной среде, как и для луча, распространяющегося в изотропной среде, коэффициент поглощения не зависит от направления распространения:

$$\kappa_{po} = \kappa_o \text{ и } \kappa_{pi} = \kappa_i. \quad (6)$$

В выражениях (4), (5) и (6) используются главные безразмерные коэффициенты поглощения и значения главных скоростей. Как было отмечено ранее, для расчётов необходимо преобразовать их с использованием публикуемых данных (справочники, интернет), то есть главных коэффициентов поглощения  $\kappa_{\lambda x}, \kappa_{\lambda y}, \kappa_{\lambda z}$  и главных коэффициентов преломления  $n_x, n_y, n_z$ . Используя соотношение (3) получаем:

$$\kappa_\lambda = \frac{n^3 (\kappa_{\lambda x} s_x^2 n_x Y_n Z_n + \kappa_{\lambda y} s_y^2 n_y X_n Z_n + \kappa_{\lambda z} s_z^2 n_z X_n Y_n)}{n_x^4 s_x^2 Y_n Z_n + n_y^4 s_y^2 X_n Z_n + n_z^4 s_z^2 X_n Y_n}, \quad (7)$$

где  $X_n = (n_x^2 - n^2)^2, Y_n = (n_y^2 - n^2)^2, Z_n = (n_z^2 - n^2)^2$ , а  $n$  – фазовый коэффициент преломления луча.

Для необыкновенного луча в одноосной среде:

$$\kappa_{\lambda e} = n^3 \left( \frac{\kappa_{\lambda o} \cos^2 \theta}{n_o^3} + \frac{\kappa_{\lambda e} \sin^2 \theta}{n_e^3} \right), \quad (8)$$

здесь  $n_o$  и  $n_e$  – главные коэффициенты преломления для обыкновенного и необыкновенного луча.

Выражение (6) для обыкновенного луча и изотропной среды практически не изменяется:

$$\kappa_{\lambda o} = \kappa_{\lambda o} \text{ и } \kappa_{\lambda i} = \kappa_{\lambda i}. \quad (9)$$

Выражения (7), (8), (9) позволяют рассчитать коэффициент преломления для луча с заданной длиной волны по спектрам преломления и поглощения среды. Учитывая расстояние, пройденное лучом и, подставляя вычисленное значение коэффициента поглощения в выражение (2), получаем изменение интенсивности луча.

### 4. АЛГОРИТМ РЕНДЕРИНГА

В данной главе рассматривается аспект интеграции построенной модели в алгоритмы лучевой трассировки. Интеграция показана на примере модели взаимодействия света с прозрачными кристаллами из работы [1].

Классический алгоритм обратной лучевой трассировки состоит из выпуска трассировочных лучей из камеры и дальнейшей трассировки их сквозь сцену, эти виртуальные лучи, отражаясь и преломляясь на границах объектов сцены как обычные лучи, формируют дерево трассировки. Лучи



пропускаются через каждый пиксель матрицы изображения и таким образом строят одно или несколько деревьев для каждого пикселя.

Построение ветви дерева обрывается, если луч попадает в источник (в нашем случае считается, что источник света не отражает и не преломляет лучи), при выполнении некоторой оценки или при достижении определённой глубины трассировки (наш случай). Расчёт (сбор) интенсивности осуществляется на обратном проходе дерева.

На прямом ходе достаточно вычислить направления лучей, их вид (отражённый или преломлённый) и тип для лучей внутри анизотропной среды (обыкновенный или необыкновенный – для одноосной среды, быстрый или медленный – для двуосной), а точнее их плоскость поляризации, т.к. именно она и определяет их тип. Расчёт состояния поляризации и интенсивности (включая расчёт поглощения) лучей может быть осуществлён только на обратном ходе, поскольку состояние поляризации виртуальных лучей не определено.

Рассмотрим этап расчёта лучей на обратном ходе:

1. Имеется входной луч, идущий от источника (лист дерева) или вычисленный на предыдущем этапе, с определённым состоянием поляризации и заданной интенсивностью (заданы через матрицу когерентности [1]). Луч пришёл из среды 1 и падает на границу со средой 2.

2. Рассчитываются направления всех отражённых и преломлённых лучей (до 4-х). Выбирается луч, который распространяется в среде 2 (или 1, для отражённого луча) и имеет заданную плоскость поляризации (тип луча), см. [1].

3. Суммируются матрицы когерентности (состояние поляризации, интенсивность) всех лучей (до 4-х) пришедших по дереву трассировки в данную точку (узел). Поскольку для каждого из них выполняются шаги 1 и 2, они будут иметь одну плоскость поляризации и одну систему координат матрицы когерентности [1].

4. Вычисляется расстояние до точки пересечения этого луча со следующей границей сред в сцене.

5. Рассчитывается поглощение луча. В зависимости от типа луча по одной из формул: (7) – для луча в двуосной среде, (8) – для необыкновенного луча (поляризован в плоскости фазовой нормали луча и оптической оси) в одноосной среде, (9) – для обыкновенного луча и луча, распространяющегося в изотропной среде. Отметим, что перечисленные формулы имеют те же обозначения, что и в работе [1], необходимо лишь заменить вектор  $s$  на  $n$ .

6. Если на этом этапе мы достигли камеры, то интенсивность помещается (добавляется) в значение пикселя изображения, обратная трассировка дерева завершена. В противном случае, получаем новый входной луч и переходим на шаг 1.

В результате работы алгоритма получаем рассчитанное спектральное изображение сцены. Изображение может быть переведено в RGB посредством преобразования CIE<sup>2</sup>.

Аналогичным образом предложенная модель поглощения света может быть встроена и в алгоритмы прямой лучевой трассировки.

## 5. ЭКСПЕРИМЕНТЫ

В экспериментах использовалась одна из опубликованных сцен из базы тестов, авторов работы [1], см. [9].

<sup>2</sup> [http://en.wikipedia.org/wiki/CIE\\_1931\\_color\\_space](http://en.wikipedia.org/wiki/CIE_1931_color_space)

Сцена состоит из одного кристалла, окружённого увеличенным габаритным боксом, грани которого являются текстурами-источниками (рис. 2). Одна из текстур белая (CIE D65) с чёрным крестом в центре, остальные текстуры серые и имеют спектр излучения 40% от CIE D65. Все текстуры излучают неполяризованный свет.

Геометрия кристалла представляет собой куб, с гранями параллельными граням бокса, расположенный над текстурой, почти, но не касаясь её. Одна из осей симметрии куба проходит через центр креста на текстуре.

С оптическими свойствами дело обстоит сложнее. Преломляющие свойства некоторых прозрачных анизотропных минералов (в особенности, применяемых в оптических приборах) хорошо известны, например, кальцита<sup>3</sup> или синтетического сапфира (почти прозрачный)<sup>4,5</sup>. Спектры поглощения (включая видимый диапазон частот) также опубликованы<sup>6</sup> для множества минералов, в том числе и сапфира (синего)<sup>7</sup>.

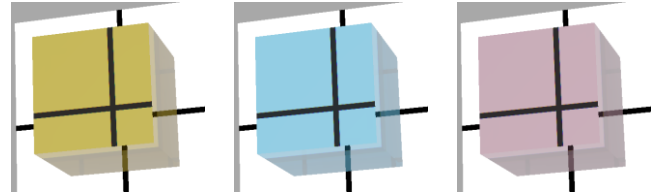


Рис. 2. Изотропные кубы: жёлтый, синий, красный.

Проблема состоит в том, что спектр преломления зависит от спектра поглощения, см. (3), (7) и их необходимо измерять для одного и того же кристалла, но результатов таких измерений найдено не было. Таким образом, существует проблема поиска данных для корректного расчёта реального кристалла и сравнения рассчитанного изображения с фотографией.

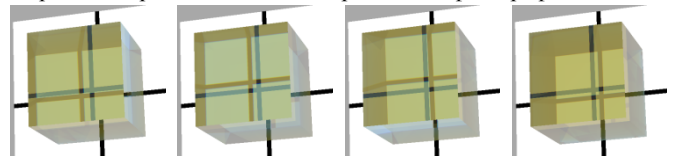


Рис. 3. Слева направо: одноосный куб, он же повернутый на 90 градусов вокруг оси X, вокруг оси Y, вокруг оси Z.

В данной работе для проверки работы алгоритма мы использовали виртуальные оптические свойства. Свойства выбраны так, чтобы кристалл обладал сильным двойным лучепреломлением и сильным плеохроизмом.

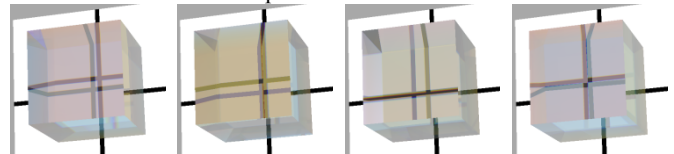


Рис. 4. Слева направо: двуосный куб, он же повернутый на 90 градусов вокруг оси X, вокруг оси Y, вокруг оси Z.

Преломляющие свойства кристаллов – виртуальные: изменяются линейно на видимом диапазоне волн от 380 до 780 нм: 1) от 1.55 до 1.45; 2) от 1.75 до 1.65; 3) от 1.95 до 1.85. Изотропная среда имеет 1-й спектр преломления, одноосная – 1-й и 2-й, а двуосная – все три.

Спектры поглощения взяты от реальных минералов для достижения естественной окраски кристалла, но умножены на различные константы для достижения примерно одинаковой

<sup>3</sup> <http://www.mkt-intl.com/pdfs/calcite.PDF>

<sup>4</sup> <http://www.tydexoptics.com/pdf/Sapphire.pdf>

<sup>5</sup> [http://en.wikipedia.org/wiki/Sellmeier\\_equation](http://en.wikipedia.org/wiki/Sellmeier_equation)

<sup>6</sup> <http://minerals.gps.caltech.edu/FILES/Visible/Index.html>

<sup>7</sup> <http://minerals.gps.caltech.edu/FILES/Visible/corundum/Index.htm>

степени полупрозрачности для заданного размера кристалла. Используются три спектра поглощения жёлтый, синий и красный. Для наглядности на рис. 2 приведены рассчитанные изображения трёх изотропных кубов (того же размера, что и на рис. 3 и 4) соответствующего цвета. Спектры используются по той же схеме, что и спектры преломления выше, т.е. изотропная среда – жёлтая, в одноосной среде используется жёлтый и синий, а в двуосной – жёлтый, синий и красный.

На рис. 3 приведены изображения сцен, содержащих одноосные кубы. Вторая, третья и четвёртая сцены отличаются от первой поворотом куба в пространстве вокруг осей, проходящих через центр куба и параллельным рёбрам куба. То есть сцены отличаются так, как если бы кубик переворачивали с грани на грань и смотрели, что получается. На рис. 4 показан тот же эксперимент, но с двуосным кристаллом.

На изображениях хорошо видно двойное лучепреломление (двойное изображение креста), плеохроизм (изменение окраски при поворотах куба). Дисперсию света можно увидеть на резких переходах цветов.

## 6. ЗАКЛЮЧЕНИЕ

В работе детально рассматривается поглощение света анизотропными средами: предложена физически корректная модель расчёта поглощения луча, в процессе его распространения сквозь такую среду, рассмотрен процесс интеграции данной модели в стандартный алгоритм лучевой трассировки. Впервые в мире рассчитаны изображения тестовых сцен, содержащих поглощающий двуосный кристалл.

Дальнейшее развитие данной работы: 1) расширение предложенной модели поглощения на оставшиеся кристаллические системы, то есть триклинную и моноклинную, отметим, однако, что представленная модель уже может быть использована для рендеринга таких кристаллов в качестве некоторого приближения; 2) сравнение рассчитанного изображения с фотографией.

Отметим, что в данном случае задача сравнения с фотографией является более сложной, чем в случае с прозрачными анизотропными кристаллами. Метод, описанный в работе [1] оказывается пригодным, однако дополнительно требуется измерить спектры поглощения и преломления образцов кристаллов (нельзя использовать табличные данные, поскольку окраска очень сильно зависит от образца) и ограничить их предварительно рассчитанным образом, чтобы максимально усилить видимость плеохроизма при фотографировании.

## 7. БЛАГОДАРНОСТИ

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 12-07-00391 а.

## 8. БИБЛИОГРАФИЯ

- [1] Debelov V.A., Kozlov D.S. *A Local Model of Light Interaction with Transparent Crystalline Media*. // *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 8, pp. 1274-1287, Aug. 2013, doi:10.1109/TVCG.2012.304.
- [2] Guy S., Soler C. *Graphics gems revisited: fast and physically-based rendering of gemstones*. // *Transactions on Graphics (SIGGRAPH '04 Proceedings)*, vol. 23, issue 3, Aug. 2004, pp. 231-238.
- [3] McClain S. C., Chipman R.A. *Polarization ray tracing in anisotropic optically active media II, theory and physics* // *Applied Optics*, vol. 10, no. 11, pp. 2383-2393 (Nov. 1993).

- [4] Sun Y., Fracchia F.D., Drew M.S. *Rendering diamonds*. // *In Proceedings of the 11th Western Computer Graphics Symposium (WCGS)*, 9-15, 2000.
- [5] Tannenbaum D.C., Tannenbaum P., Wozny M.J. *Polarization and birefringency considerations in rendering*. // *Comp. Graphics (SIGGRAPH '94 Proceedings)*, ACM Press, 221-222.
- [6] Wang Y. et al. *Complex ray tracing in biaxial anisotropic absorbing media*. // *Journal of Optics A: Pure and Applied Optics* vol. 10(7), p. 075009 (7pp).
- [7] Weidlich A., Wilkie A. *Realistic Rendering of Birefringency in Uniaxial Crystals* // *ACM Transactions on Graphics*, Vol. 27, No. 1, – 2008, pp. 6:1-6:12.
- [8] Борн М., Вольф Э. *Основы Оптики*. Москва, “Наука”, 1973.
- [9] Дебелов В.А., Козлов Д.С. *Верификация алгоритмов фотореалистического рендеринга кристаллов* // *Труды XX Международной конференции по компьютерной графике и зрению ГрафиКон-2010, 20-24 сентября 2010, СПбГУ, Санкт-Петербург, Россия, – С. 238 – 245.*

## Об авторах

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.sccc.ru.

Дмитрий Сергеевич Козлов – аспирант Новосибирского государственного университета и инженер Института вычислительной математики и математической геофизики СО РАН. E-mail: kozlov@oapmg.sccc.ru.

## Physically correct algorithm of rendering anisotropic absorbing crystals

This paper is devoted to the problem of photorealistic rendering of optically isotropic and anisotropic absorbing crystals (media), notably computing light absorption during ray propagation through a crystal media.

The paper describes a local model for calculation of ray attenuation during its propagation through isotropic, uniaxial, and biaxial media. A physically correct algorithm of rendering anisotropic absorbing crystals was built on the developed.

The model is intended to be used in ray tracing algorithms allowing their authors computing pleochroism phenomenon in a physically correct way.

**Keywords:** *Photorealistic rendering, physically based rendering, crystals, birefringence, polarization, anisotropic absorption, pleochroism, polarization, optical dispersion.*

## About the authors

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 the Novosibirsk State University. E-mail: debelov@oapmg.sccc.ru.

Dmitry S. Kozlov is a Ph.D. student of the Novosibirsk State University and an engineer of the Institute of Computational Mathematics and Mathematical Geophysics SB RAS. His contact email is kozlov@oapmg.sccc.ru.

# Количественная оценка уровня артефактов в объёмной визуализации

Николай И. Гаврилов, Вадим Е. Турлапов

Нижегородский государственный университет, Н.Новгород, Россия

[gavrilov86@gmail.com](mailto:gavrilov86@gmail.com); [vadim.turlapov@cs.vmk.unn.ru](mailto:vadim.turlapov@cs.vmk.unn.ru)

## Аннотация

Несмотря на обилие подходов к оптимизации алгоритма объёмного рендеринга в терминах качества и производительности, в литературе практически отсутствует формальный подход к количественной оценке качества изображений, получаемых в ходе визуализации научных и медицинских пространственных данных. В статье предложен метод оценки артефактов алгоритма объёмного рендеринга, связанных с недостаточно малой длиной шага луча в методе испускания лучей. На основе предложенного метода оценки в ходе экспериментов сравниваются различные подходы, призванные улучшить качество рендеринга, для разных случаев визуализации выявляются оптимальные из этих подходов по критерию качество/производительность. Предложены также новые методы устранения артефактов выборки, отличающиеся использованием таблиц предынтегрированного рендеринга в методе виртуальных выборок вместо обычной модели постклассификации.

**Keywords:** медицинская визуализация, научная визуализация, оценка качества, объёмный рендеринг, GPU, GPGPU.

## 1. ВВЕДЕНИЕ

Техника прямого объёмного рендеринга (Direct Volume Rendering, DVR) и, в том числе, техника испускания лучей (Ray Casting, RC) широко применяется для визуального анализа данных в научной и медицинской визуализации. Сегодня существует множество подходов, позволяющих производить объёмную визуализацию в реальном времени с использованием высокопроизводительных вычислений на GPU [ЕНК\*06]. Помимо использования иерархических ускоряющих структур, обеспечивающих пропуск пустых областей для повышения производительности, существуют подходы, повышающие качество визуализации без значительного ущерба её производительности [EHMDM08], [KHW\*09], [LYS\*10].

Тем не менее, в литературе отсутствуют методы количественной оценки качества рендеринга, которые позволили бы сравнивать эффективность различных методов. Обычно исследователи ограничиваются демонстрацией полученных изображений, предоставляя человеку самому оценивать качество [MNB\*00].

## 2. АРТЕФАКТЫ ОБЪЁМНОГО РЕНДЕРИНГА

Из-за более высокого качества и более высокой производительности в алгоритме DVR используют модель постклассификации, когда шаг классификации (раскраски) точек пространства происходит после шага интерполяции значения исходных данных. Однако использование данной модели приводит к значительному увеличению уровня артефактов выборки (*sampling artifacts* [KHW\*09]). Регулярная структура этих артефактов на изображении

ликвидируется техникой *jittering* – случайным сдвигом стартовых позиций лучей [Sch05]. Повышение частоты выборки вдоль луча подавляет данные артефакты, снижая при этом производительность из-за увеличения общего числа выборок из данных. Для поддержания качественной и интерактивной визуализации массивов данных больших размеров (более  $512^3$  вокселей) или при использовании сложных передаточных функций, необходимо использование техник, повышающих точность интегрирования вдоль луча [EHMDM08], [KHW\*09], [LYS\*10].

Другой тип артефактов алгоритма DVR возникает на шаге интерполяции значения данных – *артефакты фильтрации* (*filtering artifacts*), которые заметны при использовании трикубической интерполяции. К сожалению, эти артефакты не могут быть зашумлены техникой *jittering*, как артефакты выборки, но они в значительной степени устраняются при трикубической интерполяции B-сплайнами [RtHRS08].

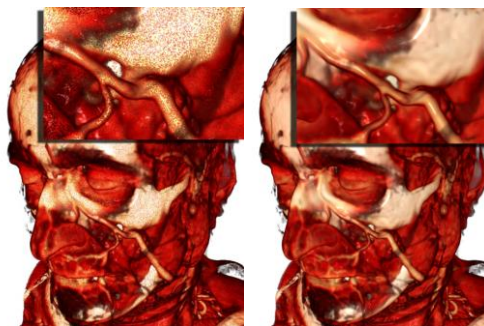


Рис. 1. Сравнение качества визуализации алгоритмом RC при частоте 1 (слева) и 8 (справа) выборок на воксель.



Рис. 2. Сравнение качества визуализации алгоритмом RC с трилинейной (слева) и трикубической (справа) интерполяцией.

## 3. КОЛИЧЕСТВЕННАЯ ОЦЕНКА КАЧЕСТВА RC

Для оценки артефактов выборки мы вычисляем ошибку в каждом пикселе. Для этого методом испускания лучей с применением техники случайного старта луча (*jittering*) производится рендеринг серии изображений с одного и того же ракурса. Таким образом, каждому пикселю соответствует серия RGB-значений. Степень разброса этих значений характеризует ошибку пикселя. Для вычисления ошибки пикселя использованы величины дисперсии ( $D$ ) цвета в

цветовых пространствах YCbCr и HSL. Для каждого пикселя с координатами (x,y) дисперсия вычисляется следующим образом:

$$D(x, y) = \frac{1}{T} \sum_{i=1}^T \left\| C_i(x, y) - \frac{1}{T} \sum_{j=1}^T C_j(x, y) \right\|^2$$

Здесь  $T$  – число изображений в серии,  $C_i(x,y)$  – цвет пикселя с координатами (x,y) из  $i$ -ого изображения. Стоит отметить, что вместо математического ожидания в формуле может быть использован цвет пикселя изображения  $\tilde{C}$  (ground truth image), полученного с максимально возможной частотой выборки:

$$D(x, y) = \frac{1}{T} \sum_{i=1}^T \left\| C_i(x, y) - \tilde{C}(x, y) \right\|^2$$

Для оценки ошибки изображения в целом использовано пиковое отношение сигнала к шуму (PSNR – Peak Signal-to-Noise Ratio), которое используется в теории сигналов и для измерения уровня искажений при сжатии изображений с потерями. В объёмном рендеринге этот подход был использован ранее, но для оценки ухудшения качества при сжатии самих объёмных данных [GS04], а не для оценки артефактов рендеринга:

$$PSNR = 10 \log_{10} \left( \frac{MAX_I^2}{MSE} \right); \quad MSE = \frac{1}{N} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} D(i, j).$$

$D(i,j)$  – дисперсия или квадратичная ошибка пикселя (i,j), качество которого мы оцениваем,  $MSE$  – среднеквадратическая ошибка изображения размером  $m \times n$ ,  $MAX_I$  – максимально возможное значение пикселя, в нашем случае равное 1,  $N$  – количество нефоновых пикселей, которое в нашем случае, так как есть смысл игнорировать фоновые пиксели, которые не содержат шума и ошибка в них будет нулевая. На рисунке 4 видно, что если при вычислении MSE учитывать фоновые пиксели (в данном случае белые), то

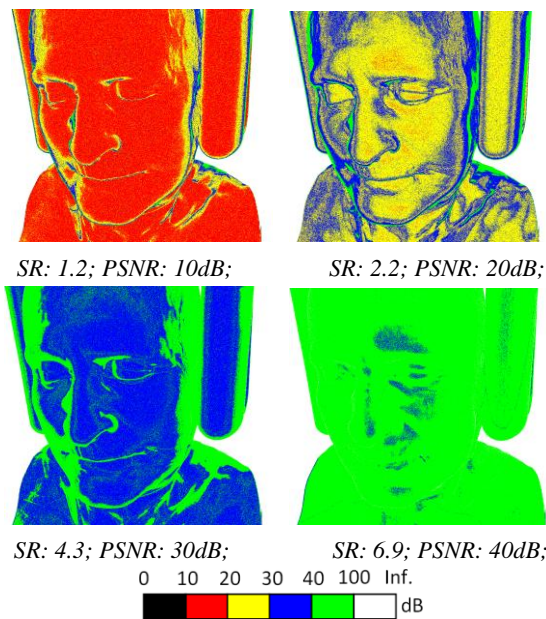
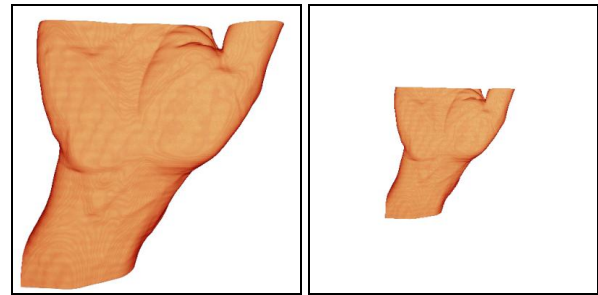


Рис. 3. Качество рендеринга при различных выборах частоты выборки SR

при удалении камеры от объекта происходит увеличение PSNR. В случае игнорирования фоновых пикселей PSNR будет неизменным.

Таким образом, оценка качества визуализации проводится с помощью оценки дисперсии пикселей. Кроме того, можно оценивать качество каждого отдельного пикселя для построения PSNR-карт (Рис. 3). Понятие PSNR в итоге адаптировано для анализа качества изображений объёмного рендеринга.



PSNR: 20.5dB      PSNR: 25.7dB  
 Рис. 4. Значения PSNR в случае учёта фоновых пикселей.

#### 4. ЭКСПЕРИМЕНТЫ

Кроме частоты выборки в методе RC существует множество факторов, влияющих на качество и производительность визуализации: сам массив данных, ракурс, передаточная функция (transfer function), размер окна вывода, размер блоков декомпозиции данных, модель освещения, используемая интерполяция значения данных, модель видеокарты и т.д. К счастью некоторые из этих параметров не влияют на качество рендеринга (на уровень различного рода артефактов). Размер окна вывода влияет только на точность оценки PSNR. Зависимость между числом обрабатываемых пикселей и временем рендеринга почти линейна, как и между частотой выборки и временем рендеринга.

В экспериментах мы использовали 10 различных массивов объёмных данных (Рис.5). Декомпозиция данных проводилась на блоки размером  $256^3$  каждый, что является оптимальным с точки зрения производительности разбиением для используемой нами видеокарты (GeForce GTX 580 3Gb).

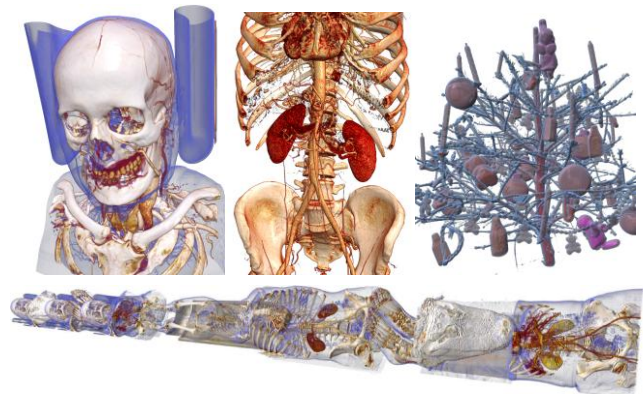


Рис.5. Примеры визуализации тестовых массивов данных размеров от 512x512x420 до 512x512x5382 (12-bit).

#### 4.1 Отношение качество-производительность

При сравнении эффективности различных методов улучшения качества и/или производительности метода RC необходимо учитывать как производительность, так и получаемое качество. В наших экспериментах мы меняем частоту выборки вдоль луча (с 1 до 8 выборков на воксель), тем самым варьируя качество и производительность. В результате каждому методу рендеринга соответствует график (Рис.7-8), показывающий зависимость качества (PSNR) рендеринга от производительности.

Были использованы такие методы рендеринга, как: UDVR (неоптимизированный DVR), PDVR (предынтегрированный рендеринг [EHMDM08]), LVS и CVS (метод виртуальных выборок соответственно с линейной и кубической (рис.6) интерполяцией вдоль луча [LYS\*10]), ASM (метод с адаптивным шагом луча), PLVS и PCVS (модификации методов LVS и CVS: вместо обычной постклассификации мы применяем предынтегрированную классификацию подобно методу PDVR).

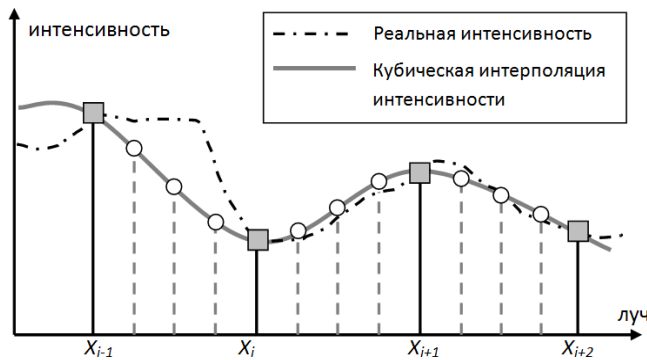


Рис.6. Виртуальные выборки (белые точки) при интегрировании вдоль луча в алгоритме Ray Casting.

Методы PLVS и PCVS предложены в данной работе, как альтернативы методу PDVR. На рисунке 6 сплошные вертикальные линии обозначают выборку значений из данных, пунктирные вертикальные линии показывают виртуальные выборки, увеличивающие точность интегрирования. Кроме интенсивности интерполируется также и градиент, что повышает точность интегрирования при использовании локального освещения в алгоритме.

При проведении экспериментов использовались различные опции, влияющие на качество и производительность: на подписях к диаграммам указано, была ли использована трикубическая фильтрация и/или локальное освещение. Диаграммы отображают усредненные результаты для разных исходных данных, ракурсов камеры, размеров окна вывода и передаточных функций. Каждая точка графика соответствует определённой длине шага луча, поэтому на усреднённой диаграмме каждая точка каждого графика – это пара чисел <средняя производительность, среднее качество>. Обобщив результаты, из диаграмм можно заключить следующее:

Предынтегрированный рендеринг (PDVR) оптимален в случаях отсутствия локального освещения, он во многом выигрывает благодаря высокой производительности, хотя в случаях использования локального освещения метод даёт недостаточно хорошее качество из-за грубости учёта локального освещения. Предынтегрированная таблица хранит результаты интегрирования без учёта затенения.

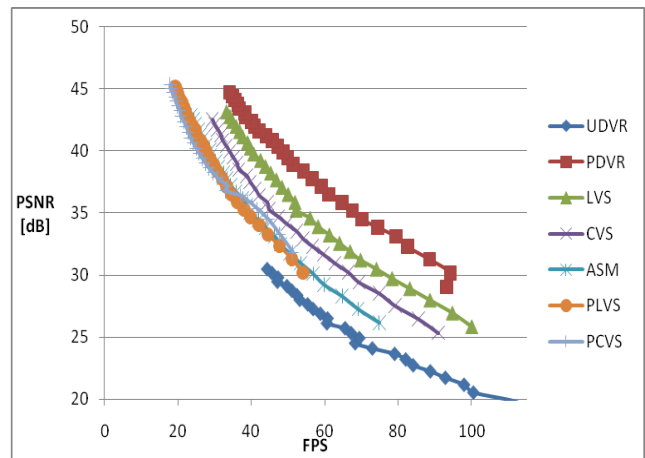


Рис.7. Диаграмма соотношений качество-производительность для случая без затенения и с трилинейной фильтрацией.

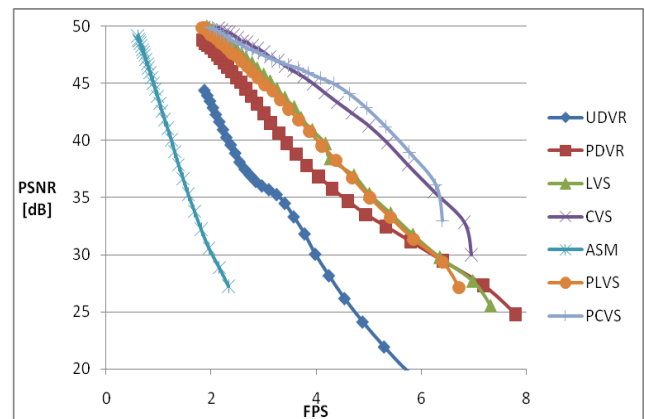


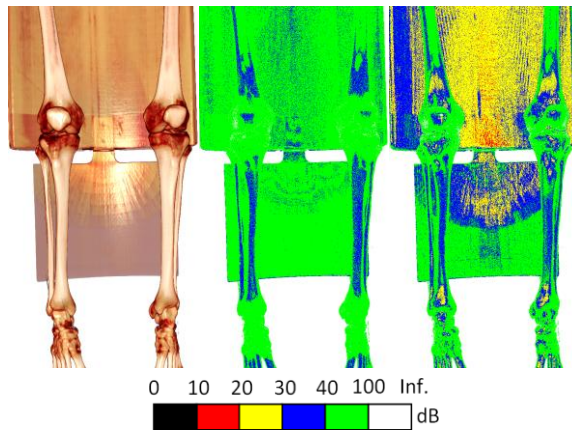
Рис.8. Диаграмма соотношений качество-производительность для случая с затенением и с трикубической фильтрацией.

Почти на всех полученных диаграммах метод UDVR неэффективен, что говорит о целесообразности оптимизации метода. Оценивая получаемое значение PSNR во внимание стоит принимать диапазон [30dB, 40dB], соответствующий приемлемому качеству рендеринга. В случае PSNR > 40dB артефакты выборки практически неразличимы.

Метод ASM хорошо показывает себя только в случае трилинейной фильтрации, т.к. выборки из данных производятся на каждом шаге, в отличие от методов LVS и CVS. Градиент также вычисляется на каждом шаге, поэтому метод даёт хорошее качество при использовании локального освещения. Однако из-за необходимости делать много выборок метод отстаёт по производительности. Таким образом, метод часто оптимален при использовании локального освещения при трилинейной фильтрации;

Методы виртуальных выборок (LVS, CVS) менее производительны, чем PDVR, поэтому во многих случаях ему уступают. Эти методы оптимальны в случае локального освещения при трикубической фильтрации, т.к. общее падение производительности из-за трикубической фильтрации делает накладные расходы на разбиение шага менее заметными.

Гибридные методы PLVS и PCVS хорошо работают в случаях трикубической интерполяции и локального освещения (рис.9). В этих случаях они превосходят метод PDVR. Благодаря разбиению шага методы дают лучшее качество при освещении. Благодаря предынтегрированной классификации, данные методы в большинстве случаев работают лучше методов разбиения шага без предынтеграции (LVS, CVS).



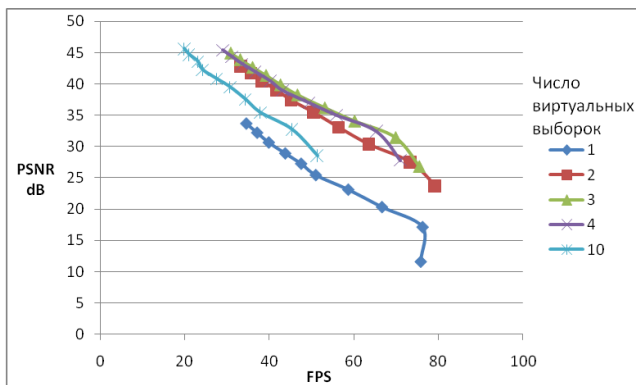
DVR Output; PSNR: 36.4dB; PSNR: 22.1dB;

**Рис.9.** Карты PSNR для метода предынтегрированного рендеринга без локального освещения (в середине) и с ним (справа); частота выборки: 2. Из-за грубого учёта затенения в предынтегрированном рендеринге возникают артефакты.

#### 4.2 Оптимизация параметров алгоритма RC

Перед сравнением методов были проведены оптимизации их параметров: во всех методах за исключением UDVR и PDVR есть возможность выбора числа разбиений шага. Очевидно, что чем больше это число, тем выше будет качество этих методов, однако при этом будет падать производительность. Для каждого из этих методов были построены диаграммы соотношения качество-производительность (рис.10), где каждая линия соответствует числу разбиений шага луча.

Для подавляющего большинства случаев оптимальным является разбиение 3, причём для каждого из исследуемых методов. Использование менее чем трёх промежуточных шагов даёт слишком низкое качество изображения, которое не компенсируется даже уменьшением длины шага луча, тогда как, начиная с трёх делений, качество почти не улучшается, а производительность продолжает падать.



**Рис.10.** Соотношения качество-производительность для метода LVS с различным числом виртуальных выборок.

## 5. ЗАКЛЮЧЕНИЕ

Предложен метод оценки артефактов DVR, вызванных недостаточно малой длиной шага луча в методе испускания лучей. Хотя в ходе экспериментов не было выявлено RC-алгоритма, оптимального с точки зрения качества и производительности в любых условиях визуализации, метод оценки показал целесообразность оптимизации алгоритма объёмного рендеринга в терминах качества и производительности.

Предложены новые методы устранения артефактов постклассификации в алгоритме RayCasting, отличающиеся использованием предынтегрированной классификации в методе виртуальных выборок вместо постклассификации.

Использование отношения PSNR позволило привести уровень шума к логарифмической шкале в дБ. Оказалось, что диапазон значений от 30 до 40 дБ, используемый как правило для сжатия изображений и соответствующий приемлемому качеству сжатия с потерями, также соответствует приемлемому качеству при оценке качества в методе Volume Raycasting.

Работа выполнена при поддержке грантов Президента РФ № НШ-1960.2012.9 и Правительства РФ № 11.G34.31.0066.

## 6. ССЫЛКИ

- [EHK\*06] Engel K., Hadwiger M., Kniss J., Rezkasama C., Weiskopf D.: Real-time volume graphics. Eurographics Association (2006), 112–114.
- [EHMDM08] El Hajjar J. F. et al. Second order pre-integrated volume rendering //Visualization Symposium, 2008. PacificVIS'08. IEEE Pacific. – IEEE, 2008. – С. 9-16.
- [GS04] Guthe S., Strasser W.: Advanced techniques for high-quality multi-resolution volume rendering. Computers & Graphics 28, 1 (2004), 51–58.
- [KHW\*09] Knoll A., Hijazi Y., Westerteiger R., Schott M., Hansen C., Hagen H.: Volume ray casting with peak finding and differential sampling. Visualization and Computer Graphics, IEEE Transactions on 15, 6 (2009), 1571–1578.
- [MHB\*00] Meissner M., Huang J., Bartz D., Mueller K., Crawford R.: A practical evaluation of popular volume rendering algorithms. In Proceedings of the 2000 IEEE symposium on Volume visualization (2000), ACM, pp. 81–90.
- [LYS\*10] Lee, B., Yun, J., Seo, J., Shim, B., Shin, Y. G., & Kim, B. (2010). Fast high-quality volume ray casting with virtual samplings. Visualization and Computer Graphics, IEEE Transactions on, 16(6), 1525-1532.
- [RtHRS08] Ruijters D., Ter Haar Romeny B., Suetens P.: Efficient gpu-based texture interpolation using uniform b-splines. Journal of Graphics, GPU, and Game Tools 13, 4 (2008), 61–69.
- [Sch05] SCHARSACH H.: Advanced gpu raycasting. Proceedings of CESC 5 (2005), 67–76.

# Тестирование систем реалистичной компьютерной графики

А.Г. Волобой, Е.Ю. Денисов  
Институт прикладной математики им. М.В.Келдыша РАН, Москва

## АННОТАЦИЯ

В работе описаны технологии тестирования программных комплексов компьютерной графики, разрабатываемых в ИПМ им. М.В. Келдыша РАН. Сложность разрабатываемых продуктов, необходимость выпуска большого числа версий (50-70 версий в год) потребовали разработки собственной системы тестирования, позволяющей эффективно выявлять ошибки и исправлять их до выпуска продукта. Тестирование продуктов компьютерной графики имеют свою специфику, на которой и сделан акцент в работе.

**Ключевые слова:** системы реалистичной визуализации, тестирование программных комплексов, регрессионное тестирование, сравнение изображений

## 1. ВВЕДЕНИЕ

Тестирование программного обеспечения, особенно программных продуктов, поставляемых дизайнерам, инженерам и проектировщикам, всегда являлось важной и сложной проблемой. Современные программные комплексы синтеза фотореалистичных изображений, основанные на моделировании физических законов распространения света, достигли такого уровня сложности, на котором классическое тестирование этих комплексов становится неэффективным, а часто и невозможным по ряду причин.

Наряду с классическими подходами к тестированию, заложенными еще Майерсом [1], тестирование таких комплексов имеет также свою специфику. Отметим два наиболее сложных аспекта этого тестирования:

- Проверка правильности моделирования физических явлений;
- Интерактивность программных комплексов.

Системы реалистичной визуализации стремятся к все большей достоверности генерируемых изображений. Для этого некоторые системы не только предлагают спектральное моделирование, но и поддерживают такие сложные явления как поляризация и флюоресценция света. Верификация соответствующих алгоритмов становится самостоятельной непростой задачей. Одна из попыток серьезного подхода к этому вопросу опубликована в [2]. В статье предложены как аналитические тесты, проверяющие правильность выполнения законов оптики, так и комплексные визуальные тесты, сравнивающие полученные итоговые изображения с фотографией.

Сложность тестирования интерактивных комплексов компьютерной графики состоит в их автоматизации. Авторами были протестированы десятки программных пакетов, позволяющих в том или ином виде записать и воспроизвести действия пользователя в интерактивной программе. Среди них: AutoIt [3], Test Automation FX [4], TestComplete [5] и другие. К сожалению, обнаруженные недостатки не позволили полноценно использовать для тестирования создаваемых программных комплексов ни один из этих пакетов. Среди этих недостатков: запись действий пользователя в абсолютных координатах, что не

позволяет воспроизводить записанную сессию на другом компьютере с другим разрешением экрана; неспособность распознать UI элементы пакета Qt, который используется нами при создании программного продукта; невозможность записи сессии в реальном времени (сценарий должен быть предварительно написан на специальном языке); отсутствие поддержки новейших версий Windows, а также ошибки в работе (иногда приводящие к сбою всей OS). Такая ситуация потребовала создания собственного средства записи и воспроизведения действий пользователя.

При разработке собственной системы тестирования программных продуктов компьютерной графики можно выделить следующие дополнительные особенности, присущие нашим продуктам:

- Много разнородных комплексов имеют в своей основе общие программные компоненты (например, ядро оптического моделирования);
- Частые выпуски новых версий;
- Ограниченные ресурсы и время.

В частности, предлагаемые решения применяются для тестирования набора из десяти программных комплексов; частота выпуска новых версий – 50-70 версий в год; количество программных компонент – более 500; коллектив разработчиков состоит из 15 человек.

В таких условиях абсолютно необходимым является комплексный подход к автоматизации тестирования. Предложенные и описанные далее методы позволяют добиться приемлемого качества тестирования, а значит и приемлемого качества выпускаемых программных комплексов.

Автоматизированному тестированию подвергаются как отдельные программные компоненты (низкоуровневое тестирование), так и весь программный продукт в целом (высокоуровневое тестирование). В основном используется регрессионное тестирование, проверяющее совпадение поведения новой реализации системы и её предыдущей реализации. За исключением намеренно внесенных изменений и модификаций алгоритмов результаты моделирования двух версий программного комплекса должны совпадать.

## 2. РЕГРЕССИОННОЕ ТЕСТИРОВАНИЕ

### 2.1 Покомпонентное тестирование

После каждого изменения программной компоненты кроме непосредственного содержательного тестирования вновь добавленной функциональности проводится также её регрессионное автоматическое тестирование. Результат вычисления сравнивается с ожидаемым, и выдается заключение "тест пройден - не пройден". В последнем случае разработчиком исследуется ожидаемый и реальный вывод и производится исправление выявленных ошибок.

Для обеспечения автоматизации тестирования потребовалось создать несколько систем для сравнения результатов. Поскольку результатами работы различных

программных компонент являются как текстовые данные, так и изображения, потребовалось создать подсистемы сравнения и анализа для каждого из этих типов данных.

На рис. 1 приведён пример автоматического регрессионного тестирования компоненты аппаратно-независимой реализации математических библиотек; в процессе тестирования проверяются как точность реализации математических функций, так и скорость работы данной компоненты.

```

--- Run math\test\bin\math_test.exe
***Testing SSE version of Cos on [0,PI/2]***
The precision is 1.35103e-007
The acceleration is 14.401666
***Testing SSE version of Cos on [0, PI]***
The precision is 5.36442e-007
The acceleration is 6.695166
***Testing SSE version of Sin on [0, PI]***
The precision is 1.02073e-006
The acceleration is 7.321569
***Testing SSE version of Arcsin***
The precision is 6.76808e-005
The acceleration is 17.923254
***Testing SSE version of Arccos***
The precision is 6.77109e-005
The acceleration is 16.912662
***Testing the Binary search...***
The acceleration is 2.564422
SUCCESS!
--- Run math\perf\test\bin\math_perf.exe
Number of iterations 262144000
    
```

Рис. 1. Пример автоматического тестирования математической компоненты.

Подсистема сравнения изображений обеспечивает качественный и количественный анализ различий в сравниваемых изображениях. Ниже приведен пример работы этой подсистемы (рис. 2): сравниваемое изображение показано слева, разница изображений, полученных разными версиями, в графическом представлении для последующей оценки разработчиком приведена на рисунке справа. Кроме этого для каждого сравнения сохраняются количественные данные: абсолютные и относительные ошибки в различных цветовых пространствах, гистограмма ошибочных пикселей и т.д. Эти данные могут быть проанализированы разработчиком при необходимости.



Рис. 2. Пример работы системы сравнения изображений.

Важной особенностью является то, что сравнение изображений происходит в автоматическом пакетном режиме. Допустимый порог погрешности (разница в значениях несовпадающих пикселей и их допустимое количество) задается предварительно, и при превышении этого порога тестируемая подсистема считается ошибочной и подлежащей дальнейшему анализу и исправлению ошибок. Весь процесс тестирования автоматически протоколируется в журнал, содержащий список проверенных модулей и результаты их тестирования.

## 2.2 Тестирование готового программного продукта

Помимо покомпонентного низкоуровневого тестирования готовый программный продукт проходит также высокоуровневое тестирование, обеспечивающее проверку качества совместной работы компонент и их правильного использования разработчиком. Такое тестирование проводится также в пакетном режиме с использованием специальных модулей пакетного выполнения, являющихся частью программного комплекса, либо с использованием скриптов (в тех комплексах, которые поддерживают их выполнение).

Для тестирования интерактивных компонент, т.е. элементов интерфейса, требующих взаимодействия с пользователем, система тестирования предоставляет разработчику возможность проверить функционирование компоненты в интерактивном режиме. На рис. 3 показана подсистема отображения графиков, которая проходит автономное тестирование.

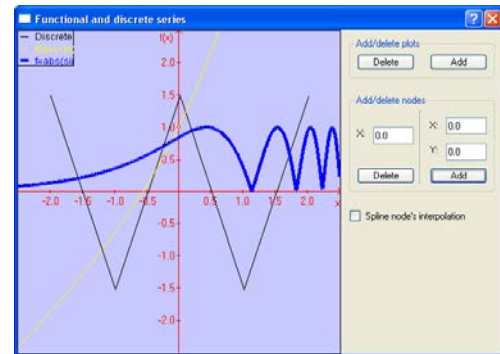


Рис. 3. Автономное тестирование компоненты.

Реально протестировать все возможные варианты использования комплекса компьютерной графики невозможно. Из-за огромного числа всевозможных опций и методов построения изображения время такого тестирования стремится к бесконечности. При этом тестирование версии должно быть произведено в разумное время. Отсюда возникает задача создания таких тестовых данных, чтобы за одно моделирование и генерацию изображения было проверено как можно большее число различных методов или их опций. На рис. 4 представлен пример одновременного тестирования нескольких функциональностей программного продукта. В данной специально созданной тестовой сцене проверяется правильность функционирования различных алгоритмов наложения текстур на различные типы и формы поверхностей. Для анализа правильности результата вновь используется подсистема сравнения изображений, описанная выше.

Предложенный подход помимо полной автоматизации тестирования позволяет производить тестирование на нескольких компьютерах параллельно, тем самым позволяя существенно сократить время, затрачиваемое на него. Используется как так называемое "натуральное распараллеливание", когда разные компоненты тестируются на разных компьютерах, так и параллельное вычисление в самих программных комплексах, которое дополнительно тестирует и функциональность параллельных вычислений.



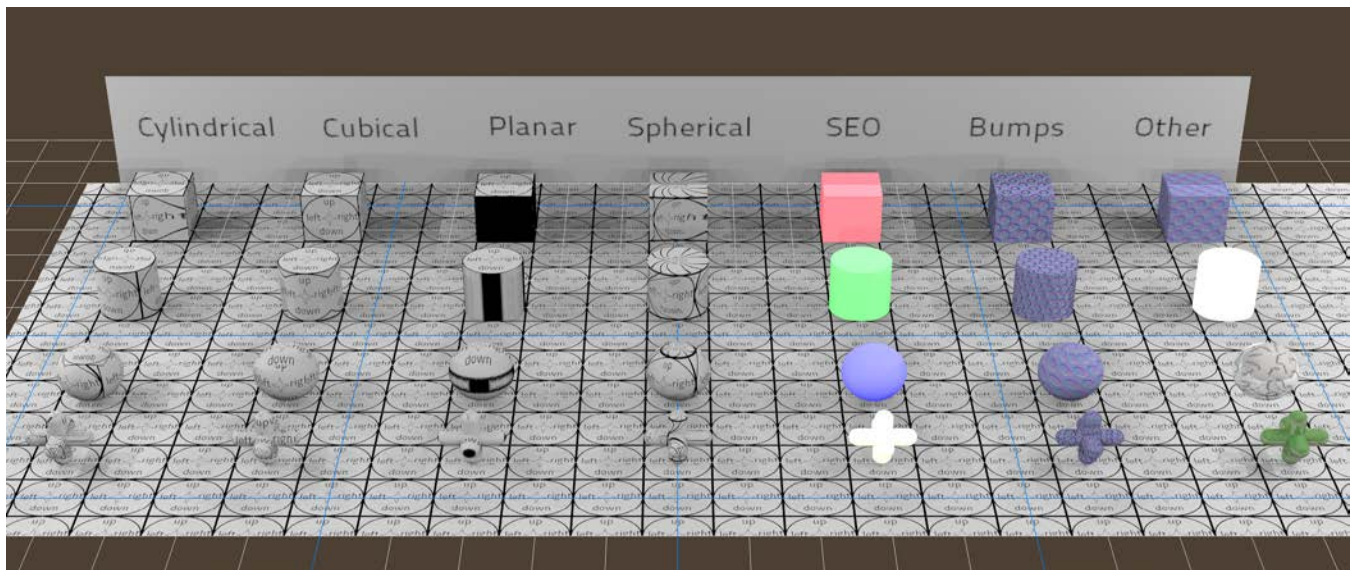


Рис. 4. Одновременная проверка различных алгоритмов

### 3. АВТОМАТИЧЕСКОЕ ТЕСТИРОВАНИЕ ИНТЕРАКТИВНЫХ ДЕЙСТВИЙ

Разработанная нами система записи и воспроизведения интерактивных действий пользователя названа TIPE – Testing Interactive Program Execution. Её функциональность можно сравнить с системами записи и воспроизведения макро команд, доступными в некоторых редакторах, однако возможности, реализуемые TIPE, гораздо шире.

#### 3.1 Базовые принципы TIPE

Все действия, производимые пользователем в программном комплексе, перехватываются и записываются модулем TIPE в файл. Позднее этот файл может быть воспроизведен, в том числе на другом компьютере, отличном по характеристикам от компьютера, использованного при записи. Выполнение записанных действий в программном комплексе произойдет в том же порядке, в каком оно было записано.

Система TIPE производит запись событий, которые изменяют состояние приложения, и реакцию приложения на эти события. Приложение рассматривается как конечный автомат. Записываемые и воспроизводимые события являются входными данными для этого автомата. Записываются не только события, непосредственно выполняемые пользователем (движения мыши, нажатия клавиш на мыши и на клавиатуре), но и другие события, приводящие к смене состояний приложения: сигналы программных таймеров, системные сообщения OS и прочие. Сам модуль TIPE и его использование являются “прозрачными” для программиста – для использования функциональности TIPE в процессе программирования не нужно прилагать никаких специальных усилий. Модуль TIPE автоматически перехватывает значимые события и реакцию приложения и производит их запись.

Для успешного функционирования системы TIPE необходимо выполнение следующих условий:

- Приложение при записи и воспроизведении должно быть тем же самым;
- Начальное состояние приложения должно совпадать;

- Изменения состояний приложения, вызванные одинаковыми действиями, должны быть одинаковыми (в частности, приложения, использующие генераторы псевдо-случайных чисел, должны иметь возможность повторять ранее созданную последовательность случайных чисел);

- Все события, приводящие к изменению состояния приложения, должны записываться TIPE.

При нарушении одного из условий воспроизведение записанной сессии невозможно. Такими нарушениями могут быть: различное начальное состояние приложения (например, отличаются загруженные данные); различное поведение приложения в ответ на одинаковые действия; изменение состояний приложения зависит от оборудования; критические ситуации в процессе воспроизведения, отсутствовавшие в процессе записи (например, нехватка дискового пространства).

Кроме как средство регрессионного тестирования системе TIPE используется также:

- Для демонстрации возможностей приложения;
- В качестве интерактивного дополнения к инструкции по эксплуатации приложения;
- Средство для визуального представления действий пользователя (например, в процессе обнаружения и исправления ошибок в приложениях).

#### 3.2 Основные свойства

Основными свойствами системы TIPE являются:

Надежное воспроизведение записанных действий – гарантируется, что при соблюдении вышеуказанных условий записанная сессия работы приложения может быть воспроизведена (независимо от того, на каких компьютерах производится запись и воспроизведение). При изменении версии приложения по возможности обеспечивается обратная совместимость: если новые функциональности не влияют на записанную последовательность изменения состояний приложения, то новая версия приложения будет успешно воспроизводить сессию TIPE, записанную на предыдущей версии приложения.

Проверка условий воспроизведения – при старте воспроизведения система проверяет соответствие

нескольких геометрических параметров (таких как разрешение экрана, размеры шрифтов, размеры окон).

Проверка совпадения последовательности событий – в процессе воспроизведения система проверяет совпадение записанных событий с событиями, генерируемыми приложением при изменении его состояния.

Операции с мышью и клавиатурой визуализируются – движения мыши визуализируются перемещением курсора, нажатия клавиш мыши и клавиатуры отображаются на экране в виде всплывающих сообщений.

Имеется возможность добавления комментариев – в процессе записи сессии возможно в любое время вставить текстовый комментарий, который будет показан во всплывающем окне в процессе воспроизведения. Комментарии могут быть изменены, удалены или вставлены так же после окончания записи, путем исправления файла, содержащего записанную сессию TPE.

При невозможности дальнейшего воспроизведения сессии (например, несовпадение начальных условий или несовпадение реакции приложения с ожидаемой реакцией) воспроизведение останавливается, и выводится сообщение об ошибке. Кроме того, в процессе записи и воспроизведения создается лог-файл, содержащий все сообщения об ошибках.

Скорость воспроизведения регулируется в широких пределах, в любой момент в процессе записи и воспроизведения процесс может быть временно приостановлен, или полностью прекращен.

Сессия TPE хранится в файле текстового формата, что допускает внесение изменений без перезаписи всей сессии, путем изменения, добавления или удаления частей сессии. Файл содержит идентификационную информацию: название, автор, дата записи, наименование и версия приложения, другая текстовая информация. Файл имеет малый размер, что допускает его пересылку любым электронным способом.

Возможна также запись новой сессии в процессе воспроизведения имеющейся. В случае остановки воспроизведения запись новой сессии может быть продолжена по другому сценарию.

Запись сессии приложения производится, даже если явной команды записи нет. Это позволяет в случае ошибки программного комплекса в дальнейшем повторить действия, приведшие к ней.

#### 4. РЕЗУЛЬТАТЫ

Разработанная и описанная в работе система тестирования постоянно используется при создании и поддержки нескольких систем синтеза реалистичных изображений и оптического моделирования, в частности, для систем, представленных в [6-7]. Использование описанного подхода позволило ускорить разработку и реализацию этих систем и одновременно снизить трудозатраты на их создание, тестирование и сопровождение.

Приведем характерные времена, затрачиваемые на тестирование программных продуктов. Автономное тестирование программных компонент проводится 1-2 раза в месяц и требует порядка суток компьютерного времени. Регрессивное тестирование продукта проводится при каждом выпуске (поддерживается несколько продуктов, поэтому частота – 2-4 раза в месяц) и занимает 2 компьютера в течение 2 суток. Тестирование

пользовательского интерфейса (ТИРЕ) проводится в среднем раз в три месяца и требует загрузки 2 компьютеров в течение 2 суток.

Безусловно, важным аспектом успешного тестирования является эффективный набор тестовых сцен. В идеале каждая сцена, приведшая к ошибке, после ее исправления должна быть включена в набор тестов. Иногда это возможно. Но при включении в тестирование всех ошибочных сцен время тестирования многократно возрастает, и такой подход становится неприменим. Поэтому разработка эффективных тестовых данных является сложной и творческой задачей.

Работа поддержана грантами РФФИ № 11-01-00870, 12-01-00560, а также фирмой Integra Inc. (Япония).

#### 5. ЛИТЕРАТУРА

- [1] Г. Майерс, Т. Баджетт, К. Сандлер Искусство тестирования программ, 3-е издание — М.: «Диалектика», 2012. — 272 с.
- [2] В.А. Дебелов, Д.С. Козлов. Верификация алгоритмов фотореалистического рендеринга кристаллов // Труды 20-й Международной Конференции по Компьютерной Графике и Зрению, Санкт-Петербург, 2010, с.238-245.
- [3] Autoit automation and scripting language, <http://www.autoitscript.com>
- [4] Test Automation FX, corporate website, <http://www.testautomationfx.com>.
- [5] SmartBear Software, corporate website, <http://smartbear.com/products/qa-tools/automated-testing-tools>.
- [6] Д.Д. Жданов, И.С. Потемин, В.А. Галактионов, Б.Х. Барладян, К.А. Востряков, Л.З. Шапиро. Спектральная трассировка лучей в задачах построения фотореалистичных изображений // "Программирование", № 5, 2011, с. 13-26.
- [7] A. Ignatenko, B. Barladian, K. Dmitriev, S. Ershov, V. Galaktionov, I. Valiev, A. Voloboy. A Real-Time 3D Rendering System with BRDF Materials and Natural Lighting // Proc. 14th International Conference on Computer Graphics and Vision GraphiCon-2004, Russia, Moscow, September 6 -10, 2004, p. 159-162.

#### Abstract

This paper describes the testing technology of computer graphics software systems being developed in the Keldysh Institute of Applied Math RAS. The complexity of products in development, the need to produce a large number of versions (50-70 versions per year) required to develop its own testing system, which effectively allows identifying errors and correcting them before the release of the product. Testing of computer graphics software has its own specifics. These specifics are addressed in the paper.

**Keywords:** *realistic rendering software, software system testing, regression testing, comparison of images*

#### Authors:

Alexey G. Voloboy, PhD, senior researcher, KIAM RAS, E-mail: [voloboy@gin.keldysh.ru](mailto:voloboy@gin.keldysh.ru)

Eugeny Yu. Denisov, researcher, KIAM RAS.

E-mail: [eed@gin.keldysh.ru](mailto:eed@gin.keldysh.ru)

# Интерактивный рендеринг при помощи сферических дизайнов для низкочастотного окружающего освещения

Свиштунов Сергей

Механико-математический факультет

Тульский Государственный Университет, Тула, Россия

sssboxmail@gmail.com

## Аннотация

Рассматривается метод интерактивного рендеринга на основе уравнения Кэджиуа при помощи сферических дизайнов (МСД). Исследуется случай низкочастотного окружающего освещения, задаваемого картой окружения. Результатом является первичное отраженное освещение. МСД решает те же задачи, что и известный метод Precomputed Radiance Transfer (PRT) [7]. Преимущество МСД проявляется в случае двунаправленной функции отражательной способности (ДФОС), не являющейся диффузной (например, ламбертовской), в частности, ДФОС Фонга.

**Ключевые слова:** *real-time rendering, illumination, spherical design, spherical harmonic, Kajiya rendering equation, low-frequency light, precomputed radiance transfer (PRT), BRDF, ambient occlusion, cubemap*

## 1. ВВЕДЕНИЕ

В настоящее время актуальной является задача физически обоснованного интерактивного (real-time, несколько десятков и более кадров в секунду) рендеринга сцены, состоящей из непрозрачной модели и сложного удаленного окружающего освещения, задаваемого картой окружения. Интерактивный рендеринг широко используется в компьютерных играх, научной визуализации, компьютерных тренажерах, кино, рекламе и т.п. Поэтому разработка эффективных методов его реализации является важной проблемой современной компьютерной 3D-графики.

Особенностью рассматриваемой задачи является учет сложного освещения в real-time. В качестве источника освещения выступает карта окружения (Cubemap), в отличие от наиболее стандартной ситуации real-time рендеринга, когда падающее освещение задается набором точечных источников. В качестве основной двунаправленной функции отражательной способности рассматривается ДФОС Фонга, потому что уже в этом, казалось бы, не сложном случае возникают проблемы у известного метода решения той же задачи. Рассматривается непрозрачная, статическая (без анимации) 3D-модель с однородными свойствами материала (в каждой точке модели ДФОС имеет один и тот же вид). Динамически изменяющиеся параметры сцены: положение камеры, вращение модели ил вращение карты окружения.

Эту задачу решает известный метод PRT, основанный на аппроксимации уравнения рендеринга для первичного (без учета переотражений от поверхности модели) отраженного освещения (Reflection equation) при помощи сферических гармоник [11, 9, 5]. Однако для случая ДФОС Фонга и других, где есть зависимость от вектора, направленного на

наблюдателя, возникают сложности, связанные с необходимостью использования больших объемов предрасчитанной информации (для уменьшения этих объемов используются специальные методы сжатия, например LPCA или CPSCA [3,7]). Соответственно само вычисление этой информации занимает значительное время (см. табл. 1).

Нами развивается метод интерактивного рендеринга на основе аппроксимации уравнения рендеринга Кэджиуа при помощи взвешенных сферических дизайнов [1], рассчитываемых по произвольным весовым функциям, в том числе задаваемых Cubemap. Этот метод позволяет получить простые расчетные формулы для случая ДФОС Фонга и не использовать большие объемы предрасчитанной информации.

## 2. ПРЕДЫДУЩИЕ РАБОТЫ

Задача интерактивного рендеринга, когда в качестве источника освещения выступает карта окружения (снимок окружающего пространства), исследовалась многими авторами. В 2002 году P. Sloan, J. Kautz и J. Snyder [11] предложили технику интерактивного рендеринга сцены со сложным освещением при помощи некоторой заранее рассчитанной информации (сферических коэффициентов Фурье освещения и функции транспорта модели, равной произведению функции видимости на ДФОС Ламберта). Она получила название метода PRT. При этом моделирование освещения основывается на уравнении рендеринга Кэджиуа. В 2003 году R. Green в статье [5] приводит подробные сведения, необходимые для реализации метода PRT. Также эти вопросы освещены в диссертации 2002 года R. Ramamoorthi [4]. После этих работ метод PRT активно развивался. В 2009 году R. Ramamoorthi [5] подробно изложил историю отмеченных задач и достигнутые на тот момент результаты по проблематике интерактивного рендеринга. Отметим, что метод PRT реализован в DirectX, но только в диффузном случае для ДФОС Ламберта. В 2007 году авторы статьи [1] для моделирования освещения предложили использовать сферические дизайны с равными весами и единичной весовой функцией. Нами эта идея развивается за счет применения взвешенных дизайнов. Кроме того используется другая более обоснованная аппроксимация уравнения рендеринга Кэджиуа.

## 3. PRT

Методы PRT и МСД основаны на аппроксимации Reflection equation

$$B_p(v, g) = \frac{1}{4\pi} \int_{S^2} L(gx) V_p(x) \rho_p(x, v) dx.$$

Здесь  $S^2 = \{x \in \mathbb{R}^3 : |x| = 1\}$  – единичная евклидова сфера,  $B_p(v, g)$  – функция яркости отраженного освещения (итоговое изображение) в точке 3D-модели  $P$  по направлению (на камеру) единичного вектора  $v \in S^2$ ,  $g$  – вращение карты окружения относительно базового положения (задаваемое матрице размерности 3),  $L(x)$  – функция яркости удаленного окружающего освещения (задается Cubemap),  $V_p(x)$  – функция видимости (равная 0, если луч, выпущенный из точки 3D-модели  $P$  по направлению  $x$ , пересекает 3D-модель, и 1 иначе),  $\rho_p(x, v)$  – двунаправленной функции отражательной способности (ДФОС) в точке 3D-модели  $P$  с учтенным ламбертовским множителем  $(n_p \cdot x)_+$ ,  $n_p \in S^2$  – вектор нормали к поверхности 3D-модели в точке  $P$ ,  $(t)_+ = \max(0, t)$ ,  $(x, y) = x \cdot y = x_1 y_1 + x_2 y_2 + x_3 y_3$  – скалярное произведение векторов  $x = (x_1, x_2, x_3)$ ,  $y = (y_1, y_2, y_3) \in \mathbb{R}^3$ .

Основная проблема заключается в том, чтобы в интерактивном режиме для каждого кадра максимально быстро и качественно вычислить  $B_p(v, g)$  во всех вершинах 3D-модели. При этом  $v$  и  $g$  могут быть произвольными.

Для решения проблемы необходимо аппроксимировать уравнение конечными суммами небольшого порядка, вычисление которых хорошо портируется на архитектуру современных GPU.

В методе PRT используется аппроксимация интеграла уравнения рендеринга при помощи разложения функции  $L$  в ряд Фурье по сферическим гармоникам [11, 5]. Для получения в расчетной формуле небольшого числа слагаемых необходимо, чтобы окружающее освещение было низкочастотным. Это освещение, которое хорошо приближается в среднеквадратичном смысле частичной суммой ряда Фурье малого порядка. В работе [9] упоминается, что в большинстве практических случаев достаточно использовать порядок 2–5, что соответствует 4–25 коэффициентам Фурье. Для расчетов коэффициентов Фурье используется метод Монте-Карло.

Расчетная формула в методе PRT для заданной точки модели представляет собой сумму произведений коэффициентов Фурье функции транспорта модели на коэффициенты освещения. Коэффициенты Фурье транспорта являются основной предрасчитанной информацией в PRT. Они сохраняются в каждой вершине модели. В диффузном случае, когда нет зависимости от направления на камеру, их немного (например, как и выше, 9–25). Если такая зависимость присутствует, как, например, в зеркальном случае ДФОС Фонга, то уже необходимо сохранять большой набор таких коэффициентов в зависимости от дискретных направлений на камеру. Частично проблему уменьшения памяти решают применением сжатия LPCA. Однако это не всегда эффективно. Далее описывается метод сферических дизайнов, в котором не требуется сохранять большой объем данных в

каждой вершине в случае ДФОС, зависящей от направления на камеру.

#### 4. МЕТОД СФЕРИЧЕСКИХ ДИЗАЙНОВ

МСД базируется на аппроксимации интеграла уравнения рендеринга при помощи взвешенных сферических дизайнов. Взвешенный сферический дизайн порядка  $s$  с весовой функцией  $W$  – это узлы  $x_{mn}$  и веса  $l_{mn}$  оптимальной (или близкой к ней) кубатурной формулы на сфере, определяемые таким образом, чтобы формула

$$\frac{1}{4\pi} \int_{S^2} f(x) \omega(x) dx = \sum_{v=1}^N \lambda_{\omega v} f(x_{\omega v})$$

была точна для всех многочленов  $f$  степени  $s$  [2, 6].

Построение дизайнов в общем случае является сложной математической проблемой. Нами разработаны методы построения взвешенных сферических дизайнов небольшого порядка с произвольной весовой функцией, задаваемой картой окружения или аналитически.

Аппроксимация интеграла в МСД начинается с учета функции видимости. Ее полноценный учет является сложной задачей. Однако во многих практически важных случаях функцию видимости без визуального ухудшения результатов можно заменить коэффициентом затенения  $V_{p0}$ , когда приближенно полагают

$$B_p(v, g) = \frac{V_{p0}}{4\pi} \int_{S^2} L(x) \rho_p(x, v) dx.$$

Существуют разные формулы для определения коэффициента затенения  $V_{p0}$ , например  $V_{p0} = \frac{1}{2\pi} \int_{S^2} V_p(x) dx$  (здесь интегрирование для непрозрачной модели фактически ведется по полусфере, определяемой нормалью в точке поверхности  $P$ ). Такой подход близок к известному методу Ambient Occlusion [8]. Коэффициенты затенения являются предрасчитанной информацией в МСД. Они вычисляются и сохраняются для каждой вершины модели. В более общем случае можно использовать аппроксимацию функции видимости не константой, а полиномом небольшой степени. В этом случае в каждой вершине, также как и PRT для диффузного случая, сохраняются небольшое число коэффициентов (например, 4–25), которые также можно сжать при помощи LPCA.

Важно, что для расчета коэффициентов затенения можно использовать возможности OpenCL/CUDA. Основной затратной операцией является определение пересечений лучей, выпущенных из вершин модели с ее полигонами (мешем). В случае параллельной реализации нами разработано два подхода. Первый подход использует параллельный обход полигонов меша с использованием в ядре (kernel) разделяемой (local) памяти. Здесь ядро – это программа, выполняемая в параллельном режиме на GPU, а разделяемая память – это быстрый тип памяти, используемый для сокращения обращений к медленной глобальной памяти графического процессора. Второй подход основан на применении карт или буферов глубины (Depth Buffers). Буфер глубины представляет собой двумерный массив данных (текстуру), где сохраняется расстояние от наблюдателя до

поверхности изображаемого объекта. В этом случае на CPU осуществляется обход вершин, а уже в каждой вершине на GPU при помощи средств OpenGL/DirectX осуществляется рендеринг сцены в текстуру карты глубины, по которой быстро вычисляется коэффициент затенения. Описанный процесс близок к известной технологии Shadow Maps [4].

Итоговая расчетная формула МСД имеет вид

$$\tilde{B}_p(v, g) = V_{p0} \sum_{v=1}^{N_p} \sum_{\mu=1}^{N_L} \lambda_{\rho v} \lambda_{L\mu} K_s(g_{n_p v} \cdot x_{\rho v} \cdot x_{L\mu}),$$

где  $\{\lambda_{\rho v}, x_{\rho v}\}_{v=1}^{N_p}$  –  $s$ -дизайн с весовой функцией, основанной на ДФОС  $\rho$ ,  $\{\lambda_{L\mu}, x_{L\mu}\}_{\mu=1}^{N_L}$  –  $s$ -дизайн с весовой функцией  $L$ ,  $K_s$  – сферическое ядро Дирихле порядка  $s$ ,  $g_{n_p v}$  – вращение, переводящее уравнение рендеринга в локальную систему координат точки  $P$  ориентированной по нормали (в это вращение так же входит вращение  $g$ ).

Вычисление по этой формуле осуществляется в реальном времени в шейдере. Как уже упоминалось, коэффициенты затенения рассчитываются заранее и сохраняются для каждой вершины. Так же необходимые дизайны могут быть вычислены заранее и сохранены. Тогда в шейдере остается вычислить вращение, сферическое ядро Дирихле (задается аналитически) и итоговую сумму. Все эти операции не являются затратными.

Для описанного метода сферических дизайнов реализовано программное обеспечение и демонстрационное приложение с использованием языка программирования Python и технологий OpenGL, OpenGL, NVIDIA CUDA.

## 5. РЕЗУЛЬТАТЫ

	МСД	PRT (см. [11])
Модель (количество вершин)	130k	50k
Память (в каждой вершине модели)	1–9 float (коэффициент затенения)	(1–9) x 25 x 25 float (матрица транспорта)
Предварительные расчеты	35 мин при использовании буферов глубины 10 мин при использовании CUDA	2.5 часа
Результат (FPS)	100–400	3.6

Таблица 1: Сравнение методов МСД и PRT.

Разработан метод физически обоснованного интерактивного рендеринга для низкочастотного окружающего освещения, дающий визуальные результаты, аналогичные известному методу PRT. При этом он имеет преимущества для случая недиффузной ДФОС, требует меньше времени на предварительные расчеты и памяти для хранения

предрасчитанной информации, имеет более простую расчетную формулу и, как следствие, шейдерную реализацию. В табл. 1 приводится сравнение МСД и PRT для случая низкочастотного окружающего освещения порядка 2 в случае ДФОС Фонга при схожих характеристиках компьютеров и видеокарт.

Подробные результаты работы приложения можно увидеть на видео презентации по ссылке <http://goo.gl/CeEWO>, а также на рис. 1.



Рисунок 1: Результат работы метода сферических дизайнов

## 6. ЗАКЛЮЧЕНИЕ И БУДУЩИЕ РАБОТЫ

Отметим возможные улучшения предложенных алгоритмов.

В рассматриваемой задаче динамически меняющимися компонентами являются обзор наблюдателя, окружающее освещение, материал и меш модели.

Динамическое изменение положение наблюдателя полностью учитывается в текущей реализации МСД. Для изменения в реальном времени окружающего освещения и материала модели требуется строить в реальном времени взвешенные дизайны. Для этого можно воспользоваться возможностями современных GPU (технологиями CUDA/OpenCL) и адаптировать под них алгоритмы расчета дизайнов, что вполне реально.

Для использования МСД в случае динамической (анимированной) модели требуется в интерактивном режиме вычислять коэффициенты затенения. Для этого можно использовать ускоряющие структуры (kd-деревья, пространственное хеширование) на GPU с применением технологий CUDA/OpenCL. Сейчас в разработанном программном обеспечении используется технология CUDA, но в виду определенной сложности не применяются ускоряющие структуры. Так же можно адаптировать подход описанный в [10], где для быстрого вычисления коэффициентов Фурье функции видимости используется аппроксимация модели набором сфер.

## 7. ССЫЛКИ

- [1] Горбачев Д.В., Иванов В.И., Странковский С.А. Моделирование освещения в интерактивной графике при помощи сферических дизайнов // Изв. ТулГУ. Сер. Естественные науки. 2007. Т. 1. Вып. 1. С. 17–36.
- [2] Мысовских И.П. Интерполяционные кубатурные формулы. М.: Наука, 1981.
- [3] Gassenbauer V. Improving Performance and Accuracy of Local PCA <http://cgg.mff.cuni.cz/~jaroslav/papers/2011-pg-lpca/2011-gassenbauer-lpca.pdf>
- [4] Gary King. Shadow Mapping Algorithms [ftp://download.nvidia.com/developer/presentations/2004/GPU\\_Jackpot/Shadow\\_Mapping.pdf](ftp://download.nvidia.com/developer/presentations/2004/GPU_Jackpot/Shadow_Mapping.pdf)
- [5] Green R. Spherical Harmonic Lighting: The Gritty Details. <http://www.research.scea.com/gdc2003/spherical-harmonic-lighting.html>
- [6] Hardin R.H., Sloane N.J.A. McLaren's Improved Snub Cube and Other New Spherical Designs in Three Dimensions // Discrete and Computational Geometry. 1996. V.15. P. 429–441. <http://www.research.att.com/~njas>
- [7] Kambhatla N., Leen T. Dimension reduction by local principal component analysis // Neural Comput. 9. 1997, 1493–1516 [http://www.ohsu.edu/xd/education/schools/school-of-medicine/departments/basic-science-departments/biomedical-engineering/people/upload/kambhLeen\\_NC97-DimRed.pdf](http://www.ohsu.edu/xd/education/schools/school-of-medicine/departments/basic-science-departments/biomedical-engineering/people/upload/kambhLeen_NC97-DimRed.pdf)
- [8] Kontkahan J., Samuli L. Ambient occlusion fields. <http://www.tml.tkk.fi/~janne/aofields/aofields.pdf>
- [9] Ramamoorthi R. Precomputation-Based Rendering // Foundations and Trends in Computer Graphics and Vision. 2009. V. 3, № 4. P. 281–369. <http://www.cs.berkeley.edu/~ravir/>
- [10] Ren Z., Wang R., Snyder J. Real-time Soft Shadows in Dynamic Scenes using Spherical Harmonic Exponentiation. <http://research.microsoft.com/en-us/people/bainguo/p977-ren.pdf>

- [11] Sloan P., Kautz J., Snyder J. Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments. <http://web4.cs.ucl.ac.uk/staff/j.kautz/publications/prtSIG02.pdf>

## Об авторах

Свистунов Сергей – аспирант ММ ТулГУ. Его адрес: sssboxmail@gmail.com.

# Эффективная и практичная реализация фотонных карт на GPU

Владимир Фролов<sup>(1,2)</sup>, Александр Харламов<sup>(2)</sup>, Владимир Галактионов<sup>(1)</sup>, Константин Востряков<sup>(2)</sup>

(1) Институт Прикладной Математики им. Келдыша Российской Академии Наук; (2) Nvidia;

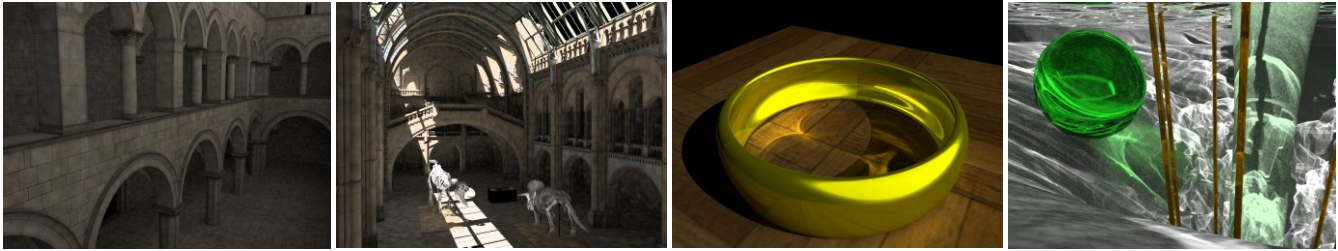


Рисунок 1. Изображения, полученные после 10 проходов (и соответствующие 10 миллионам фотонов) алгоритма прогрессивных фотонных карт. Время рендеринга – от 4 до 8 секунд в разрешении 1024x768 на GTX680.

## Abstract

This paper introduce an effective and simple photon mapping implementation for GPU via constructing Multiple Reference (MR) octrees. Although MR-octrees are hierarchical structures, we successfully ignore their hierarchical nature and present an approach with plain construction, compact data layout and stack-less traversal.

**Keywords:** Multiple Reference Octree, GPU, Photon Mapping

## Аннотация

В данной статье предлагается эффективная и исключительно простая реализация алгоритма фотонных карт на GPU на основе окто-деревьев с множественными ссылками. Предлагаемый алгоритм построения дерева состоит из 4 шагов, использует линейные структуры данных и только 2 параллельных примитива для построения дерева. Результирующая структура позволяет производить эффективный поиск ближайших фотонов в окто-дереве с заданным радиусом сбора без использования стека.

**Keywords:** Фотонные карты, множественные ссылки.

## 1. ВВЕДЕНИЕ

Окто-дерево является достаточно популярной ускоряющей структурой в силу своей регулярности и относительно небольшой глубины. Классическое окто-дерево рекурсивно разбивает трехмерное пространство на 8 частей, сохраняя в узлах (как на рис. 1, слева) ссылки на объекты, которые целиком находятся внутри узла.

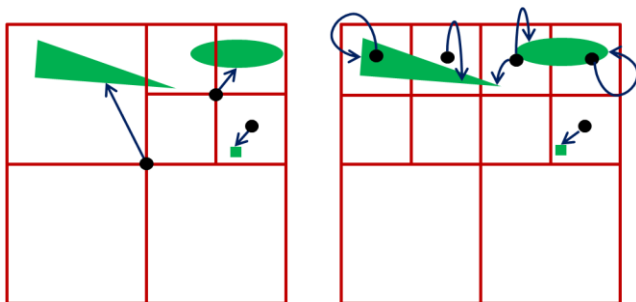


Рисунок 2: Классическое окто-дерево (слева) и окто-дерево с множественными ссылками (справа).

Окто-дерево с множественными ссылками представляет из себя модификацию классического варианта, в котором все ссылки хранятся только в листьях и на каждый объект ссылаются все листья, имеющие с данным объектом геометрическое пересечение (рис. 1, справа). Этот факт позволяет реализовать поиск в дереве без использования рекурсии или стека, поскольку для поиска ближайших объектов в заданной точке трехмерного пространства достаточно спуститься от корня дерева к листу; каждый лист уже содержит список объектов, которые нужно перебирать последовательно.

## 2. ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ

### 2.1 Окто-дерево с множественными ссылками

Традиционный подход к построению окто-деревьев с множественными ссылками наиболее часто используется в технике кэширования освещенности [1]. Алгоритм позволяет добавлять новые элементы в дерево, постепенно достраивая его. При добавлении новые элементы спускаются вниз по дереву во все узлы, с которыми они имеют геометрическое пересечение.

### 2.2 Построение окто-деревьев на GPU

Алгоритмы построения окто-деревьев на GPU используют факт взаимосвязанности окто-дерева и трехмерной Z-кривой, которая получается в результате вычисления Z-индекса (или кода Мортонa) [2]. Зафиксируем некоторую глубину дерева 'k'. Пусть у нас имеется такой набор точек в трехмерном пространстве, что в каждый узел построенного в будущем окто-дерева попадет не более чем 1 точка. Если вычислить для точек коды Мортонa и затем отсортировать точки по этим кодам, мы получим массив, эквивалентный окто-дереву, где в каждом листе хранится ровно по одному элементу (одной точке). Выполняя далее поиск в трехмерном пространстве в произвольной точке (x,y,z), мы вычисляем ее код Мортонa и, применив бинарный поиск для полученного кода Мортонa (ZInxed(x,y,z)), находим листовую узел окто-дерева, в который попала точка (x,y,z).

В работе [3] было впервые замечено соответствие между Z-кривой и окто-деревом и использовано свойство отбрасывания младших битов Z-индекса для получения идентификаторов старших узлов. Окто-дерево в [3]

представлялось набором массивов  $L_1..L_k$  размером  $8^i$  элементов для каждого уровня глубины  $i$  (т.е. несуществующие узлы хранились в памяти на регулярной сетке). Вопрос хранения каких-либо элементов данных в окто-дереве в работе [3] не рассматривается. Таким образом, построенное окто-дерево не может быть использовано непосредственно для пространственного поиска элементов, а может только отвечать на вопрос имеются ли искомые элементы в некотором объеме или нет.

В работе [4] окто-дерево было использовано для построения поверхности из набора точек при помощи алгоритма марширующих кубов. Алгоритм построения использовал серию из параллельных операций уплотнения и сортировки. В отличие от работы [3] Zhou и др. сохраняют только существенные узлы в памяти, а также сохраняют списки точек на всех уровнях дерева. При этом в [4] было замечено, что имея отсортированный массив точек по их Z-индексу в узле, достаточно сохранять лишь 2 смещения, обозначающих начало и конец последовательности точек для данного узла в отсортированном массиве. Причем при переходе от уровня  $k$  на уровень  $k-1$  это свойство сохраняется и точки пересортировывать не нужно. Достаточно пересчитать указатели, соответствующие началу и концу последовательности.

Если в рассмотренных ранее работах только отдельные уровни дерева строились параллельно, то в работе [5] был предложен метод, использующий массивный параллелизм GPU при построении всего дерева. Основная идея алгоритма заключается в том, чтобы пронумеровать все узлы дерева при помощи индексов таким образом, чтобы индекс родительского узла всегда являлся префиксом по отношению к индексам его дочерних узлов, если рассматривать индексы как битовые строки. Тогда, отсортированный массив индексов будет эквивалентен дереву. При этом элементы ненулевого размера (треугольники в работе [5]) были использованы только при построении BVH деревьев, поскольку только в этом случае размер элемента не имеет значения, а ограничивающий бокс примитива используется для расчета ограничивающего бокса листа уже после построения дерева и в самом построении участия не принимает. Это возможно благодаря тому, что в BVH деревьях различные узлы дерева могут пересекаться друг с другом своими ограничивающими объемами.

В предложенных ранее подходах построения окто-деревя в качестве примитивов использовались точки – элементы, не имеющие размера. Так как каждая точка при этом условии попадает не более чем в 1 лист, это позволяло использовать тот факт, что искомые элементы данных всегда лежат последовательно в отсортированном по их Z-индексам массиве [4]. Однако, в случае элементов, имеющих размер, это условие нарушается, поскольку один и тот же элемент может попасть в несколько узлов. Для решения этой проблемы мы используем окто-дерева с множественными ссылками и предложенный нами алгоритм построения таких деревьев на GPU.

### 2.3 Фотонные карты на GPU

Фотонные карты на GPU – довольно популярная тема для исследований. Одна из основных трудностей при реализации фотонных карт – эффективное построение ускоряющих структур для последующего поиска

ближайших фотонов. Нам необходим метод, который бы позволял быстро находить ближайшие фотоны в заданном радиусе вокруг заданной точки трехмерного пространства. В силу ограниченного места мы отметим лишь наиболее универсальные и практичные подходы.

В [6] был предложен эффективный алгоритм построения kd-деревьев для поиска  $k$ -ближайших фотонов; [7] использовал BVH-дерево и фиксированный радиус сбора. Основным недостатком подходов, использующих классические деревья, является рекурсивный алгоритм обхода дерева при сборе освещенности. Такой алгоритм неэффективен в основном по причине высокой степени дивергентности потоков на GPU и использования локальной памяти.

В работе [8] предложен метод пространственных хэш-таблиц на основе “хэширования кукушки”. Авторы не исследовали метод в применении к алгоритму фотонных карт, и в работе отмечается, что для различных применений могут потребоваться различные хэш-функции.

В работе [9] использовали хэш-таблицы для реализации фотонных карт, причем, для того чтобы исключить коллизии во время поиска, дополнительно производили сортировку пар (ключ, значение) по ключу. При этом поиск ближайших фотонов в точке  $(x,y,z)$  производился в ячейке  $\text{hash}(x,y,z)$  и также в 27 соседних ячейках. [9] отмечает, что для глобальной фотонной карты по сравнению с kd-деревом из работы [6] этот метод не дает ускорения сбора с ростом искомого числа ближайших фотонов (т.е. с ростом радиуса сбора). Это может быть объяснено потерей производительности на ветвлениях из-за наличия вложенного цикла, что необходимо для поиска в 27 соседних ячейках. Мы покажем, что в нашем подходе такой потери нет, поскольку сбор освещенности со всех фотонов состоит целиком из 1 цикла.

В работе [10] представлен эффективный подход на основе лексикографической сортировки координат фотонов. Над отсортированным массивом строилась воксельная ускоряющая структура на основе 3D текстур. Недостатком этого подхода является высокий расход памяти.

В работе [11] предложен упрощенный подход на основе пространственных хэш-таблиц. При возникновении коллизии авторы предлагают стохастически сохранять только 1 фотон на элемент таблицы. Данный подход замедляет трассировку фотонов (в 3-4 раза в соответствии со средним числом коллизий), что может быть нецелесообразно, т.к. при сложных условиях освещения трассировка фотонов будет занимать много больше времени, чем построение ускоряющих структур при помощи любого из известных методов.

В работе [12] впервые высказана идея построения ускоряющей структуры на GPU при помощи сортировки индексов ссылок (а не индексов центров объектов). Размер ячейки в [12] был в 2 раза больше размера самого большого объекта, что гарантирует не более 8 ссылок на объект. Такое ограничение размера ячейки может негативно сказаться на скорости поиска при разных размерах объектов.

Поскольку операция сбора освещенности для 1 порции фотонов может выполняться множество раз, необходимо ускорять сбор освещенности настолько быстро, насколько это возможно. Однако, при этом метод можно считать практичным, если время построения ускоряющей структуры



меньше времени трассировки фотонов и не замедляет ее саму.

### 3. ПРЕДЛОЖЕННЫЙ ПОДХОД

Мы позиционируем наш метод, как более эффективный по скорости сбора освещенности на GPU, чем методы [9], [10] и [11] и при этом не замедляющий трассировку фотонов. Чтобы избежать рекурсивного обхода и собрать ближайшие фотоны в заданном радиусе, мы используем окто-дерево с множественными ссылками. Принимая во внимание то, что радиус сбора фиксирован, мы можем рассматривать фотоны как сферы с радиусом, равным радиусу сбора освещенности. И строить окто-дерево, содержащее в качестве объектов не точки, а сферы. Этот известный факт используется при построении окто-деревьев с множественными ссылками для кэша освещенности [1].

#### 3.1 Построение окто-дерева

В начале работы наш алгоритм выбирает некоторый фиксированный уровень дерева  $k$ , для которого определяется размер его листьев. Уровень  $k$  выбирается так, чтобы размер листового узла был меньше или равен среднему размеру объектов. При этом все трехмерное пространство может быть разбито на воксели в виде сетки. Каждый воксел – лист будущего окто-дерева. Однако не все листья будут существовать в действительности, а только те, для которых будут иметься ссылки на объекты. Используя тот факт, что в окто-дереве с множественными ссылками каждый лист ссылается на все объекты, которые имеют с ним пересечение, мы можем сразу сгенерировать массив всех ссылок, не строя пока что само дерево, а вычислив для каждого объекта множество ссылок на него (рис.3).

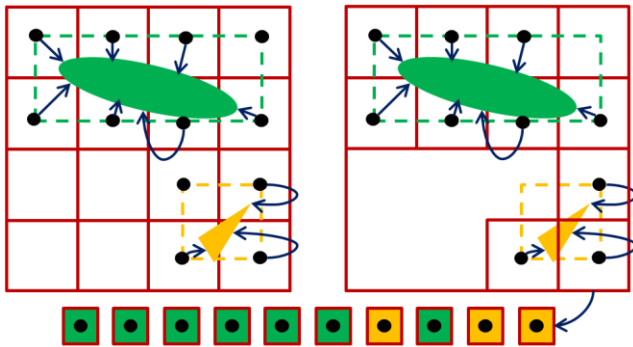


Рисунок 3: Регулярная сетка (слева); существующие ссылки-воксели (справа); массив сгенерированных ссылок (внизу).

Оказывается, что если отсортировать массив всех ссылок (всех ссылок для всех объектов) по  $Z$ -индексу узлов, на которые они указывают, мы уже получим практически готовое окто-дерево. Ниже представлен предлагаемый алгоритм:

```
refs := AppendRefs (Voxelize (objects)); -- 1
refs.sort (compareByZIndex); -- 2
nodes := AppendUniqueRefs (refs); -- 3
nodes.sort (compareByZIndex); -- 4
```

Шаг номер 1 – генерация множества всех ссылок, описанная ранее. Все ссылки добавляются в общий буфер при помощи параллельной операции `append`. Такая операция может быть

реализована через атомарные операции в CUDA или, что более эффективно, через функциональность графических API ‘Transform Feedback’ (OpenGL) и ‘Stream Out’ (DirectX).

Шаг номер 3 представляет из себя выделение множества узлов из множества ссылок и является прямым аналогом шага 4 ‘remove the duplicates using parallel compaction’ из работы [5]. Мы заменили операцию уплотнения (compaction) на последовательность из операций (append, sort), чтобы не выделять дополнительный массив, размер которого равен количеству ссылок. Это позволяет экономить память.

Ниже представлены типы данных, используемые алгоритмом.

```
type Reference is record
  zIndex      : integer; -- Morton code
  dataIndex   : integer; -- offset to user data
end record;
type Node is record
  zIndex      : integer; -- Morton code
  refStart    : integer; -- first ref
  refEnd      : integer; -- last ref
end record;
```

Наша реализация использует способ хранения данных SOA (Structure Of Arrays), благодаря чему мы имеем возможность освобождать неиспользуемые массивы после построения дерева. Так мы поступили с массивом для хранения  $Z$ -индексов ссылок и в результирующей структуре ссылка представлена, таким образом, одним 32 битовым числом.

#### 3.2 Сбор фотонов

Алгоритм сбора ближайших фотонов не использует стек и является прямым аналогом выборки из кэша освещенности на основе окто-деревьев с множественными ссылками [1]. Его основное преимущество заключается в том, что после спуска от корня к листу дерева искомые фотоны итерировуются единственным циклом, перебирающим список всех ссылок на фотоны.

### 4. РЕЗУЛЬТАТЫ

В зависимости от радиуса сбора (и как следствие различного числа попадающих в сферу фотонов) наш алгоритм (назовем его ‘фотонные карты на основе окто-деревьев со множественными ссылками’) ускоряет сбор освещенности в 2-5 раз по сравнению с классическим окто-деревом (рис. 4, таблица 1). В отличие от [9] наш алгоритм продолжает значительно выигрывать у рекурсивного решения при увеличении числа собираемых фотонов. Поскольку в работе [11] не было представлено анализа скорости сбора освещенности, мы проводим аналогию с работой [9] и заключаем, что наш подход более эффективен на GPU из-за наличия только одного цикла итерирования фотонов в отличие от [9], [10] и [11], где необходим еще 1 вложенный цикл.

Мы сравнивали время построения нашего дерева с обычным окто-деревом из работы [5]. Так как известно, что алгоритм [5] ограничен производительностью сортировки, за наилучшее время построения обычного окто-дерева по методу [5] мы взяли время сортировки 1 миллиона точек.

Полученное время построения ожидаемо, так как в рассматриваемом нами типе окто-дерева на 1 фотон может сгенерироваться от 10 до 20 ссылок; поэтому время построения окто-дерева с множественными ссылками в 10-20 раз больше времени построения обычного окто-дерева на GPU на основе подхода [5]. Однако мы считаем, что данное замедление не имеет эффекта на конечное время рендеринга, поскольку дерево строится 1 раз после этапа трассировки фотонов. Причем трассировка фотонов, даже с учетом оптимизированной реализации трассировки лучей, сравнимой по производительности с [13], занимает в 5-10 раз больше времени, чем построение дерева (таблица 2, колонка “ph-trace”).

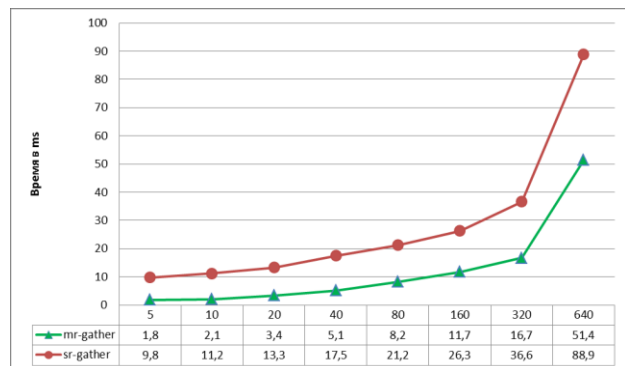


Рисунок 4. Зависимость времени сбора в миллисекундах (ось ‘y’) для 1 миллиона первичных лучей на сцене “Спонза” от среднего числа фотонов участвующих в сборе (ось ‘x’). Увеличение этого числа эквивалентно увеличению радиуса сбора или суммарному количеству обрабатываемых фотонов.

Сцена	mr-build	sr-build	mr-gather	sr-gather
Спонза	46 мс	3.1 мс	3.1 мс	14 мс
Музей	52 мс	3.2 мс	4.4 мс	16 мс
Кольцо	24 мс	3.0 мс	5.6 мс	9.9 мс
Вода	26 мс	3.0 мс	8.3 мс	18 мс

Таблица 1. Сравнение скорости построения окто-дерева с множественными ссылками (mr-build) со скоростью построения обычного окто-дерева (sr-build). Соответствующее время сбора для 1 миллиона первичных лучей по нашему методу (mr-gather) и с использованием обычного окто-дерева (sr-gather).

Сцена	ссылки	память	ph-trace	ph-trace (opt)
Спонза	10M	115MB	155 мс	22 мс
Музей	11M	128MB	470 мс	26 мс
Кольцо	2.0M	23MB	174 мс	27 мс
Вода	2.4M	25MB	300 мс	16 мс

Таблица 2. Время трассировки фотонов. Количество ссылок в окто-дерева (ссылки), занимаемая алгоритмом построения память (память), реальное время трассировки фотонов (ph-trace) и гипотетически оптимальное время трассировки фотонов (ph-trace (opt)).

Последняя колонка в таблице 2 получена как реальное время трассировки фотонов “ph-trace”, умноженное на отношение числа аккумулированных фотонов к числу фотонов, выпущенных из источника света. Это позволяет нам оценить время трассировки фотонов при применении такой гипотетической оптимизации, при которой все фотоны испускаются из источника света в правильных направлениях и всегда сохраняются в фотонной карте. Таким образом, мы можем отметить, что наше время построения окто-дерева сравнимо с недостижимым на практике теоретическим минимумом времени трассировки фотонов (таблица 2, колонка “ph-trace (opt)”), и поэтому мы делаем вывод о том, что для фотонных карт наш алгоритм имеет достаточную производительность построения дерева. Все замеры производились в разрешении 1024x768 на GTX680. Программная реализация выполнена с использованием технологии CUDA и библиотеки thrust.

## 5. ЛИТЕРАТУРА

- [1] Křivánek, J., Gauthron, P., Ward, G., Jensen, H. W., Christensen, P. H., and Tabellion, E. 2008. *Practical global illumination with irradiance caching*. In ACM SIGGRAPH 2008 Classes (Los Angeles, California, August 11 - 15, 2008). SIGGRAPH '08. ACM, New York, NY, 1-20.
- [2] Morton, G. M. (1966), *A computer Oriented Geodetic Data Base; and a New Technique in File Sequencing*, Technical Report, Ottawa, Canada: IBM Ltd.
- [3] Prekshu A., Rhushabh G., Sharat C., Srinivas A. Fast, Parallel, GPU Based space filling curves and octrees. 2008. Department of Computer Science & Engineering, IIT Bombay.
- [4] Zhou, K., Gong, M., Huang, X., and Guo, B. 2010. *Data-parallel octrees for surface reconstruction*. IEEE Transactions on Visualization and Computer Graphics.
- [5] Tero Karras. *Maximizing parallelism in the construction of BVHs, octrees, and k-d trees*. In *Proceedings of the Fourth ACM SIGGRAPH / Eurographics conference on High-Performance Graphics (EGGH-HPG'12)*, Eurographics Association, 2012, Aire-la-Ville, Switzerland, Switzerland, 33-37.
- [6] Kun Zhou, Qiming Hou, Rui Wang, Baining Guo. Real-Time KD-Tree Construction on Graphics Hardware. *ACM Transactions on Graphics (SIGGRAPH Asia 2008)*
- [7] Fabianowski D., Dingliana J. *Interactive Global Photon Mapping* In Computer Graphics Forum 28(4), [EGSR 2009], pages 1151-1159, July 2009.
- [8] Alcantara, D. A., Sharf, A., Abbasinejad, F., Sengupta, S., Mitzenmacher, M., Owens, J. D., Amenta, N. 2009. Realtime parallel hashing on the gpu. In SIGGRAPH Asia '09: ACM SIGGRAPH Asia 2009 papers, ACM, New York, NY, USA, 1-9.
- [9] Fleisz M. Photon Mapping on the GPU. Master of Science Computer Science School of Informatics University of Edinburgh 2009.
- [10] Д.К. Боголепов., В.Е. Турлапов. Моделирование каустик в реальном времени на основе комбинированных возможностей OpenCL и шейдеров. *Вестник Нижегородского университета им. Н.И. Лобачевского*, 2011, No 3 (2), с. 180–186
- [11] Hachisuka T., and Jensen H. W. 2010. *Parallel progressive photon mapping on GPUs*. In ACM SIGGRAPH ASIA 2010 Sketches (SA '10). ACM, New York, NY, USA, Article 54, 1 pages.
- [12] Scott Le Grand. “Broad-Phase Collision Detection with CUDA”. In *GPU Gems 3*, Addison-Wesley, 2008, pp. 697–721.
- [13] Aila, T. and Laine, S. 2009. *Understanding the efficiency of ray traversal on GPUs*. In Proceedings of the Conference on High Performance Graphics 2009 (New Orleans, Louisiana, August 01 - 03, 2009). S. N.

## 6. БЛАГОДАРНОСТИ

Работа поддержана грантами РФФИ № 12-01-31027 мол\_а, № 12-01-00560 и стипендией президента РФ № СП-4053.2013.5.

# Реконструкция непрозрачных включений в объектах с высоким показателем преломления

Андрей Лебедев, Виктор Гаганов, Алексей Игнатенко  
факультет Вычислительной Математики и Кибернетики  
Московский Государственный Университет им. М.В. Ломоносова, Москва, Россия  
{alebedev, vgaganov, ignatenko}@graphics.cs.msu.ru

## Аннотация

Задача реконструкции включений в алмазном сырье занимает важное место во всем технологическом процессе добычи и огранки драгоценных камней. Включения – это инородные объекты, которые образуются в сырье естественным путем во время формирования алмаза (Рис. 1). Предложенный в данной статье алгоритм производит полную реконструкцию трехмерных моделей включений по фотоизображениям. Трехмерные модели включений используют воксельное представление. Предложенный алгоритм производит раскраску вокселей, сегментирует воксельную сетку и строит полигональные модели включений. Проведенные эксперименты на реальных и синтетических данных показывают, что предложенный алгоритм способен реконструировать трехмерные модели средних и больших включений, которые могут служить первым приближением для методов уточнения формы включений.

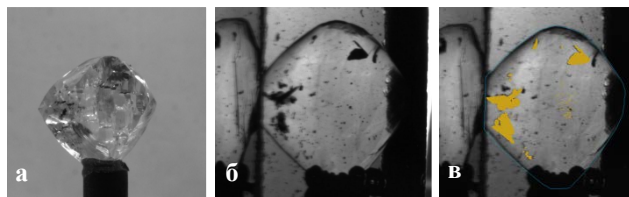
**Ключевые слова:** реконструкция сцены по фотоизображениям, воксельное представление, обработка изображений, сегментация изображений, алмазная промышленность

## 1. ВВЕДЕНИЕ

В области ювелирной обработки драгоценных камней существует множество задач, требующих разработки программного обеспечения для автоматизации сложного технологического процесса [11]. В добываемом алмазном сырье содержится множество различных инородных включений, попавших в алмаз в процессе его роста. При распиле алмаза на отдельные драгоценные камни необходимо отсечь все включения, поскольку наличие включений в бриллиантах отрицательно сказывается на их визуальном качестве, а значит и на их цене. В связи с этим необходимо реконструировать включения внутри сырья и избавляться от них при распиле.

Визуальное обнаружение включений в алмазах сильно затруднено по причине сложной геометрии камней и высокого коэффициента преломления сырья. Для того чтобы сделать включения более заметными, алмаз заливается в иммерсионное стекло с коэффициентом преломления, близким к коэффициенту преломления алмаза. Иммерсионное стекло имеет форму параллелепипеда, что позволяет рассмотреть неискаженные включения через плоскости куба. Иммерсионное стекло подсвечивается специальным освещением так, чтобы все включения выглядели наиболее контрастно. Примеры фотографий исходного сырья и запаянного в иммерсионное стекло приведены на Рис. 1. Включения становятся хорошо заметными, что сильно упрощает их реконструкцию. Однако из-за высокого коэффициента преломления алмаза направления обзора внутри камня разорваны и образуют малые углы, что

усложняет задачу реконструкции. Предложенный в данной статье алгоритм позволяет реконструировать средние и большие включения, локализуя их с высокой точностью. Полученные трехмерные модели включений могут уточняться вручную оператором, либо служить первым приближением для автоматических алгоритмов уточнения.



**Рис. 1:** а) Алмаз до впайки и б) после впайки в иммерсионное стекло; в) разметка включений (отмечена желтым)

Статья организована следующим образом. В разделе 2 приведен обзор существующих методов реконструкции трехмерных моделей по изображениям. Предложенный алгоритм описан в разделе 3. Раздел 4 содержит описание и результаты тестирования предложенного алгоритма. В разделе 5 подводятся итоги и описываются планы на будущее.

## 2. ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ

Среди методов реконструкции трехмерных объектов по фотоизображениям можно выделить следующие: на основе построения видимой оболочки по силуэтам [2, 6, 8, 10], согласования цветов (photo-consistency) [3, 12], резьбы по пространству (space carving) [5], оптимизации трехмерной модели [4], сопоставления и трекинга особых точек [9], оптической томографии [1].

Алгоритм построения видимой оболочки [6] требует наличия выделенных силуэтов искомого объекта на фотоизображении. Каждый из силуэтов на фотографии порождает конус, являющийся геометрическим местом точек, которые при проецировании на данное изображение лежат внутри силуэта. Искомая трехмерная модель является пересечением конусов. Алгоритм прост в реализации, и в случае идеально заданных и согласованных силуэтов он обладает высокой точностью для выпуклых объектов. Тем не менее, на практике выясняется, что алгоритм обладает множеством недостатков, которые были учтены при доработке базового алгоритма. Так, в работе [2] решалась проблема восстановления формы объектов с нетривиальной топологией, когда в трехмерной модели есть сквозные отверстия. При восстановлении объектов, состоящих из нескольких компонент связности, необходимо решать проблему удаления фантомных объектов, которые получаются при пересечении всевозможных конусов между собой. Алгоритм, позволяющий удалять часть фантомных объектов был предложен в работе [8]. В той же работе

рассмотрена и решена проблема ограниченности обзора фотографии, когда силуэт объекта обрезан границей фотографии. Также серьезной проблемой исходного метода видимой оболочки является неустойчивость к несогласованным входным силуэтам. Силуэты могут отфильтровываться как выбросы или приводиться в согласованное состояние [10]. В целом, методы на основе видимой оболочки пригодны для использования в случае наличия уже сопоставленных между собой силуэтов.

Подходы на основе согласования цветов и резьбы по пространству используют воксельное представление объектов. Идея алгоритмов на основе согласования цветов состоит в раскраске воксельного куба путем проецирования каждого вокселя на все изображения и расчета цвета вокселя на основе значений пикселей изображений в области проекции [3]. При этом алгоритм работает как с силуэтами, так и с произвольными изображениями, в том числе цветными. Как и в методе видимой оболочки в данном подходе возникает проблема фантомных объектов. При раскраске вокселей необходимо учитывать согласованность данных с различных фотографий и отфильтровывать выбросы [12]. Алгоритмы на основе резьбы по пространству [5] выделяют группу камер, с которых должны получаться согласованные изображения, затем просматривают слои воксельной сетки от ближнего к дальнему относительно камер и удаляют несогласованные для этих камер воксели. Общим недостатком подходов, использующих воксельную сетку в качестве представления геометрии объектов, является дискретизация пространства. С одной стороны, точность трехмерной модели ограничена размером вокселя, с другой – маленький размер вокселя требует хранения большого объема данных. Можно использовать модифицированное воксельное представление, суть которого заключается в том, что воксельная сетка является неравномерной и она более детализована на краях модели, тогда как вдали от краев изображения размер вокселя больше.

Подходы на основе оптимизации трехмерной модели [4] основаны на формировании оптимизационного функционала, показывающего насколько заданная трехмерная модель согласованна с фотографиями. Функционал может зависеть от множества параметров (в пределе это могут быть положения всех точек трехмерной модели). Далее функционал оптимизируется, на выходе получается модель, которая согласованна с фотографиями. В целом, данный подход является наиболее сложным, но в то же время, наиболее устойчивым к входным данным. Алгоритмы оптимизации трехмерной модели больше нацелены на уточнение уже имеющегося приближения односвязной модели, поэтому для решения задачи реконструкции всех включений данных алгоритмов недостаточно.

В силу сильной разорванности направлений обзора при переходе с одной плоскости иммерсионного стекла на другую, а также произвольности формы включений, методы сопоставления и трекинга особых точек [9] оказываются малоприменимыми. Непрозрачность включений затрудняет применение методов оптической томографии [1], т.к. включения практически полностью поглощают свет.

Предлагаемый нами алгоритм, как и подход на основе согласования цветов, раскрашивает воксельную сетку. Для полной реконструкции включений помимо раскраски вокселей используется сегментация отдельных включений. В настоящее время существует большое число методов сегментации и классификации объектов в воксельном кубе.

Методы сегментации можно разделить на послойные, когда каждый срез воксельной сетки сегментируется независимо от других, и трехмерные, когда используется информация о соседних срезах [7]. В разделе 3 нами предложен алгоритм сегментации, который при обнаружении включений учитывает специфику получаемых воксельных сеток.

### 3. ПРЕДЛАГАЕМЫЙ АЛГОРИТМ

Предлагаемый алгоритм принимает на вход следующую информацию: трехмерную модель алмаза  $M_{rough}$  и иммерсионного стекла  $M_{imm}$ , коэффициенты преломления алмаза  $RI_{rough}$  и иммерсии  $RI_{imm}$ , набор фотографий алмаза, запаянного в иммерсионное стекло  $I_k$ ,  $k = \overline{1, n}$  (пример одного из изображений приведен на Рис. 1б), операторы проекции  $Proj_k$  точек трехмерного пространства на каждую фотографию (с учетом преломления в средах). Фотографии снимаются стационарной камерой при вращении камня вокруг вертикальной оси. Общее число фотографий  $n$  колеблется от 100 до 400. На выходе алгоритм выдает реконструированные трехмерные модели включений.

Сначала по заданным фотографиям  $I_k$  строятся двумерные вероятностные карты  $Prob_k$  включений, такие что для каждого пикселя  $p \in I_k$   $Prob_k(p) = P(p \in Proj_k(M_{incl}))$ , где  $M_{incl}$  – трехмерные модели искомым включений,  $P(\cdot)$  – вероятность. Для построения таких карт можно использовать различные методы, например, инвертировать исходные фотографии (поскольку включения темные на светлом фоне).

Алгоритм реконструкции включений по вероятностным картам состоит из двух этапов: раскраска вокселей и сегментация воксельной сетки.

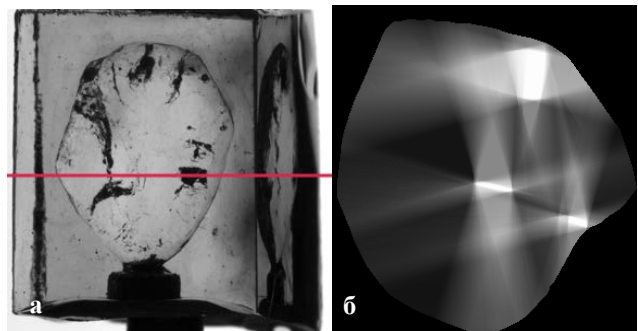


Рис. 2: а) Линия среза воксельной сетки; б) пример среза раскрашенной воксельной сетки.

#### 3.1. Раскраска вокселей

Весь объем модели камня  $M_{rough}$  разбивается воксельной сеткой  $V$ . Спроецируем каждый воксель  $V_i \in V$  на все карты вероятности  $Prob_k$ ,  $k = \overline{1, n}$ . Пусть  $S_i^k = Proj_k(V_i)$  – проекция вокселя на карту достоверности, а  $p_i^k = \max_{p \in S_i^k} (Prob_k(p))$  – максимальное значение вероятности в области проекции вокселя. Тогда цвет, в который раскрашивается воксель, вычисляется как  $Col(V_i) = \sum_{k=1}^n p_i^k / n$ . Пример одного из раскрашенных воксельных срезов  $Sec_j$  (обычно делается всего около  $m=400$  срезов) приведен на Рис. 2б. На Рис. 2а показана линия, по которой производился срез воксельной сетки.

Стоит отметить, что простое усреднение  $p_i^k$  позволяет более стабильно оценивать вероятность нахождения включения в вокселе, по сравнению с идеей пересечения конусов в

алгоритме видимой оболочки, которая эквивалентна тому, что цвет вокселя рассчитывается как  $Col(V_i) = \min_k p_i^k$ .

### 3.2. Сегментация воксельной сетки

Перед описанием алгоритма сегментации отдельных включений, проанализируем раскрашенные срезы воксельной сетки  $Sec_j$ . Очевидно, что чем выше значение цвета в вокселе, тем более вероятно, что этот воксель принадлежит включению. Также можно отметить, что искомые включения расположены в точках локального максимума воксельного среза. При этом окружающий включения фон может иметь различную яркость. В силу разрывности направлений обзора между перпендикулярными гранями иммерсионного стекла все включения оставляют за собой «шлейфы» в направлениях обзора (см. Рис. 26). Предложенный алгоритм сегментации позволяет избавиться от «шлейфов», а также позволяет детектировать области включений.

Поиск небольших и более крупных включений производится независимо двумя разными алгоритмами. Для поиска небольших включений (как правило, это включения линейным размером не больше 5 вокселей) применяется алгоритм адаптивного порога. Пусть  $Sec_j^{med}$  – результат применения медианного фильтра к воксельному срезу  $Sec_j$  с радиусом  $R_{med}$ . Множество найденных мелких включений в воксельном срезе будет определяться как:

$$Small_j = \{(x, y): Sec_j(x, y) - Sec_j^{med}(x, y) > T_{adapt}\}.$$

Все мелкие включения получают объединением:  $Small = \cup_{j=1}^m Small_j$ . Значения радиусов медианного фильтра  $R_{med}$  и порога  $T_{adapt}$  подбираются экспериментальным путем. В разделе 4 будут указаны все значения используемых констант (значения констант подбираются один раз для алгоритма).

Алгоритм поиска более крупных включений состоит из следующих частей:

- Обнаружение семян для последующей сегментации
- Отброс ложных семян
- Непосредственно сегментация воксельной сетки
- Постобработка включений

Для обнаружения семян используется метод, известный как алгоритм расширения максимумов (extended-maxima transform) [13]. На первом этапе из воксельного среза  $Sec_j$  вычитается наперед подобранная константа  $H$ . К полученной разности применяется алгоритм морфологической дилатации для полутоновых изображений, который сглаживает яркостные локальные максимумы в изображении:  $Sec_j^{morph} = Morph(Sec_j - H)$ . Наконец, множество искоемых семян определяется как:

$$Seed = \{(x, y): Sec_j^{morph}(x, y) - Sec_j(x, y) > 0\}.$$

После обнаружения семян необходимо отсеять среди них ложные. На первом этапе выделяются связанные компоненты семян  $Seed_i$ . Для отсеивания отдельной компоненты  $Seed_i$  используется два признака:

- Средняя яркость компоненты должна быть больше порога:  $\sum_{p \in Seed_i} Sec_j(p) / card(Seed_i) > T_{br}$
- Средняя яркость изображения с примененной морфологией должна быть также больше порога:  $\sum_{p \in Seed_i} Sec_j^{morph}(p) / card(Seed_i) > T_h$

Для сегментации куба применяется алгоритм разрастания областей. Разрастание начинается из отфильтрованных на предыдущем этапе связанных компонент (в качестве семени

служит точка  $p_i \in Seed_i^{flt}$ , такая что  $Sec_j(p_i) = \max_{p \in Seed_i^{flt}} Sec_j(p)$ . Алгоритм разрастания областей

останавливается, когда больше нет ни одной точки  $p \in Sec_j$ , граничащей с текущей областью сегментации  $S_k$  ( $S_0 = \{p_i \in Seed_i^{flt}\}$ ), для которой разность между  $Sec_j(p)$  и максимально яркой точкой области  $S_k$  меньше порога  $T_{rg}$ . В итоге, мы получили послойно отсегментированную воксельную сетку.

Последним этапом является постобработка результата с помощью трехмерной математической морфологии, которая применяется ко всей воксельной сетке. В качестве структурного элемента для последовательных операций расширения и закрытия используется куб.

## 4. ЭКСПЕРИМЕНТЫ

Предложенный алгоритм был протестирован на синтетических данных. Синтетические вероятностные карты  $Prob_k$  были получены следующим образом. Сначала по реальным фотографиям  $I_k$  были получены ручные разметки включений  $M_{incl}$  в трехмерном пространстве (см. Рис. 3а). Для этого каждое из включений  $Incl \in M_{incl}$  вручную размечалось на разных фотографиях. По полученным разметкам реконструировались трехмерные модели методом видимой оболочки. Положим

$$Prob_k(x, y) = \begin{cases} 1, & \text{если } (x, y) \in Proj_k(M_{incl}) \\ 0, & \text{в противном случае} \end{cases} \quad (\text{см. Рис. 3б}).$$

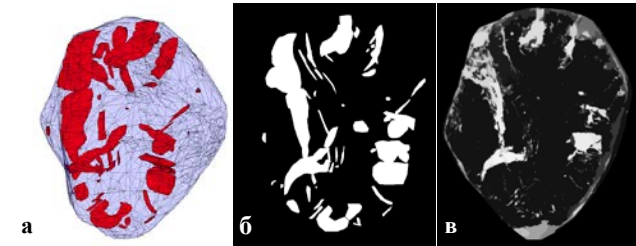
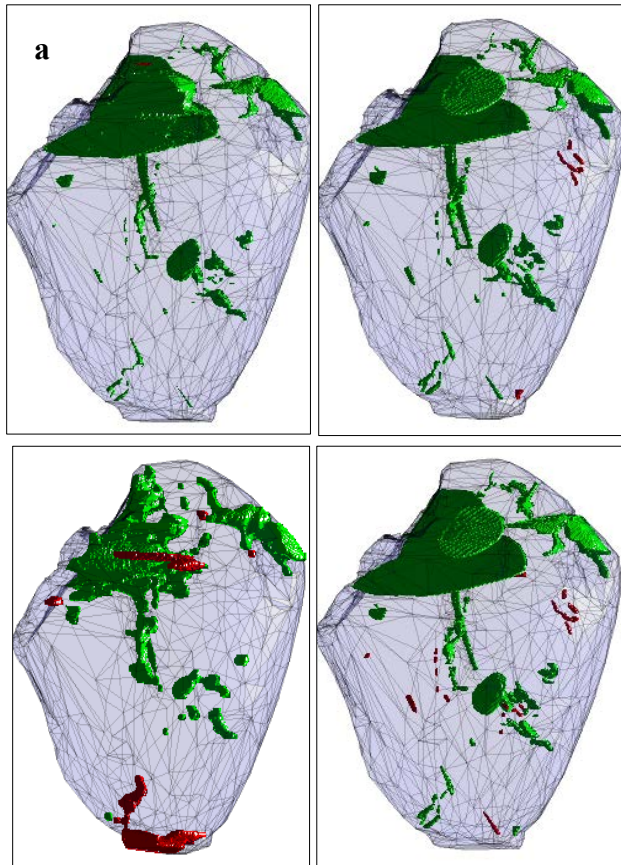


Рис. 3: а) пример ручной разметки трехмерных включений; б) синтетические карты вероятности; в) реальные карты достоверности.

Для проведения экспериментов было вручную размечено 18 алмазов и сгенерирован набор синтетических вероятностных карт  $Prob_k$  (всего 100 изображений на каждый алмаз). По сгенерированным силуэтам с помощью предложенного алгоритма были построены включения  $Rec_{incl}$ . При этом использовались следующие значения параметров алгоритма (значения параметров выбирались один раз для алгоритма):  $R_{med} = 9$  (в вокселях),  $T_{adapt} = 0.1$ ,  $H = 0.118$ ,  $T_{br} = 0.7$ ,  $T_h = 0.059$ ,  $T_{rg} = 0.39$ .

Реконструированные трехмерные модели включений  $Rec_{incl}$  сравнивались с исходными ручными разметками  $M_{incl}$  (далее будем называть их эталонными). Мы оценивали число верно локализованных дефектов и число ложных дефектов. Включение из эталонной разметки  $GT \in M_{incl}$  считается пропущенным, если  $Vol(Dil(GT, R_{dil}) \cap Rec_{incl}) < 0.5 \cdot Vol(GT)$ , где  $Dil(\cdot, \cdot)$  – оператор морфологического расширения,  $R_{dil}$  – радиус расширения (в нашем случае он составлял 3 линейных размера вокселя),  $Vol(\cdot)$  – объем. В противном случае эталонное включение  $GT$  считается покрытым построенным. Построенное алгоритмом включение  $Det \in Rec_{incl}$  считается ложным, если  $Vol(Dil(Det, R_{dil}) \cap M_{incl}) < 0.5 \cdot Vol(Det)$ . В противном

случае считается, что построенное включение покрыто эталонным. На Рис. 4 представлен пример сравнения реконструированных трехмерных моделей (Рис. 4а) с эталонной разметкой (Рис. 4в). На изображении слева зеленым отмечены включения, покрытые эталонными, ложные отмечены красным. На изображении справа зеленым отмечены эталонные включения, покрытые построенными, пропущенные отмечены красным.



**Рис. 4:** Пример сравнения построенных трехмерных моделей (**а**) на основе синтетических данных, **б**) на основе реальных данных) с эталонной разметкой (**в**) и **г**) соответственно).

Средняя статистика по 18 наборам показывает, что процент найденных (непропущенных) составляет 82%, процент ложно найденных включений при этом составляет 4,5%. Стоит отметить, что большая часть пропущенных эталонных включений являются мелкими и не могут быть обнаружены из-за конечности линейного размера вокселя. Ложно найденные дефекты возникают в случае пересечения «шлейфов» от двух различных дефектов в воксельном кубе.

Кроме того алгоритм был протестирован на реальных данных, для которых карты вероятности  $Prob_k$  были получены автоматически с помощью типовых алгоритмов с использованием машинного обучения. Пример такой карты приведен на Рис. 3в. Статистика работы алгоритма показывает, что процент найденных включений составляет 62% при проценте ложных 40%. На Рис. 4 представлен пример сравнения реконструированных по реальным данным трехмерных моделей (Рис. 4б) с эталонной разметкой (Рис. 4г). В целом, эксперименты показывают, что предложенный алгоритм может применяться для решения задачи поиска включений крупного и среднего размера.

## 5. ЗАКЛЮЧЕНИЕ

В данной статье предложен алгоритм, реконструирующий включения в алмазном сырье. Результаты экспериментов показывают, что с помощью данного алгоритма можно реконструировать большие и средние включения. Точность построенных включений, ограниченная снизу линейным размером вокселя, не позволяет напрямую использовать результаты для эффективного распила камней. Тем не менее, данный алгоритм позволяет получать первое приближение моделей включений, необходимое для алгоритмов уточнения включений (например, с помощью подходов оптимизации трехмерной модели [4]), что позволит создать полностью автоматический алгоритм в будущем.

## 6. ЛИТЕРАТУРА

- [1] Arridge S. R. Optical tomography in medical imaging. Inverse problems, 1999, Vol. 15, No. 2, R41.
- [2] Boyer E., Franco J. S. A hybrid approach for computing visual hulls of complex objects. CVPR, 2003, pp. 695-701.
- [3] Eisert P., Steinbach E., Girod B. Multi-hypothesis, volumetric reconstruction of 3-D objects from multiple calibrated camera views. Acoustics, Speech, and Signal Processing, 1999, Vol. 6, pp. 3509-3512.
- [4] Gargallo P., Prados E., Sturm P. Minimizing the reprojection error in surface reconstruction from images. ICCV, 2007, pp. 1-8.
- [5] Kutulakos K. N., Seitz S. M. A theory of shape by space carving. International Journal of Computer Vision, 2000, Vol. 38, No. 3, pp. 199-218.
- [6] Laurentini A. The visual hull concept for silhouette-based image understanding. Pattern Analysis and Machine Intelligence, 1994, Vol. 16, No. 2, pp. 150-162.
- [7] Lucchi A., Smith K., Achanta R., Knott G., Fua P. Supervoxel-based segmentation of mitochondria in EM image stacks with learned shape features. Medical Imaging, 2012, Vol. 31, No. 2, pp. 474-486.
- [8] Michoud B., Bouakaz S., Guillou E., Briceño H. Largest silhouette-equivalent volume for 3d shapes modeling without ghost object. M2SFA2, 2008.
- [9] Sinha S. N., Frahm J. M., Pollefeys M., Genc Y. Feature tracking and matching in video using programmable graphics hardware. Machine Vision and Applications, 2011, Vol. 22, No. 1, pp. 207-217.
- [10] Sinha S. N., Pollefeys M. Camera network calibration and synchronization from silhouettes in archived video. ICCV, Vol. 87, No. 3, pp. 266-283.
- [11] Sivovolenko S., Mistry J., Holloway G., Shelementiev Y. Buying Diamonds: Consumers' Problems and a Solution [HTML]([http://www.pricerscope.com/journal/buying\\_diamonds\\_consumers\\_problems\\_and\\_solution](http://www.pricerscope.com/journal/buying_diamonds_consumers_problems_and_solution)).
- [12] Slabaugh G. G., Culbertson W. B., Malzbender T., Stevens M. R., Schafer R. W. Methods for volumetric reconstruction of visual scenes. ICCV, 2004, Vol. 57, No. 3, pp. 179-199.
- [13] Vincent L. Morphological grayscale reconstruction in image analysis: applications and efficient algorithms. Image Processing, 1993, Vol. 2, No. 2, pp. 176-201.

# Калибровка источника света в сцене с прозрачным объектом по фотографиям

В.В. Афанасьев, А.В. Игнатенко, И.О. Тисевич

*ИПМ РАН им. М.В. Келдыша,*

*Лаборатория Компьютерной Графики и Мультимедиа*

*Факультета ВМК МГУ им. М.В. Ломоносова*

[vafanasjev@graphics.cs.msu.ru](mailto:vafanasjev@graphics.cs.msu.ru) [ignatenko@graphics.cs.msu.ru](mailto:ignatenko@graphics.cs.msu.ru) [itisevich@graphics.cs.msu.ru](mailto:itisevich@graphics.cs.msu.ru)

## Аннотация

В работе предлагается решение задачи калибровки освещения для сцены с прозрачным объектом, который освещается рассеянным светом диффузора со светодиодной подсветкой. Вычислено распределение яркости по диффузору, имеется возможность подобрать виртуальную экспозицию для заданной фотографии. Калибровка позволяет добиться большего фотореализма при визуализации сцены.

## 1. Введение

Задача синтеза фотореалистичных изображений требует точного задания всех параметров сцены для достижения лучшего результата. Важнейшую роль играет освещение (здесь и далее под освещением понимается совокупность источников света в сцене). От того, насколько соответствует освещение виртуальной сцены освещению реальной сцены, напрямую зависят правильность цветов и яркости синтезированного изображения. Освещение сцены, как правило, сложно напрямую измерить, и оно может меняться в зависимости от условий съёмки фотографий.

Реконструкция полного освещения сцены по фотографиям в общем виде очень сложна, в данной работе эта задача решается в частном случае для специального вида сцены и освещения, встречающихся, например, при сканировании драгоценных камней в ювелирной промышленности.

## 2. Обзор

Задача восстановления освещения решается при различных допущениях, позволяющих вычислить освещение в рамках принятой модели.

В работе [1] для построения изображений широкого динамического диапазона восстанавливается яркость видимой части сцены по набору изображений с разной экспозицией. Учитываются преобразования, которые претерпевает исходная яркость сцены в оптической системе камеры. Источники света, находящиеся вне кадра, не строятся.

Авторы [2] предлагают метод восстановления видимого и невидимого освещения снятого кадра для одного источника света. Предполагается, что поверхности сцены диффузные. Незвестное положение источника света авторы предлагают оценить с помощью теней от объектов. Реконструированное освещение применяется для реалистичного внедрения виртуального объекта в фотографию.

Способ представления освещения сцены в виде бесконечно удалённой сферической панорамы предложен в [3]. Освещение восстанавливается вместе с формой объекта интереса. Во многих исследованиях принята такая модель освещения сцены, она позволяет существенно снизить

сложность задачи и не даёт погрешностей, если настоящие источники света не расположены близко к освещаемому объекту.

В работе [4] освещение сцены представлено вейвлет-разложением кубической карты, что является комбинацией подхода с бесконечно удалённой панорамой и реконструкцией набора точечных источников света. Поддерживаются модели освещения Ламберта и Фонга, что даёт возможность использовать диффузные и бликующие поверхности.

В рассмотренных исследованиях восстановление освещения сцены основано на отражении света от объектов и восстановлении только отражательных свойств. В данной работе объект интереса прозрачен и находится сравнительно близко к источнику освещения.

## 3. Задача

Сцена состоит из:

- Первичных источников света (набора светодиодов)
- Матового рассеивателя (диффузора)
- Прозрачного объекта, освещённого рассеянным светом диффузора
- Камеры, снимающей прозрачный объект и, возможно, часть диффузора

Известные параметры (входные данные):

- Форма диффузора
- Положение и направление светодиодов подсветки
- Угол половинной яркости светодиодов
- Модель прозрачного объекта, материал (известны показатель преломления  $n$ , возможно, показатель поглощения  $k$ )
- Фотографии фона без прозрачного объекта
- Экспозиция фотографий фона в градациях серого
- Фотографии прозрачного объекта в градациях серого

Неизвестные параметры (выходные данные):

- Виртуальная экспозиция (или коэффициент яркости)  $L$
- Полная диаграмма направленности светодиодов  $D(\vec{r}_i)$
- ДФО краски, которой покрыт диффузор  $F(\vec{r}, \vec{r}_i)$
- Как следствие, неизвестно освещение прозрачного объекта диффузором

Цель – построить модель освещения прозрачного объекта в заданной сцене по его фотографиям таким образом, чтобы оно как можно точнее соответствовало настоящему освещению для прозрачного объекта. То есть нужно найти такие  $F(\vec{r}, \vec{r}_i)$ ,  $D(\vec{r}_i)$ ,  $L$ ,  $k$ , чтобы минимизировать среднее по кадру отклонение результата визуализации от фотографии.

#### 4. Распространение света в сцене

Сцена построена так, чтобы в камеру попадал в основном вторичный свет, отражённый диффузором, и светодиоды не освещали объект напрямую.

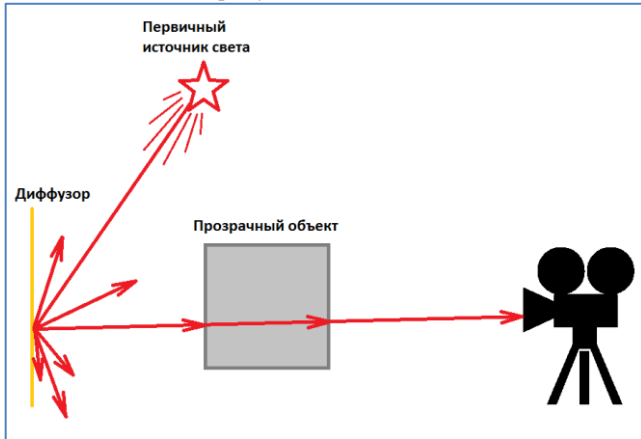


Рис. 1. Схема сцены

Кадр, снятый с камеры сканера, захватывает только часть диффузора. Оставшаяся часть также участвует в освещении, но её вклад напрямую не наблюдается и выражен только в яркости некоторых областей объекта.

Фотография фона без объекта, на которой фон не пересвечен, даёт точное распределение освещённости по видимой части диффузора.

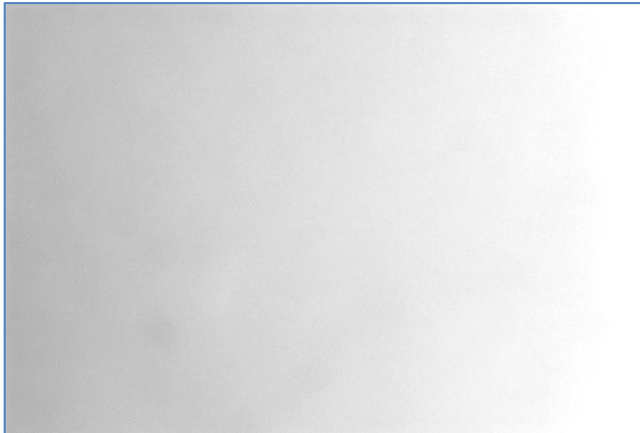


Рис. 2. Пример фотографии фона

Простейший подход – экстраполировать освещённость на невидимую часть и при визуализации использовать карту освещённости модели диффузора, либо сферическую панораму. Однако это не всегда можно сделать из-за следующих факторов:

- Будет получена освещённость, видимая из камеры, т.е. излучение в направлении камеры. Но направления падения света с диффузора на объект могут существенно отличаться от направления на камеру, т.к. объект и диффузор имеют сопоставимые размеры, находятся на небольшом расстоянии друг от друга.
- Экстраполяция возможна только при малых размерах невидимой части диффузора, иначе возникнет значительная погрешность при удалении от границ видимой части.

Поэтому выбран другой подход. Освещённость диффузора во всех точках предлагается рассматривать не как одну неизвестную величину, а как комбинацию нескольких известных и неизвестных величин. Далее, используя фотографии, подобрать неизвестные величины, опираясь на известные.

Так, яркость фона без объекта может быть представлена в виде  $I_0(\vec{r}) = L * D(\vec{r}_l) * F(\vec{r}, \vec{r}_l)$ , где:

$\vec{r}$  – направление излучения диффузора

$L$  – коэффициент яркости светодиода

$D$  – диаграмма направленности светодиода

$F$  – ДФО поверхности диффузора

$\vec{r}_l$  – направление из точки диффузора на светодиод

Соответственно, яркость точки изображения  $I = I_0 * A$ , где  $A$  – коэффициент поглощения света в прозрачном материале для данной траектории, задаётся законом Бугера-Ламберта-Бера в зависимости от показателя поглощения и длины пути света в среде. Отклик камеры принят за линейный и не рассматривается, хотя может быть добавлен в модель.

Калибровка яркости освещения для заданной сцены проводится в 2 этапа:

1. Определение неизвестных значений ДФО диффузора и диаграммы направленности светодиодов по фотографии фона.
2. Определение коэффициента яркости и уточнение показателя поглощения по фотографиям прозрачного объекта.

#### 5. Вычисление диаграммы направленности светодиодов

Зону диффузора, видимую в кадре, освещают лучи света, пришедшие из светодиодов с углами из достаточно ограниченного диапазона. Теоретически существует возможность точно вычислить диаграмму направленности только в этом диапазоне углов. Чему она равна во всех остальных точках (кроме угла половинной яркости), неизвестно. Поэтому выбран достаточно простой вид диаграммы направленности. Она представлена кусочно-линейной функцией, значения которой в наборе ключевых точек определяются параметром  $t$ .

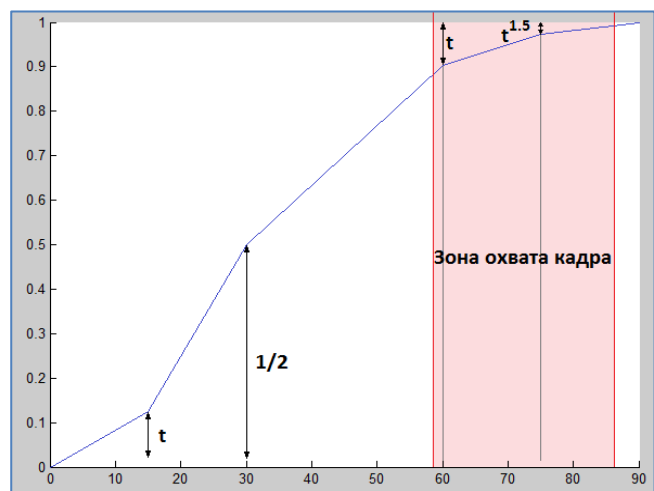


Рис. 3. Параметрическая функция для диаграммы направленности светодиода



Фотографии фона позволяют варьировать диаграмму направленности, подбирая распределение яркости по кадру и минимизируя абсолютную разность между фотографией фона и результатом визуализации фона. При этом ДФО диффузора либо явно задаётся (например, измеряется), либо параметризуется и подбирается вместе с диаграммой направленности.

Таким образом, полученные диаграмма направленности светодиодов подсветки и ДФО диффузора задают параметры распространения света при неизменной геометрии и материалах сцены. Если известны эти две функции, то синтезированное изображение фона можно сравнивать с фотографией фона с точностью до коэффициента яркости  $L$ , который зависит от яркости светодиодов в момент съёмки и времени экспозиции. Его нужно подбирать при каждом изменении яркости подсветки или времени экспозиции.

## 6. Вычисление коэффициента яркости

Коэффициент яркости  $L$  нужно определить для каждого снятого набора кадров по фотографиям, где есть не только фон, но и прозрачный объект. Фона может вообще не быть в кадре. Фон может быть пересвечен, или иметь нормальную яркость.

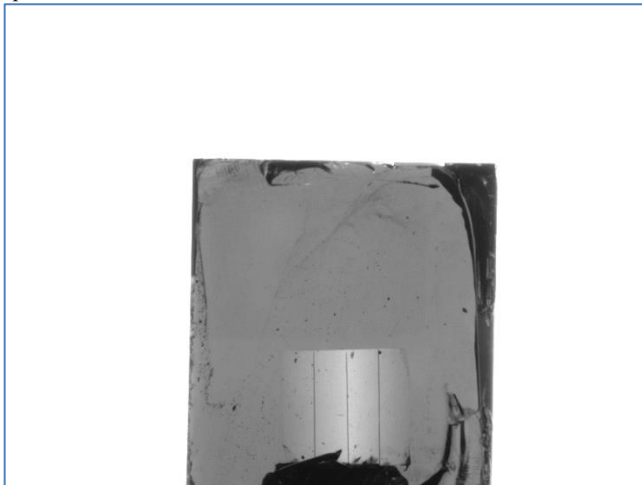


Рис. 4. Пример фотографии куба из стекла с пересвеченным фоном

Коэффициент яркости вычисляется путём минимизации средней абсолютной разности между фотографией и результатом визуализации по параметрам  $L$  (коэффициент яркости) и  $k$  (показатель поглощения). Исследование этой функции при вычислении яркости по одному кадру показало некоторые особенности. Для примеров с кубом, повернутым гранью на камеру, функция невязки выглядит следующим образом:

- В случае наличия в кадре фона нормальной яркости функция невязки имеет чётко выраженный минимум.

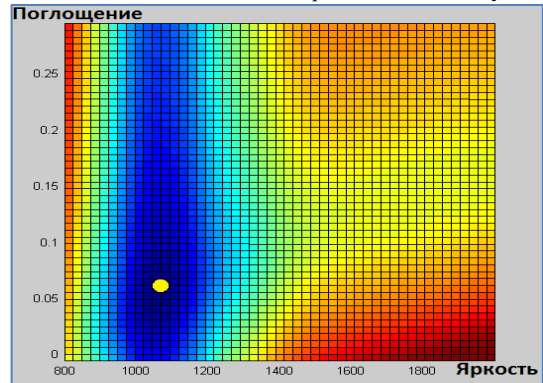


Рис. 5. Функция невязки для фона нормальной яркости

- В случае пересвеченного фона минимальное значение функции распределено вдоль криволинейного «луча», т.е. имеется нижний предел яркости, определяемый минимально необходимым значением яркости для пересвечивания фона. Для большей яркости находится такое значение показателя поглощения, которое минимизирует функцию при любой яркости.

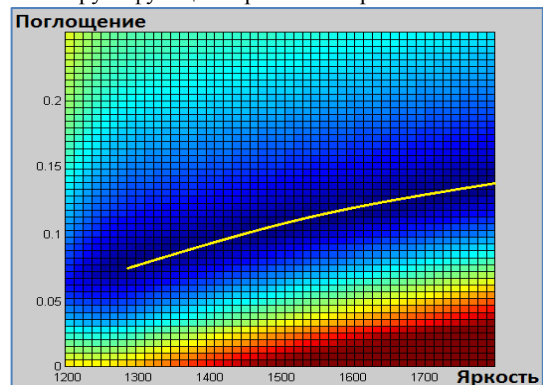


Рис. 6. Функция невязки для пересвеченного фона

- При отсутствии фона в кадре наблюдаем такой же эффект, но теперь оптимальные значения параметров находятся вдоль некоторой линии без ограничений с обеих сторон.

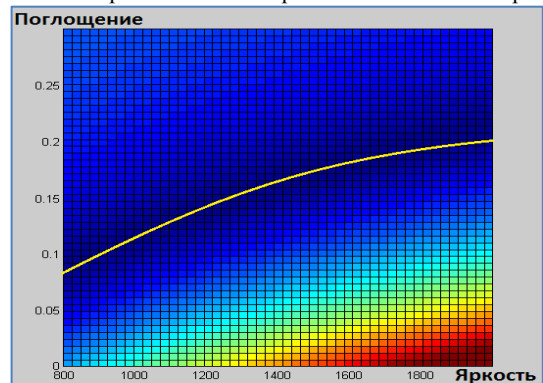


Рис. 7. Функция невязки при отсутствии фона в кадре

В двух последних случаях неоднозначность возникает из-за того, что фон полностью или частично перестаёт давать информацию о яркости. Компенсировать это должен прозрачный объект, но, хотя он и не пересвечен, в

рассмотренных примерах (куб, повернутый гранью в сторону камеры) поглощение света оказалось распределено по объекту равномерно. Это позволяет всем пикселям объекта в визуализации пропорционально изменять яркость при изменении коэффициента яркости освещения. Поэтому все пиксели объекта могут одновременно приблизиться по яркости к пикселям на фотографии. Исправить это можно несколькими способами. Принципиально нужно добиться неравномерного распределения поглощения по объекту.

- Взять для вычисления яркости такой кадр, где объект имеет разную толщину в разных точках. Например, куб может быть повернут под углом. Для объектов более сложной формы подходят почти все ракурсы.
- Использовать не один кадр, а несколько для разных положений объекта, среди которых также должны найтись кадры с разной видимой толщиной.
- Если по каким-то причинам выполнить такую съёмку нельзя, нужно зафиксировать значение коэффициента поглощения, измерив его другим способом. Тогда яркость будет вычислена однозначно.

Функция невязки может иметь несколько локальных минимумов, например, из-за загрязнения внутри прозрачного объекта. Для нахождения истинных значений яркости и показателя поглощения необходимы методы глобальной оптимизации.

## 7. Результаты

Разработанный метод был применён при калибровке освещения в сканере для съёмки драгоценных камней.

Сканер имеет светодиодную подсветку и матовый отражающий диффузор с неизвестной ДФО.

ДФО была задана простой функцией с 1 параметром, задающим силу рассеивания света. Далее проведена совместная оптимизация по параметру ДФО и параметру диаграммы направленности, в результате чего максимальное отклонение яркости фона на визуализированном кадре от фона на фотографии составило менее 1%.

Драгоценный камень, залитый в иммерсионное стекло, имеет низкий показатель поглощения, по сравнению со стеклом, что обеспечивает разную видимую толщину стекла в пределах одного кадра, и функция невязки имеет чёткий минимум даже при отсутствии фона в кадре. Это позволяет однозначно вычислить значения показателя поглощения стекла и коэффициента яркости.

Полученное освещение было использовано для фотореалистичной визуализации. Синтезированные изображения отличаются по яркости от фотографии в среднем по кадру на значения от 1% до 5%, в зависимости от снимаемого объекта.

## 8. Ссылки

- [1] Paul E. Debevec, Jitendra Malik, Recovering high dynamic range radiance maps from photographs. Proc. 24th annual conference on Computer graphics and interactive techniques, p.369-378, August 1997.
- [2] T. Grosch, S. Mueller, and W. Kresse. Goniometric light reconstruction for augmented reality image synthesis. Graphiktag im Rahmen der GI Jahrestagung, Frankfurt am Main, 2003.

- [3] D. Hough and N. Ahuja. Estimation of the Light Source Distribution and Its Use in Integrated Shape Recovery from Stereo and Shading, Proc. IEEE Fourth Int'l Conf. Computer Vision, pp. 148-155, 1993.
- [4] C. Upright, D. Cobzas, M. Jagersand. Wavelet-based Light Reconstruction from a Single Image. Computer and Robot Vision, pp. 305 – 312, 2007.

# Алгоритм автоматического обнаружения изображений номерных знаков железнодорожного подвижного состава на основе метода адаптивного бустинга

И.В. Бекетова, С.Л. Каратеев, Ю.В. Визильтер

Федеральное государственное унитарное предприятие «Государственный научно-исследовательский институт авиационных систем», г. Москва

irus@gosniias.ru

## Аннотация

Предложен алгоритм обнаружения номерных знаков железнодорожного подвижного состава, решающий задачи селекции изображения по пространственным частотам с целью фильтрации сопутствующей помеховой цифробуквенной информации, и классификации изображения в пространстве признаков, формируемого с помощью модифицированных фильтров Хаара. Классификатор построен на основе разработанного авторами оригинального алгоритма адаптивного бустинга AdaBoost. При моделировании на тестовой выборке изображений железнодорожного подвижного состава алгоритм показал устойчивость и эффективность работы.

**Ключевые слова:** распознавание символьной информации, машинное обучение, адаптивный бустинг, каскадный классификатор, вейвлет-преобразование Хаара.

## 1. ВВЕДЕНИЕ

Обнаружение и локализация изображений номерных знаков железнодорожного подвижного состава представляется сложной задачей вследствие неопределенности условий наблюдения и многовариантности описания объекта обнаружения. Неопределенность условий наблюдения обусловлена необходимостью функционирования системы в разное время суток, при различных погодных условиях (дождь, снег, туман) и разных условиях освещенности (затененность, солнечная засветка, световые блики). Неопределенность объекта обнаружения складывается из существования множества различных типов номерных знаков железнодорожного подвижного состава, отличающихся стилем начертания, количеством символов, негативным или позитивным представлением символов относительно фона, динамически меняющимся фоном, к тому же зачастую покрытых грязью, пылью, потеками. Точные размеры номерного знака и его положение на изображении не определены, поскольку неизвестно расположение железнодорожного состава по глубине кадра и неизвестно размещение номерного знака на поверхности железнодорожного вагона, контейнера или цистерны.

Существует два основных подхода к анализу цифровых изображений и обнаружению на них объектов заданных классов: восходящий и нисходящий. При восходящем подходе [2, 5, 7] анализ изображения начинается на пиксельном уровне, где изображение сегментируется на зоны, соседние области. Далее эти области классифицируются как принадлежащие к одному из заданных классов. Данный

подход работает хорошо, когда объекты, представляющие интерес, имеют относительно однородные атрибуты, например, интенсивность, цвет, или текстура. Вместе с тем, существует методы прямого обнаружения объектов на изображении, которые предполагают анализ изображения в целом, с последующим разделением на области конкретного типа. Методы данной категории базируются на инструментарию машинного обучения [1, 3, 4].

Среди методов распознавания изображений на основе машинного обучения наиболее популярным и широко применяемым в различных приложениях (обнаружение лиц, автомобилей, пешеходов и т.д.) методом машинного обучения является метод на основе бустинга. Особенностью алгоритма бустинга AdaBoost, разработанного Виолой и Джонсом [6] для обнаружения лиц, является механизм выделения характерных признаков, базирующийся на использовании модифицированных фильтров Хаара. Выбор признакового пространства этого типа, для обнаружения лиц кажется вполне естественным - основные черты лица (глаза, нос, рот) имеют аналогичную форму и пространственное положение для большинства изображений лиц. В то же время существует устойчивое мнение, что нерационально использовать модифицированные фильтры Хаара в качестве признаков для обнаружения цифробуквенной информации [4]. Изображения цифробуквенной информации имеют большую вариативность по сравнению с изображениями лиц и отличаются значительным разнообразием структуры и начертаниями символов. В данной работе показано, что формирование информативных признаков на основе преобразований Хаара позволяет построить простой, быстрый и эффективный алгоритм обнаружения номерных знаков на изображениях железнодорожных вагонов.

## 2. ОПИСАНИЕ АЛГОРИТМА

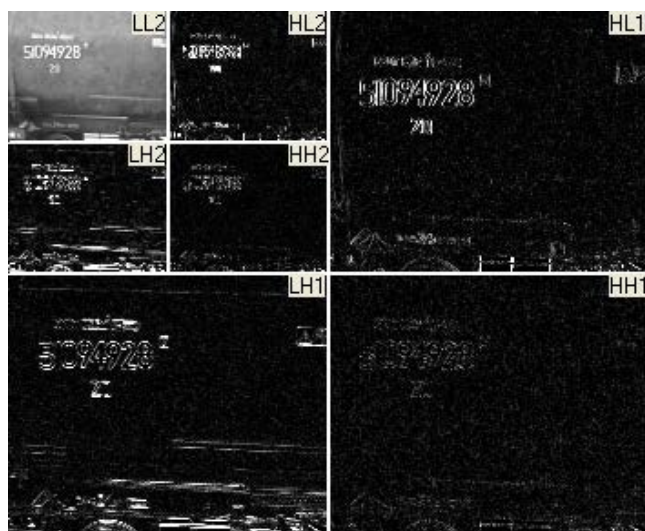
Для решения задачи обнаружения номерных знаков железнодорожного подвижного состава разработан оригинальный алгоритм, состоящий из следующих этапов:

селекция изображения по пространственным частотам;

классификация изображения в пространстве признаков, формируемого с помощью модифицированных фильтров Хаара.

Изображения номерных знаков железнодорожных вагонов, контейнеров, цистерн отличаются от сопутствующей цифробуквенной информации, нанесенной на их поверхности, своими геометрическими характеристиками. Для информационной заметности номерных знаков их делают, как правило, существенно крупнее. Крупные размеры

изображений номерных знаков позволяют их селектировать по пространственным частотам. На первом этапе алгоритм выполняет селекцию изображения по пространственным частотам. В качестве математического инструмента используется дискретное двумерное вейвлет-преобразование Хаара. Матрица коэффициентов вейвлет-преобразования отражает иерархию уровней детализации изображения и состоит из низкочастотной аппроксимации LL и высокочастотных (HL, LH, HH) квадрантов, содержащих уточняющие высокочастотные компоненты. На рис.1. представлена визуализация матрицы компонент 2-х кратного дискретного двумерного вейвлет-преобразования Хаара изображения поверхности вагона.



**Рис.1.** Визуализация матрицы компонент 2-х кратного дискретного двумерного вейвлет-преобразования Хаара изображения поверхности вагона.

Кратномасштабная декомпозиция с одновременной низкочастотной фильтрацией позволяет эффективно отфильтровать избыточную помеховую информацию, сохраняя информацию о номерном знаке.

На втором этапе решается целевая задача обнаружения изображения номерного знака, где локальная область поиска классифицируется в пространстве информативных признаков изображения на два класса: объект (изображение номерного знака) или не объект. Изображения цифробуквенной последовательности можно отнести к объектам, пространственная структура которых не так ярко выражена, как, например, у изображений человеческих лиц, но имеет четкую тенденцию. Типы шрифтов и способы начертания символов могут варьироваться в широких пределах, вместе с тем последовательность символов имеет определенную регулярность. Учет характерных особенностей, присущих цифробуквенной последовательности, при игнорировании несущественной высокочастотной информации о мелких деталях, обеспечивается в пространстве интегральных признаков, вычисляемых с помощью фильтров типа Хаара. В работе использовались фильтры, состоящие из двух областей (признаки краев), и фильтры, состоящие из трех областей (признаки линий), как показано на рис.2. Значения признаков определяются перепадами яркостей областей внутри

прямоугольника, который может занимать различные положения и масштабы на изображении.



**Рис. 2.** Фильтры Хаара для выделения характерных признаков номерных знаков.

Классификатор построен на основе разработанного авторами оригинального алгоритма адаптивного бустинга [8]. Алгоритм обучения классификатора является оригинальным вариантом алгоритма машинного обучения AdaBoost [6]. Разработанный алгоритм обладает эффективной процедурой обучения классификатора, основными особенностями которой являются:

использование LUT-классификаторов, что позволяет использовать информацию о законах распределений значений компонент вектора признаков;

быстрое обучение за один проход по обучающей выборке.

Обучение проводится в режиме онлайн, когда обучающая выборка формируется непосредственно в процессе обучения. На каждой итерации обучения меняются характеристики распределения обучающих примеров и применяются специальные процедуры для исключения корреляции между откликами слабых классификаторов, например исключение использованных классификаторов для построения последующих каскадов. Модифицированная схема обучения ансамбля классификаторов демонстрирует быструю сходимость алгоритма и значительно сокращает потребность для обеспечения требуемой точности количество слабых классификаторов. Ансамбли LUT-классификаторов объединены в каскадную структуру. В ходе обучения каскада для каждого ансамбля классификаторов используется ограничение на ошибку пропуска объекта, которая не должна быть выше 0.01.

### 3. РЕЗУЛЬТАТЫ ОБУЧЕНИЯ И ТЕСТИРОВАНИЯ

Для обучения классификатора использовалась база данных изображений номерных знаков железнодорожных вагонов и цистерн, вручную размеченных и вырезанных из реальных изображений. Фоновые изображения получались из тех же реальных изображений за исключением из них областей, содержащих номер. Обучающая выборка состояла из порядка 4000 изображений номерных знаков и порядка 4000 фоновых изображений, приведенных к эталонному размеру 48x12 пикселей.

В результате обучения построен каскадный классификатор, для обнаружения изображений номерных знаков, состоящий из 3-х слоев, и имеющий характеристики, приведенные в таблице 1.

**Таблица 1.** Структура и характеристики каскадного классификатора.

Номер слоя в каскаде	Число слабых классификаторов	Вероятность правильного обнаружения	Вероятность ложного срабатывания
1	10	0.993	0.25

2	25	0.99	0.027
3	50	0.99	0.0033
Весь каскад	85	0.973	$2.2 \cdot 10^{-7}$

Построенный в результате машинного обучения классификатор использует для принятия решения ограниченное количество признаков и работает в режиме реального времени. На рис.3 показаны фильтры признаков номерных знаков, которые по версии данного алгоритма обучения обладают наилучшими обобщающими свойствами.



**Рис.3.** Фильтры признаков номерных знаков, обладающих наилучшими обобщающими свойствами по версии данного алгоритма обучения.

Примеры работы алгоритма обнаружения номерных знаков на основе обученного каскадного классификатора показаны на рис.4.



**Рис.4.** Примеры работы алгоритма обнаружения номерных знаков железнодорожного подвижного состава.

Проведённое математическое моделирование показало, что при работе по случайно выбранной совокупности тестовых изображений рассмотренный алгоритм автоматического обнаружения изображений номерных знаков железнодорожного подвижного состава обеспечивает вероятность правильного обнаружения не менее 0.95 при вероятности ложного обнаружения цифробуквенной информации не более 0.01.

## 4. ЗАКЛЮЧЕНИЕ

Предложен подход к решению задачи обнаружения номерных знаков железнодорожного подвижного состава на основе метода адаптивного бустинга. Формирование информативных признаков на основе преобразований Хаара позволяет построить простой, быстрый и эффективный алгоритм обнаружения изображений номерных знаков. Полученный в результате машинного обучения классификатор использует для принятия решения ограниченное количество признаков и работает в режиме реального времени. Разработанный алгоритм обнаружения номерных знаков может быть использован в системах технического зрения регистрации железнодорожных составов.

## 4. БЛАГОДАРНОСТИ

Работа была выполнена при поддержке гранта РФФИ 12-07-13132 офи\_м\_РЖД.

## 5. СПИСОК ЛИТЕРАТУРЫ

- [1] X. Chen and A. L. Yuille. Detecting and reading text in natural scenes. CVPR, 2:366–373, 2004.
- [2] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 2963–2970.
- [3] K. Kim, K. Jung, and J. Kim. Texture-base approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 25, no. 12, pp. 1631–1639, 2003.
- [4] J.-J. Lee, P.-H. Lee, S.-W. Lee, A. Yuille, and C. Koch. Adaboost for text detection in natural scene. Proceedings of International Conference on Document Analysis and Recognition (ICDAR), 2011, pp. 429–434.
- [5] C. Mancas-Thillou and B. Gosselin. Color text extraction with selective metric-based clustering. Computer Vision and Image Understanding, vol. 107, no. 1-2, pp. 97–107, 2007.
- [6] P. Viola, M. Jones. Robust Real Time Object Detection // IEEE ICCV Workshop Statistical and Computational Theories of Vision. – 2001. – Pp. 1- 25.
- [7] C. Yi and Y. Tian. Text string detection from natural scenes by structure-based partition and grouping. IEEE Trans. Image Processing, vol. 20, no. 9, pp. 2594–2605, 2011.
- [8] Бекетова И.В., Каратеев С.Л., Визильтер Ю.В., Желтов С.Ю. Алгоритм автоматического обучения каскадного классификатора на основе метода адаптивного усиления AdaBoost. // Многопроцессорные вычислительные и управляющие системы (МВУС-2009): Материалы международной научно-технической конференции. Т.2, с.239-240 – Таганрог: Изд-во ТТИ ЮФУ, 2009

# Концептуальный компонент в гибридной системе анализа изображений

Геннадий Хахалин, фрилансер

gkhakhalin@yandex.ru

Сергей Курбатов, Алексей Лобзин

Научно-Исследовательский Центр Электронной Вычислительной Техники, Москва, Россия

curbatow.serg@yandex.ru

lobzin@rambler.ru

## Аннотация

В работе рассматривается концептуальный компонент гибридной системы анализа изображений. В качестве базы знаний для этого компонента служит прикладная онтология, которая представлена на языке семантического гиперграфа.

Предлагается достаточно гибкое взаимодействие двух компонент гибридной системы. Показано, каким образом привлекаются другие части интегральной системы искусственного интеллекта (синтез изображений и анализ, и синтез естественного языка) для «инспектирования» процесса распознавания и концептуального «дообучения». Многие моменты этих процессов иллюстрированы примерами из прикладной области.

**Ключевые слова:** гибридная система, концептуальный анализ изображения, предметная онтология, семантический гиперграф.

## 1. ВВЕДЕНИЕ

При решении многих задач распознавания изображений однокомпонентные системы оказываются неэффективными. И если при распознавании «непроизводных» элементов эффективными оказываются дискриминантные (числовые) методы, то при распознавании композиционных объектов в целом, обладающих существенной структурной информацией, ведущая роль принадлежит структурным (символьным) методам [3].

## 2. ГИБРИДНАЯ СИСТЕМА АНАЛИЗА

Под задачей полного анализа изображения подразумевается описание распознаваемых объектов и их расположение на изображении в терминах и в структурах языка прикладной онтологии. Результатом анализа должно быть описание ситуации на изображении, состоящей из экземпляров распознанных классов объектов с их означенными характеристиками и отношениями между ними.

На входном изображении в общем случае задается не один образ, а целая ситуация (композиция) и ее надо описать в терминах взаимосвязей означенных структур классов объектов.

В качестве предметной среды рассматривается мир плоских двумерных изображений объектов: планиметрические фигуры и «детские» рисунки (прямоугольники, трапеции, окружности и т.п., и кораблики, домики, ромашки и т.п.).

Выбрана двухуровневая система: нижний уровень, который непосредственно работает с изображением, и верхний уровень, который работает с моделью изображения (по результатам работы системы нижнего уровня), но с мощным языком символьного (концептуального) описания среды. Систему нижнего уровня назовем PR-система (Pattern Recognition), а систему верхнего уровня – KB-система (Knowledge Based). Они работают последовательно с возможностью управления процессом распознавания на нижнем уровне с концептуального уровня, реализуя т.н. «экслюзивный» анализ (см. ниже).

Функциональная схема двухуровневой системы анализа изображений представлена на рис. 1.

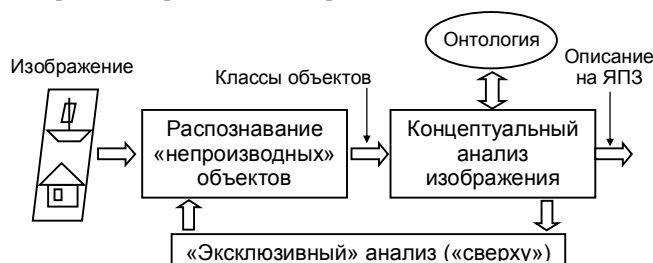


Рис 1: Схема двухуровневой системы анализа изображений.

Данная гибридная система анализа изображений не является изолированной, а представляет собой элемент интегральной системы, в которую входят дополнительно системы концептуального синтеза изображений и анализа и синтеза текстов естественного языка [6]. Взаимодействие этих систем осуществляется на основе общей прикладной онтологии [1].

Для иллюстраций в выбранной предметной области мы используем реальную PR-систему MyScript Notes [2]. Эта система решает стандартные задачи предварительной обработки изображений, включая кодирование, сглаживание, фильтрацию, сегментацию и т.д., избавляя от этих процедур верхний уровень гибридной системы. На вход программы MyScript Notes подаются нарисованные геометрические рисунки. На выходе получаем для каждого класса распознаваемых объектов: имя класса, вероятность распознавания и значения геометрических параметров, привязанных к полю изображения (координатной системе). Кроме того дается альтернативная информация распознавания, когда приводятся возможные другие кандидаты на распознавание данного объекта с соответствующей вероятностью.

Множество распознаваемых объектов ограничивается перечнем планиметрических фигур (отрезок прямой,

треугольник, прямоугольник, стрелки, ромб, дуга, окружность и др.) и классом неизвестных объектов (рисунки произвольной, нераспознаваемой формы - *drawing*).

Пример изображения композиции приведен на рисунке 2. На нем представлены как объекты, распознаваемые программой MyScript Notes (*прямоугольник, окружность* и др.), так и нераспознаваемые на этом уровне классы объектов (*домик, кораблик, паровозик* и др.).

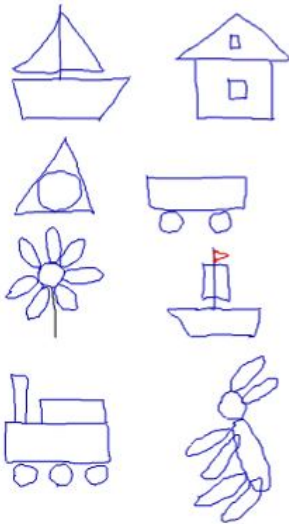


Рис 2: Композиция фигур.

Результатом распознавания, например *Паровозика*, будет список, представленный фрагментом на рис. 3, где указаны имена предполагаемых классов с указанием вероятности распознавания и конкретных характеристик объектов, привязанных к системе координат изображения (подчернутые имена классов имеют максимальную вероятность).

rectangle 0.753047 625.545 395.106 764.597 390.574 777.322 780.953 638.269 785.485  
 parallelogram 0.752861 624.92 390.086 764.34 396.322 777.918 785.459 638.498 779.224  
rectangle 0.894784 877.211 785.921 876.636 587.349 1391.45 585.859 1392.02 784.432  
 parallelogram 0.89469 880.424 785.418 872.933 586.889 1388.14 586.386 1395.63 784.915  
rectangle 0.956212 1440.37 1090.52 584.024 1099.19 580.978 798.668 1437.32 789.989  
 parallelogram 0.95617 1445.39 1091.08 589.294 1098.52 575.977 798.106 1432.07 790.666  
circle 0.67984 1281.93 1187.42 75.2383  
 ellipse 0.679622 1281.93 1187.42 78.6176 71.8591 -1.4689  
 arc 0.679404 1281.93 1187.42 78.6176 71.8591 -1.4689 0 6.28319  
circle 0.609414 746.45 1192.97 82.0498  
 ellipse 0.609176 746.45 1192.97 83.6068 80.4929 -0.325574  
 arc 0.608938 746.45 1192.97 83.6068 80.4929 -0.325574 0 6.28319  
circle 0.609228 1024.86 1177.54 75.0216  
 ellipse 0.60899 1024.86 1177.54 80.0398 70.0035 -0.590248  
 arc 0.608752 1024.86 1177.54 80.0398 70.0035 -0.590248 0 6.28319

Рис 3: Фрагмент результата работы PR-системы.

Для нераспознанных и нераспознаваемых PR-системой объектов как раз и служит верхний уровень гибридной системы – KB-система.

### 3. КОНЦЕПТУАЛЬНЫЙ АНАЛИЗ ИЗОБРАЖЕНИЙ

Основным результатом работы KB-системы является описание, состоящее из экземпляров распознанных классов объектов на языке представления знаний. Поскольку система работает в комплексе, результат может быть также

представлен в виде синтезированного изображения композиции или в виде текста на естественном языке.

В качестве основы для системы анализа верхнего уровня выбран метод контекстного целенаправленного анализа изображений, который подробно описан в [1].

Построение описаний в онтологии предполагает наличие достаточно выразительных средств представления знаний о изображениях. Прикладная онтология организуется в виде множества структур, каждая из которых соответствует обобщенному описанию некоторого класса объектов на изображении с его внутренней организацией и взаимосвязями с другими классами.

На наш взгляд наиболее адекватным языком представления концептуальных знаний является язык гиперграфов в качестве расширения семантических сетей, где естественным образом представляются *n*-арные отношения, которые позволяют задавать не только атрибуты объектов, но и представлять структурные, «целостные» описания объектов с взаимосвязями своих компонентов. Здесь вершины соответствуют элементам описываемого изображения, а дуги – отношениям между этими элементами.

Если элементам гиперграфа приписаны цепочки символов (имена понятий и отношений онтологии), то такой гиперграф называется *семантическим гиперграфом* [4, 8].

Фрагмент прикладной онтологии в виде семантического гиперграфа представлен на рис. 3., где красные кружки – концепты онтологии (объекты среды), красные стрелки – родовидовые связи концептов, синие кружки и стрелки – свойства объектов, зеленые замкнутые области – описание структур объектов, бирюзовые стрелки – связь типа «часть-целое» («входит\_в\_структуру»).

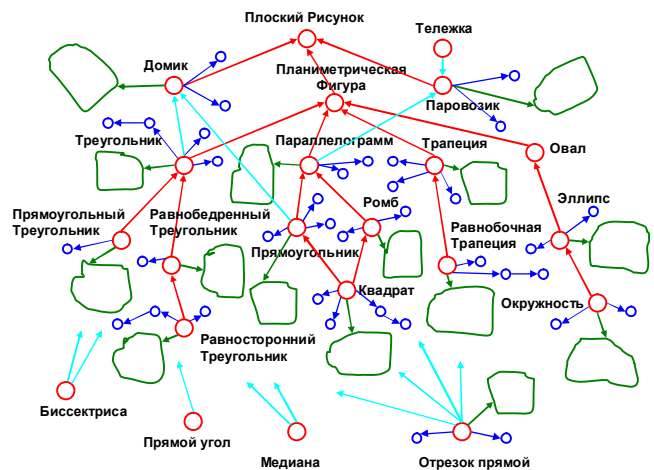


Рис 4: Фрагмент прикладной онтологии.

В прикладной онтологии представлены и понятия нижнего уровня (*точка, треугольник* и т.д.), дублирующие классы PR-системы, для исправления ошибок распознавания и для анализа тех классов, которые PR-системой не распознаются в принципе (например, *трапеция*), и структурные образы рисунков, составленных из классов PR-системы (*кораблики, домики, ромашки* и т.д.).

Алгоритм работы KB-системы заключается в следующем.

Представление результатов работы PR-системы в виде концептов «непроизводных» объектов – получение гиперграфа модели изображения.

Выдвижение кандидатов-гипотез из онтологии на сопоставление (поиск пути в семантическом гиперграфе).

Сопоставление графа модели изображения с графами структур объектов прикладной онтологии (поиск изоморфизма в семантическом гиперграфе). Требуемые по онтологической структуре отношения проверяются на координатной сетке с помощью вызова присоединенных процедур, связанных с этими отношениями.

В случае удачи полного сопоставления – получение означенного экземпляра онтологической структуры понятия.

В случае частичного сопоставления – выделение «ареала» на изображении и поиск гипотетического непроизводного элемента с помощью PR-системы («что хотим найти» - предполагаемый непроизводный объект, «где хотим найти» - ареал, т.е. «место обитания») этого гипотетического объекта, и «с какими ограничениями» - необходимые диапазоны изменения параметров: вся эта информация берется из онтологической структуры).

Эффективный поиск путей в онтологии и выбор наиболее оптимальных гипотез осуществляется «хождением» по семантическому гиперграфу: по связям «вх\_в\_структуру», «род-вид» и др.

### 3.1 Эксклюзивный анализ

Эксклюзивный анализ работает в двух режимах:

- Последовательный анализ: PR-система полностью обрабатывает изображение композиции, а потом включается KB-система и при «коллизиях» она обращается к PR-системе.
- Итеративный анализ: PR-система частично обрабатывает изображение, после этого подключается KB-система, выдвигает гипотезы «чего и где искать» и передает управление на PR-систему.

В первом случае KB-система выдвигает гипотезу о «дораспознавании» на реальном изображении. Для вызова соответствующих процедур PR-системы она формирует управляющую информацию, которая включает: что надо искать (это известно из онтологической структуры-гипотезы); где надо искать (вычисленный сегмент-«ареал» по модели изображения); с какими ограничениями, т.е. с необходимыми параметрами (из онтологической структуры).

После получения управляющей информации PR-система реально анализирует выделенную область и реализует процедуру распознавания только в этом сегменте, как будто на целом изображении присутствуют только компоненты этого объекта. Реальные эксперименты показывают, что присутствие композиции усложняет процесс распознавания отдельных объектов, поскольку глобального априорного решения задачи сегментации не существует.

Если в результате повторного распознавания будет выделен предполагаемый объект, то KB-система корректно выполнит свое предназначение. В противном случае выдвигаются

другие гипотезы или система распознавания обращается к пользователю за советом.

При итеративном анализе реализуется стратегия «сверху-вниз» (от онтологии к изображению) и при «стандартном» распознавании. По частичным результатам PR-системы выдвигаются гипотезы, которые последовательно проверяются на реальном изображении. При этом процесс сегментации носит «осмысленный» целенаправленный характер, т.к. она проводится на основе частичных результатов распознавания, выдаваемых PR-системой, в рамках проверяемых гипотез, которые формируются KB-системой.

### 3.2 Использование концептуального синтеза

Концептуальный синтезатор изображений по означенному онтологическому описанию графической ситуации строит и визуализирует на плоскости отображения графическое изображение. Он состоит из планировщика и визуализатора. Планировщик реализует «обход» онтологической структуры и выделение описаний подобъектов для визуализатора; согласование параметров подобъектов и означивание их свойств; генерация текста для визуализации. В качестве визуализатора выбран пакет OpenGL, на вход которого поступает текст интерпретации планировщика, а выходом является 3D или 2D изображение (более подробно см. в [5, 7]).

Этот синтезатор используется в двух ипостасях: как средство отображения результатов работы гибридной системы распознавания и как средство отладки при модификации и пополнении онтологии.

В первом случае визуализируются результаты распознавания. Небольшой фрагмент визуализации распознанной композиции (рис. 2) дан на рис. 5 (представлен конкретный экземпляр образа *Паровозик*).



Рис 5: Фрагмент визуализации результата распознавания.

Наименование концепта необходимо, поскольку без него не совсем понятно на каком уровне распознан объект: на уровне составляющих (три прямоугольника и три окружности распознает PR-система) или на уровне целостного объекта (распознает KB-система). Синтезированное изображение при этом будет одно и то же.

Другое использование концептуального синтезатора описано в следующем пункте.

### 3.3 Модификация и пополнение онтологии

Естественно KB-система «видит только то, что знает», т.е. распознает комплексные объекты, информация о которых присутствует в онтологии. В интегральной системе модификация и расширение онтологии предполагает использование двух каналов: для инженера по знаниям (для



разработчика) – непосредственный ввод структур новых концептов или модификацию старых на языке семантического гиперграфа; для пользователя – то же самое на естественном языке. В отличие от стандартных методов обучения для статистических систем такое обучение носит «дедуктивный» характер. Чтобы такое дедуктивное обучение не приводило к «хаосу», необходимо применить методику наполнения онтологии с верификацией вводимых описаний с помощью системы концептуального синтеза изображений. По новой введенной структуре система синтезирует множество изображений. Если с точки зрения учителя (разработчика или пользователя) множество корректно, то структура фиксируется в онтологии. Если среди синтезированных объектов присутствуют изображения, не относящиеся к данному классу, то структура модифицируется. И так до тех пор, пока множество не будет корректным.

Проиллюстрируем это на примере концепта *Паровозик*. До ввода структуры объекта *Паровозик* в онтологию гибридная система распознавания представит его описание в виде взаимосвязанного набора прямоугольников и окружностей. Введя в онтологию описание структуры понятия *Паровозик*, как это представлено на рис. 6., и получив множество синтезированных изображений, данных на рис. 7. по этой структуре, можно предположить, что теперь гибридная система распознавания будет распознавать изображение как целостный объект планиметрической среды.

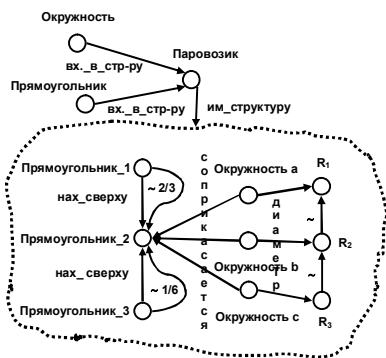


Рис 6: Онтологическая структура концепта *Паровозик*.

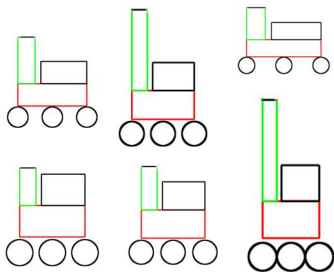


Рис 7: Множество синтезированных изображений концепта *Паровозик*.

#### 4. ЗАКЛЮЧЕНИЕ

Подобная гибридная система дает возможность говорить о «понимании» изображения, если рассматривать этот процесс в рамках интегральной системы.

Дальнейшая работа в рамках данной системы направлена на расширение онтологии для других областей и на более скрупулезную проработку вопросов верификации онтологии.

#### 5. БЛАГОДАРНОСТИ

Работа была выполнена при поддержке гранта РФФИ, проект № 12-07-00531а.

#### 6. ССЫЛКИ

[1] Khakhalin G., Kurbatov S., Naidenova K., Lobzin A. *Integration of the Image and NL-text Analysis/Synthesis Systems. In book: "Intelligent Data Analysis for Real-Life Applications: Theory and Practice" (Ed. Rafael Magdalena et al.). – N.Y., USA: IGI Global, 2012, p. 160-185.*

[2] MyScript Notes, 2010. *Режим доступа: http://www.visionobjects.com/en/webstore/myscript-studio/description/*. – [Электронный ресурс].

[3] Roli Fabio, Serpico Sebastiano B., and Vernazza Gianni. *A Hybrid System for Two-Dimensional Image Recognition // Proceedings of the IEEE, VOL. 84, NO. 11, November 1996, pp. 1659-1680.*

[4] Баранович А.Е. *Семиотико-хроматические гиперсети: унифицированная модель представления знаний // Труды Международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» OSTIS-2011, Минск, Беларусь. – 2011. – С. 71-86.*

[5] Курбатов С.С., Литвинович А.В., Лобзин А.П., Хахалин Г.К. *Концептуальный синтез графических образов по структурам прикладной онтологии // Труды XIII национальной конференции по Искусственному Интеллекту, Т. 2.. – Белгород: БГТУ, 2012. – С. 120-127.*

[6] Курбатов С.С., Лобзин А.П., Найденкова К.А., Хахалин Г.К. *Гибридная схема анализа изображений // Труды Международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» OSTIS-2012. Минск, Беларусь. – 2012. – С. 327-334.*

[7] Литвинович А.В. *Система синтеза изображений по тексту на естественном языке // Динамика сложных систем — XXI век, № 2, 2013, С. 65-68.*

[8] Хахалин Г.К. *Прикладная онтология на языке гиперграфов // Труды второй Всероссийской Конференции "Знания-Онтологии-Теории" (ЗОНТ-09). Новосибирск, 2009, с. 223-231.*

#### Об авторах

Геннадий Хахалин – фрилансер. Его адрес: [gkhakhalin@yandex.ru](mailto:gkhakhalin@yandex.ru).

Сергей Курбатов – к.т.н., НИЦЭВТ. Его адрес: [curbatow.serg@yandex.ru](mailto:curbatow.serg@yandex.ru).

Алексей Лобзин – с.н.с., НИЦЭВТ. Его адрес: [lobzin@rambler.ru](mailto:lobzin@rambler.ru).

# Стеганографическое преобразование цветного изображения в полутоновое для распределения цифровых фотографий

В.Н. Горбачев, И. К. Метёлев, Е.М. Кайнарова, Е.С. Яковлева

Северо-Западный институт печати

Санкт-Петербургского государственного университета технологии и дизайна, Санкт-Петербург, Россия.

{Елена Кайнарова} helenkainarova@gmail.com

## Аннотация

Для процесса преобразования цветного изображения в полутоновое и обратно рассмотрена стеганографическая система с секретным ключом. Для встраивания использованы битовые плоскости яркостной компоненты модели YUV и оригинальный алгоритм блочного кодирования, который позволяет скрывать информацию в старших битовых плоскостях. С помощью предложенной системы обсуждается задача о распределении цифровых изображений, когда пользователь может предварительно ознакомиться с интересующим его изображением. Представлен протокол, который позволяет исключить незаконное копирование цветного оригинала.

**Keywords:** *стеганография, преобразование цветного изображения в полутоновое*

## 1. ВВЕДЕНИЕ

Современная стеганография, которая занимается встраиванием цифровых данных одного объекта в цифровые данные другого, имеет широкое применение для защиты медианформации в частности для охраны авторского права. Однако, методы стеганографии позволяют решать ряд других задач, в число которых входит преобразование цветного изображения в полутоновое и обратно, известное как Color to Gray and Back Reversible Transformation (CGRT). Идея заключается в том, чтобы информацию о цвете исходного изображения с помощью стеганографии встроить в его полутоновую версию, а потом, например, после печати извлечь из черно-белого оттиска цвет и восстановить оригинал.

В литературе указывают следующие примеры применений такого преобразования.

Считают, что оно может быть полезным для офисов, где цветные документы появляются редко, поэтому для печати используется централизованный мощный черно-белый принтер [1]. Другим примером служит система резервного копирования Hardcopy Image Backup Copy для хранения аналоговых копий, сделанных с документов и цветных фотографий [2]. Известно, что при длительном хранении, 30-50 лет, происходит изменение цвета вплоть до полной его потери. Поэтому система создает копию со встроенной информацией, которая позволяет восстановить исходный цвет изображения.

Особенность CGRT можно пояснить следующим образом. Для заданного RGB изображения в качестве его полутоновой версии может служить яркостная компонента из модели YCbCr, что соответствует телевизионному стандарту NTSC (National Television Standards Committee). При этом для обычного представления цветной пиксел с энтропией 24 бита

заменяется на полутоновый с энтропией 8 бит. Энтропия не сохраняется, поэтому преобразование необратимо как с физической, так и математической точек зрения. Решение дает стеганография, ее методы позволяют встроить информацию о цвете в полутоновую версию, из которой потом восстанавливается цветное изображение. Строго говоря, такое преобразование также необратимо, однако оригинал и восстановленное изображение оказываются визуально неразличимы, поэтому говорят, что преобразование обратимо.

Для встраивания цвета в полутоновое изображение могут использоваться стандартные стеганографические методы, такие как дискретное вейвлет-преобразование [3]. Можно использовать особенности процесса печати, встраивая информацию через алгоритмы растривания [4] путем модулирования параметров оттиска.

Цель настоящей работы рассмотреть схему встраивания цвета с секретным ключом.

Ключ в соответствии с принципом Керхгоффса обеспечивает теоретическую секретность системы и вместе с тем позволяет передать изображение пользователю, которому оно предназначено.

В стеганографии роль ключа часто играет позиция пиксела, в которую встраивается информация, если пиксел выбирается случайно [5]. Вместо этого мы выбрали случайную матрицу. Она используется вместе с нашим алгоритмом блочного кодирования [6]. Этот алгоритм основан на условии сохранения яркости блока, в который встраивается бит сообщения, что увеличивает объем скрываемой информации.

Применение нашей схемы мы рассмотрели на примере распределения цифровых изображений из банка цветных фотографий. Задача заключается в том, чтобы пользователь предварительно ознакомился с интересующим его изображением перед тем как его приобрести. Задача является актуальной, поскольку сейчас спрос на высококачественные цифровые изображения, в частности на базы с экспертными оценками, растет. Предложенное решение основано на компромиссе, который достигается, если использовать идею  $\beta$ -версии. Вместо того чтобы предоставить пользователю полноценное цветное изображение, он получает полутоновую версию со встроенным цветом. Мы описали протокол, который с помощью распределения секретного ключа, позволяет законному пользователю восстановить из  $\beta$ -версии исходное цветное изображение.

Применение преобразования цветного изображения в полутоновое и обратно для этой задачи авторам неизвестно.

## 2. ПРЕОБРАЗОВАНИЕ ЦВЕТНОГО ИЗОБРАЖЕНИЯ В ПОЛУТОНОВОЕ

Задача Color to Gray and Back Reversible Transformation является интересным примером применения стеганографических методов для приложений. Мы кратко поясним ее на основе результатов [1], где использовано дискретное вейвлет-преобразование.

Исходное цветное изображение представлено в модели YCbCr, яркостная компонента которой Y играет роль контейнера для встраивания цвета, информация о котором представлена в хроматических компонентах Cb и Cr. Дискретное вейвлет-преобразование яркостной компоненты DWT(Y) состоит из четырех блоков коэффициентов, каждый из которых имеет размер равный четверти исходного изображения. Два блока заменяются на одну четверть коэффициентов из хроматических компонент Cb и Cr.

Получается новый вейвлет-образ яркостной компоненты Y', от которого вычисляется обратное вейвлет-преобразование IDWT(Y')=Yc. В результате получается полутоновая версия исходного цветного изображения, это Yc, в которую встроены цвет. Восстановление цветного изображения осуществляется в обратном порядке. Поскольку встроено только 1/4 всей информации о цвете, восстановленное изображение не равно исходному, однако они визуально неразличимы равно как и полутоновые компоненты Y и Yc.

Мы использовали стеганографическую систему с секретным ключом. В ней сообщение M встраивается в контейнер C с помощью секретного ключа K. Заполненный контейнер S называют стегоконтейнером.

Если E и D алгоритмы встраивания и детектирования, то стеганографическая система описывается соотношениями [7]

$$E: C \otimes M \otimes K \rightarrow S, \\ D: C \otimes S \otimes K \rightarrow M.$$

Два основных свойства стеганографической системы это неразличимость пустого и заполненного контейнера,  $C \approx S$  и секретность, которая обеспечивается ключом.

В нашей схеме цветное RGB изображение преобразовывается в модель YUV с яркостной компонентой Y и хроматическими компонентами X=U,V, из которых с помощью секретного ключа строились сообщения

$$M = (K + X) \bmod(255), \quad (1)$$

где K случайная матрица, размеры которой равны размеру исходного изображения. Далее M встраивается в яркостную компоненту Y, которая играет роль контейнера.

## 3. БЛОЧНОЕ ВСТРАИВАНИЕ С СОХРАНЕНИЕМ ЯРКОСТИ

Алгоритм блочного встраивания [6] использует битовые плоскости яркостной компоненты. Он работает следующим образом. Каждая битовая плоскость  $Y_V, V=1, \dots, 8$ , где младшая -  $Y_1$ , разделяется на непересекающиеся блоки  $Y_{V_a}$  размера  $h \times h$ . В каждом блоке определяется бит четности побочной диагонали

$$d_a = \bigoplus_{x \in Y_{V_a}} y[x, h - x]$$

Встраивание бита сообщения m в блок  $Y_{V_a}$  осуществляется по правилу

$$E: Y_{V_a} \rightarrow S_{V_a} = \begin{cases} Y_{V_a}, & \text{if } d_a \oplus m = 0, \\ ZY_{V_a}, & \text{if } d_a \oplus m = 1. \end{cases}$$

Здесь оператор Z осуществляет либо изменение бита в побочной диагонали, либо ищет блок  $S_{V_a}$  с яркостью равной или близкой к яркости  $Y_{V_a}$ . В результате яркость блока сохраняется максимальным образом.

Приведем пример работы оператора. Z для  $h=2$

$$M = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \rightarrow Y_{V_a} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}.$$

$$S_{V_a} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Здесь сообщение представлено  $2 \times 2$  матрицей M. Каждый бит сообщения встраивается в один из четырех блоков  $Y_{V_a}$ . Стегоизображение  $S_{V_a}$  состоит из четырех блоков. Видно, что только один из блоков, правый верхний, изменил яркость (сумму элементов) на 1.

Алгоритм детектирования не использует исходного контейнера, такое детектирование называют слепым. Встроенное сообщение M извлекается из блока стегоизображения  $S_{V_a}$  с помощью вычисленного бита четности побочной диагонали. Его четность 0 или 1 равна значению бита сообщения. Затем, используя секретный ключ, находят хроматические компоненты  $X=U,V$

$$X = (M - K + 255) \bmod(255).$$

Условие сохранения яркости позволяет увеличить объем встраиваемой информации. Так, оказывается возможным использовать больше битовых плоскостей, начиная от младшей  $V=1$  вплоть до  $V=5,6$ , при этом сообщение не будет «просвечивать» [6]. Чем больше встроено информации о цвете в полутоновый контейнер, тем меньше будет отличаться от оригинала восстановленное изображение.

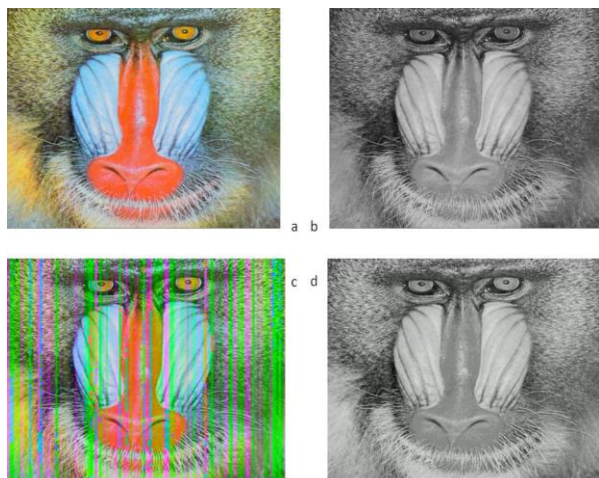
## 4. ЭКСПЕРИМЕНТ

Исходное RGB-изображение, рис.1-а, преобразовывалось в модель YUV. Компонента Y, в которую будет встраиваться цвет, приведена на рис.2-б. Число бит  $H_Y$ , которые можно зарезервировать в Y, определяется числом взятых битовых плоскостей T, числом пикселей n и величиной блока h, который используется алгоритмом встраивания:  $H_Y = nT/h^2$ . Ясно, что весь объем цветовой информации не встроить.

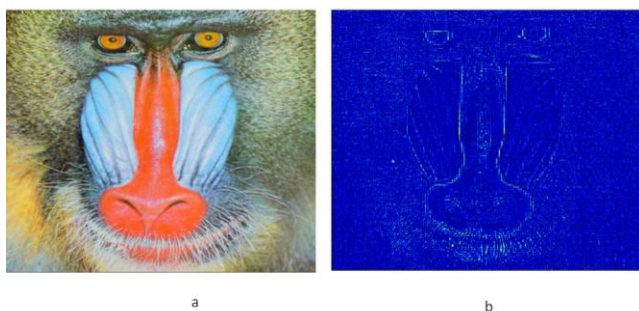
Поэтому разрешение хроматических компонент  $X=U,V$  уменьшалось путем усреднения по окрестности  $a \times a$ . В телевизионном стандарте передачи данных  $a=2$ , это называют 4:1:1. Мы использовали случай  $a=4$ . В результате объем цветовой информации  $H_C$  будет равен 1/8 от Y или  $H_C=(1/8)nk$ , где k битовая глубина канала Y,U и V. Обычное значение  $k=8$ . При  $h=2$  найдем, что  $H_Y = (T/4)H_C$ . Это означает, что при  $T=4$  мы встроим всю информацию о цвете  $H_C$ . Наш метод кодирования максимальным образом сохраняет яркость блоков. Поэтому при использовании не одной, а четырех битовых плоскости сокрытая в Y информация не «просвечивает». На рис.1-д показана яркостная компонента, в четыре битовых плоскости которой  $V=1,2,3,4$  встроены цвет.

Чтобы восстановить исходное цветное изображение нужен ключ. Если ключ неизвестен, то подобрать его практически

невозможно. Формально здесь возникает задача перебора  $2^{nk}$  полутоновых матриц, что с вычислительной точки зрения является сложной задачей класса NP. На рис.1-с приведен пример восстановленного изображения, когда некоторые биты в ключе специально изменены, «испорченный ключ». Однако, если взять исконный ключ, то восстановленное изображений, рис.2-а, визуально не отличить от исходного. Вместе с тем, различие есть. На рис.2-б приведена разность между синими компонентами.

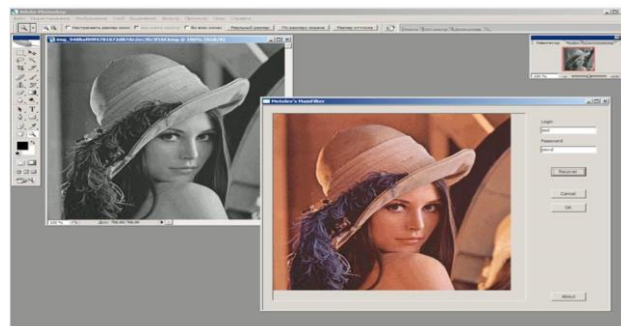


**Рис. 1:** Встраивание цвета в полутоновое изображение. а) исходное цветное изображение; б) яркостная компонента модели YUV, в которую встраивается цвет; в) восстановленное цветное изображение с неполноценным ключом; д) яркостная компонента со встроенным цветом.



**Рис. 2:** Восстановление цветного изображения из полутонового. а) восстановленное изображение; б) разность между синими каналами исходного и восстановленного изображения.

Работа рассмотренной схемы встраивания цвета реализована в полнофункциональной информационной системе, включающую в себя клиентскую часть, интегрированную в пакет *Adobe Photoshop*, рис.3, и серверную часть, сочетающую в себе веб-сервер и сервер приложений.



**Рис. 3:** Интерфейс в Adobe Photoshop.

## 5. ПРОТОКОЛ РАСПРЕДЕЛЕНИЯ ЦВЕТНЫХ ИЗОБРАЖЕНИЙ

Приведенная схема встраивания цвета с секретным ключом дает решение задачи о распределении цветных изображений, которые собраны в фотобанке. Задачу можно сформулировать следующим образом. У Алисы есть цветное изображение *A*, которое она хочет продать. Боб желает приобрести его, но предварительно он хочет ознакомиться с ним, получив его в свое распоряжение. Однако Алиса ему не доверяет.

Решение основано на следующем компромиссе. В целях осторожности Алиса передает Бобу вместо цветного изображения *A* полутоновое *B*, в яркостной компоненте которого встроена информация о цвете. После того, как Боб сообщил о своем намерении приобрести интересующее его изображение, Алиса высылает ему секретный ключ и Боб извлекает из *B* цветное изображение.

Компромисс использует идею версии, он заключается в том, что Алиса сохраняет свои авторские права, будучи защищена секретным ключом, а Боб получает возможность детально ознакомиться, однако, с полутоновым изображением.

Разумеется, эту задачу можно решить иначе. Пусть у Алисы есть два отдельных изображения, цветное *A* и его полутоновая версия *B<sub>0</sub>*, которые оба должны быть отправлены через канал получателю.

Это простое решение, но в нем есть недостаток, который заключается в том, что злоумышленник сразу может получить цветное изображение *A* при атаке на канал. В первом случае эта возможность исключена, поскольку по каналу цветное изображение в явном виде не передается, а для передачи секретного ключа Алиса может использовать стандартные протоколы криптографии.

## 6. ВЫВОДЫ

Методы стеганографии позволяют решать задачи, которые не сводятся только к защите медиаинформации и охране авторского права. В области обработки изображений хорошо известны необратимые преобразования цветного изображения в полутоновое. Однако такое свойство как обратимость может быть рассмотрено с точки зрения зрительного восприятия, когда исходное цветное изображение и изображение, восстановленное из его полутоновой версии, оказываются неразличимы. Для этого случая в литературе используется специальный термин Color to Gray and Back Reversible Transformation.

Чтобы восстановить цветное изображение из полутонового, в последнем должна быть информация о цвете, которая встраивается и извлекается методами современной стеганографии.

В нашей работе рассмотрен вариант стеганографической системы с секретным ключом. Наш алгоритм блочного кодирования позволяет увеличить объем встраиваемой информации, путем использования старших битовых плоскостей вплоть до 4-5 включительно без заметного визуального искажения.

Схемы с Color to Gray and Back Reversible Transformation в литературе традиционно включают процесс печати, когда цветное изображение восстанавливается из полутонового оттиска. Однако, процесс печати очень сильно усложняет задачу извлечения встроеной информации.

Вместо этого мы рассмотрели применение для случая распределения цифровых изображений, которые собраны в фотобанке. Задача заключается в том, чтобы пользователь мог предварительно ознакомиться с интересующей его картинкой перед тем как ее купить. Предложенное решение построено на идее  $\beta$ -версии, когда пользователю вместо полноценного цветного изображения предоставляется его полутоновая версия. Полутоновая версия содержит информацию о цвете, который можно извлечь, однако, лишь при наличии секретного ключа. Мы привели протокол, который позволяет исключить незаконное копирование исходного цветного изображения.

## 7. REFERENCES

- [1] Braun K. and de Queiroz R.L. Color to Gray and Back: Color Embedding Into Textured Gray Images. Proc. IS&T/SID 13th Color Imaging Conference, pp.120-124, 2005.
- [2] Hoarau E., Tastl I., Moroney N.A. Hardcopy Backup and Reconstruction System for Digital Images. Proc.of 2009 ICIP, Cairo, Egypt Nov., 2009.
- [3] de Queiroz R. L., Braun K. M. Color to Gray and Black: Color Embedding Into Texture Gray Image. ISEE Translation on Image Processing, V. 15, No. 6, JUNE 2006, p. 1464-1470.  
Ko K.-W., Kwon O.-S., Son C.-H., Kwon E.-Y., and Ha Y.-H. Color Embedding and Recovery Based on Wavelet Packet Transform / Journal of Imaging Science and Technology. — 2008. — vol. 52.— pp. 010501-1–010501-10.
- [4] Oztan B., Sharma G. Multiplexed Clustered-dot halftone watermarks using bi-directional phase modulation and detection. Proceedings of 2010 IEEE 17th International Conference on Image Processing, Hong Kong, September 26-29, 2010.  
Ulichney R., Tastl I., Hoarau E. Analog Image Backup with Steganographic Halftones, Color Imaging XVI: Displaying, Processing, Hardcopy, and Applications, IS&T/SPIE Electronic Imaging Symposium, San Francisco Airport, CA, 7866-53, January 23-27, 2011.
- [5] Nohara F., Horiuchi T., Tominaga S. An Accurate Algorithm For Color to Gray and back / ICIP. — 2009. — pp. 485-488.
- [6] В.Н. Горбачев, Е.М. Кайнарова, И.К. Метелев. Один алгоритм блочного встраивания цифрового водяного знака в наименее значащие биты на основе условия равенства яркости. /Известия высших учебных заведений. Проблемы полиграфии и издательского дела. М: МГУП, 2010, №2, с. 60 – 70.

Cox I.J., Miller M.L., Bloom J.F., Fridrich J., Kaler T. Digital Watermarking and Steganography', ELSEVIER, Morgan Kaufmann Publishers, 2008.

Fridrich J. Steganography in Digital Media. Principles, Algorithms, and Application, Cambridge University Press, New York, 2010.

## Авторы

Валерий Николаевич Горбачев, преподаватель Северо-Западного института печати СПб государственного университета технологии и дизайна (СЗИП СПб ГУТД).

Иван Кириллович Метелев, аспирант СЗИП СПб ГУТД, [ivan.metelev@yandex.ru](mailto:ivan.metelev@yandex.ru).

Елена Михайловна Кайнарова, преподаватель СЗИП СПб ГУТД, [helenkainarova@gmail.com](mailto:helenkainarova@gmail.com).

Елена Сергеевна Яковлева, преподаватель СЗИП СПб ГУТД, [2305lena@mail.ru](mailto:2305lena@mail.ru).

## О субполосном анализе изображений

Е.Г. Жилияков, А.А Черноморец

Факультет компьютерных наук и телекоммуникаций

Белгородский государственный национальный исследовательский университет,

Белгород, Россия

{zhilyakov, chernomorets}@bsu.edu.ru

### Аннотация

Рассматривается задача описания свойств изображений с позиций разбиения области пространственных частот на подобласти (субполосный анализ). При этом основное внимание уделяется задаче вычисления точных значений долей энергии изображения в заданных подобластях пространственных частот и оптимальному выделению субполосных компонент.

**Ключевые слова:** изображение, доли энергии, подобласть пространственных частот, субполосные компоненты

In this article we consider the image properties describing task using the decomposition of spatial frequencies area to subareas subband analysis). We consider methods of computing the accurate image energy parts values in the given subareas of spatial frequencies and optimal selection of image subband components.

**Keywords:** image, energy parts, subarea of spatial frequencies, subband components

### 1. ВВЕДЕНИЕ

Опыт показывает, что частотные представления [1-5] служат адекватной математической основой для многих задач обработки изображений

$$f_{ik} = \iint_{-\pi-\pi}^{\pi-\pi} F(x, y) \exp(jx(i-1)) \exp(jy(k-1)) dx dy / 4\pi^2, \quad (1)$$

где  $j$  – мнимая единица ( $j^2 = -1$ );  $F$  – частотная характеристика, в качестве которой наиболее часто используется трансформанта Фурье  $F^\Phi(x, y)$ ,

$$F^\Phi(x, y) = \sum_{k=1}^M \sum_{i=1}^N f_{ik} \exp(-jx(i-1)) \exp(-jy(k-1)), \quad (2)$$

где  $f_{ik}$ ,  $i=1,2,\dots,N$ ,  $k=1,2,\dots,M$ , – значения изображения  $\Phi = \{f_{ik}\}$  в соответствующих пикселях;  $x, y$  – пространственные частоты (ПЧ).

При этом многие задачи анализа и синтеза можно решать, используя разбиение области определения трансформанты Фурье на ряд подобластей пространственных частот (ПОПЧ) [2, 3], так что (1) принимает вид

$$f_{ik} = \frac{1}{4\pi^2} \sum_{s=1}^R \sum_{r=1}^S \iint_{(x,y) \in V_{sr}} F^\Phi(x, y) \exp(jx(i-1)) \exp(jy(k-1)) dx dy, \quad (3)$$

где

$$V_{sr} = \{ (x \in [-\Omega_{s2}, -\Omega_{s1}] \cup [\Omega_{s1}, \Omega_{s2}]) \cap (y \in [-\Omega_{r2}, -\Omega_{r1}] \cup [\Omega_{r1}, \Omega_{r2}]) \}, \quad (4)$$

$$\Omega_{s1} = 0 (s=1), \quad \Omega_{s,2} = \pi; \quad \Omega_{r1} = 0 (r=1), \quad \Omega_{r,2} = \pi;$$

$$\Omega_{s+1,1} = \Omega_{s,2}; \quad \Omega_{r+1,1} = \Omega_{r,2}.$$

Имеется в виду, что в подобласти  $V_{sr}$  переменная  $x$  принимает значения из интервала (субполосы) оси абсцисс плоскости ПЧ  $D_s = [-\Omega_{s2}, -\Omega_{s1}] \cup [\Omega_{s1}, \Omega_{s2}]$ , тогда как одновременно переменная  $y$  попадает в следующий интервал (субполосу) оси ординат  $G_r = [-\Omega_{r2}, -\Omega_{r1}] \cup [\Omega_{r1}, \Omega_{r2}]$ . Подобласть  $V_{sr}$  схематично изображена на рис. 1.

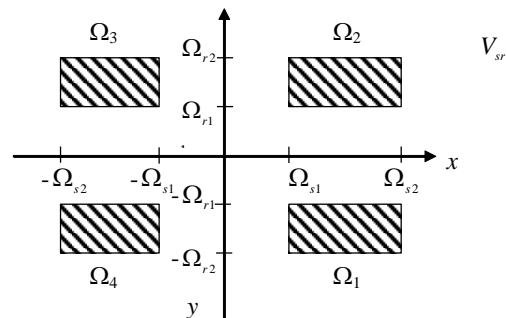


Рис. 1. Двумерная подобласть ПЧ  $V_{sr}$

В работе рассмотрены субполосные модели, которые означают описание изображений с позиций некоторого разбиения области определения двумерной трансформанты Фурье на подобласти, например выделение компонент, определяемых одним из интегралов в соотношении (3) или оценивание их энергий.

### 2. СУБПОЛОСНЫЕ КОМПОНЕНТЫ ИЗОБРАЖЕНИЙ

В рамках данной работы субполосной компонентой изображения из некоторой ПОПЧ вида (4) называется изображение  $Y^{sr} = \{y_{ik}^{sr}\}$ ,  $i=1,\dots,N$ ,  $k=1,\dots,M$ , определяемое соотношением

$$y_{ik}^{sr} = \frac{1}{4\pi^2} \iint_{(x,y) \in V_{sr}} F^\Phi(x, y) \exp(jx(i-1)) \exp(jy(k-1)) dx dy. \quad (5)$$

Существенное значение имеет то обстоятельство, что субполосная компонента полностью определяется двумерным отрезком трансформанты Фурье из выбранной ПОПЧ. Поэтому корректное выделение этой компоненты позволяет адекватно судить о ее свойствах в пространственной области. В частности, субполосные компоненты могут проявлять характер периодичности вдоль некоторого направления, например, вдоль одной из осей координат.

Возникает проблема вычисления субполосных компонент, так как непосредственное использование определения (5) невозможно, потому что требует вычисления интегралов от предварительно вычисленных в континууме точек ПОПЧ двумерных отрезков трансформант Фурье.

В работе доказано следующее утверждение. Условию (5) удовлетворяет следующая компонента

$$y_{ik}^{sr} = \sum_{n=1}^N \sum_{m=1}^M f_{nm} a_{in}^s b_{km}^r, \quad (6)$$

где

$$a_{in}^s = \int_{x \in D_s} \exp(-jx(i-n)) dx / 2\pi; \quad (7)$$

$$b_{km}^r = \int_{x \in G_r} \exp(-jx(k-m)) dx / 2\pi. \quad (8)$$

Напомним, что субполосы  $D_s$  и  $G_r$  определены выше.

Обозначив  $A_s$  и  $B_r$  множество значений (7), (8),

$$A_s = \{a_{in}^s\}, i, n = 1, \dots, N; B_r = \{b_{km}^r\}, k, m = 1, \dots, M, \quad (9)$$

совокупность соотношений (6) нетрудно представить в матричной форме

$$Y^{sr} = A_s \Phi B_r. \quad (10)$$

Отметим, что сумма субполосных компонент вида (10) при выполнении условий (4) на разбиение области определения исходной трансформанты Фурье в точности равна соответствующему изображению. Кроме того, выделение субполосных компонент можно использовать для фильтрации изображений и повышения их четкости [1].

### 3. МЕТОД АНАЛИЗА РАСПРЕДЕЛЕНИЯ ЭНЕРГИИ ИЗОБРАЖЕНИЙ ПО ЗАДАНЫМ ПОДОБЛАСТЯМ ПРОСТРАНСТВЕННЫХ ЧАСТОТ

Одним из основных фактов теории ортогональных представлений является равенство Парсевала, которое в случае трансформант Фурье имеет вид

$$\int_{-\pi}^{\pi} \int_{-\pi}^{\pi} |F^{\Phi}(x, y)|^2 dx dy / 4\pi^2 = \sum_{k=1}^M \sum_{i=1}^N f_{ik}^2. \quad (11)$$

В контексте субполосных анализа и синтеза это равенство целесообразно представить в виде

$$\|f\|^2 = \sum_{k=1}^M \sum_{i=1}^N f_{ik}^2 = \sum_{s=1}^S \sum_{r=1}^R P_{sr}(\Phi), \quad (12)$$

где

$$P_{sr}(\Phi) = \frac{1}{4\pi^2} \iint_{(x,y) \in V_{sr}} |F^{\Phi}(x, y)|^2 dx dy. \quad (13)$$

Представляется естественным интегралы вида (13) называть частями энергии (евклидовой нормы) изображения, попадающей в соответствующие ПОПЧ вида (4).

Доказано следующее утверждение. Точные значения частей энергий изображения вида (13) определяются соотношением

$$P_{sr}(f) = \text{tr}(A_s \Phi B_r \Phi^T), \quad (14)$$

где символ  $\text{tr}$  означает след матрицы, а  $T$  – транспонирование.

Справедливость этого утверждения доказывается непосредственной подстановкой в определение (13) представления (2).

Наиболее существенное значение этого результата заключается в том, что он позволяет определить подобласти пространственных частот, в которых сосредоточена подавляющая доля энергии изображения (выявление информационных подобластей ПЧ).

### 4. ОПТИМАЛЬНОЕ ВЫДЕЛЕНИЕ СУБПОЛОСНЫХ КОМПОНЕНТ ИЗОБРАЖЕНИЯ ИЗ ЗАДАНЫХ ПОДОБЛАСТЕЙ ПРОСТРАНСТВЕННЫХ ЧАСТОТ

Рассмотрим задачу разделения изображения на аддитивные компоненты

$$\Phi = Y + \Phi_0, \quad (15)$$

так, что первая из них  $Y = (y_{ik})$ ,  $i = 1, 2, \dots, M$ ,  $k = 1, 2, \dots, N$ , должна иметь трансформанту Фурье

$$F^Y(u, v) = \sum_{i=1}^{M_1} \sum_{k=1}^N y_{ik} e^{-ju(i-1)} e^{-jv(k-1)}, \quad -\pi \leq u, v < \pi, \quad (16)$$

которая в «идеальном» случае должна удовлетворять условию

$$F^Y(u, v) = F^{\Phi}(u, v), \quad (u, v) \in V_{sr}, \quad (17)$$

$$F^Y(u, v) = 0, \quad (u, v) \notin V_{sr}, \quad (18)$$

где  $V_{sr}$  – заданная ПОПЧ вида (4).

Поскольку отрезки конечной длины не могут иметь трансформанты Фурье с финитными областями определения, то удовлетворить условиям (17) и (18) «идеальной» фильтрации в области ПЧ не представляется возможным. Вместе с тем можно предложить вариант оптимального решения сформулированной задачи разделения изображения на аддитивные составляющие (выделение субполосных компонент), когда в некотором смысле наилучшим образом эти условия аппроксимируются. Ясно, что в качестве критерия целесообразно использовать меру погрешностей приближения к идеальному случаю. Очевидно, что такой подход соответствует постановке задачи оптимальной фильтрации изображений в соответствии с выбранной ПОПЧ.

В качестве такого критерия предлагается использовать функционал следующего вида

$$W_{sr}(Y, \gamma) = (1-\gamma) \iint_{(u,v) \in V_{sr}} |F^{\Phi}(u, v) - F^Y(u, v)|^2 dudv / 4\pi^2 + \gamma \iint_{(u,v) \notin V_{sr}} |F^Y(u, v)|^2 dudv / 4\pi^2, \quad (19)$$

где  $\gamma$  определяет весомость соответствующих интегралов, причем предполагается выполнение неравенства

$$0 < \gamma < 1. \quad (20)$$

Естественным принципом оптимизации является следующее вариационное условие

$$W_{sr}(Y, \gamma) = \min, \quad (21)$$

где минимум ищется по всем изображениям размерности  $(M \times N)$  в предположении, что параметр  $\gamma$  является постоянной величиной.

Доказано следующее утверждение. При фиксированном значении параметра  $\gamma$  с выполнением условия (20) решением вариационной задачи (21) является изображение, компоненты которого удовлетворяют системе линейных уравнений

$$y_{lp} = \gamma^{-1}(1-2\gamma) \sum_{m=1}^M \sum_{n=1}^N y_{mn} a_{ml}^s b_{pn}^r + \sum_{m=1}^M \sum_{n=1}^N f_{mn} a_{ml}^s b_{pn}^r, \quad (22)$$

$$l=1, \dots, M, p=1, \dots, N.$$

Очевидно, что для решения системы уравнений (22) можно применять итерационные методы. В частности в методе простых итераций начальному приближению вида

$$y_{mn}^0 = 0, \quad m=1, \dots, M, \quad n=1, \dots, N,$$

соответствует следующее приближение

$$y_{lp}^1 = \sum_{m=1}^M \sum_{n=1}^N f_{mn} a_{ml}^s b_{pn}^r, \quad l=1, \dots, M, p=1, \dots, N. \quad (23)$$

Совокупность последних соотношений нетрудно представить в матричной форме

$$Y^1 = A_s \Phi B_r, \quad (24)$$

которая, очевидно, совпадает с (10). Иными словами компонента вида (10) может быть использована в качестве начального приближения.

С другой стороны, при следующем значении параметра

$$\gamma = 0.5, \quad (25)$$

что соответствует выбору в (19) равных весовостей компонент меры погрешности искомой аппроксимации, система уравнений (22) снова преобразуется к виду (23). Следовательно, представление (10) определяет аддитивную частотную компоненту, оптимальную в смысле выполнения вариационного условия (21) при равной важности величины меры отклонения искомой трансформанты Фурье от нуля вне заданной ПОПЧ и её отклонения от трансформанты Фурье исходного изображения в этой ПОПЧ.

Возникает естественный вопрос выбора метода решения системы уравнений (22), матричная запись которой имеет вид

$$\gamma Y + (1-2\gamma) A_s Y B_r = (1-\gamma) A_s \Phi B_r. \quad (26)$$

Если ввести векторы  $\bar{w} = (w_1, \dots, w_{M \times N})^T$ ,  $\bar{v} = (v_1, \dots, v_{M \times N})^T$ , компоненты которых формируются на основе строк матриц  $Y$  и  $\Phi$  следующим образом

$$w_{(i-1) \times M + k} = y_{ik}, \quad v_{(i-1) \times M + k} = f_{ik}, \quad i=1, \dots, M, k=1, \dots, N, \quad (27)$$

и блочную матрицу

$$C^{sr} = \left\{ C_{ik}^{sr} \right\}, \quad i, k=1, \dots, M, \quad (28)$$

с блоками вида

$$C_{ik}^{sr} = a_{ik}^s B_r, \quad i, k=1, \dots, M,$$

то уравнение (26) можно преобразовать к системе линейных алгебраических уравнений

$$(\mathcal{A} + (1-2\gamma)C^{sr})\bar{w} = (1-\gamma)C^{sr}\bar{v}, \quad I = \text{diag}(1, \dots, 1). \quad (29)$$

Здесь  $I = \text{diag}(1, \dots, 1)$  – единичная матрица соответствующей размерности.

Отметим, что введенная матрица  $C^{sr}$  является кронекеровским произведением субполосных матриц  $A_s$  и  $B_r$ , обозначаемым в виде

$$C^{sr} = A_s \otimes B_r.$$

Показано, что матрица в левой части уравнения (29) является неособенной. Поэтому искомое решение имеет вид

$$\bar{w} = (1-\gamma)(\mathcal{A} + (1-2\gamma)C^{sr})^{-1} C^{sr}\bar{v}. \quad (30)$$

## 5. ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ

Одним из свойств субполосных компонент вида (10) является отсутствие влияния на данные компоненты энергии исходного изображения из смежных подобластей пространственных частот. Для демонстрации этого свойства были проведены эксперименты по сравнению абсолютных значений трансформант Фурье изображения, вычисленных его субполосных компонент и результатов КИХ-фильтрации. Так, на рис. 2 представлено модельное изображение (модель полосовых помех) размерности 512x512 пикселей и абсолютные значения фрагмента его трансформанты Фурье в области пространственных частот  $D$ ,

$$D = \{(x = 0.5\pi) \cap (y \in [0.12\pi, 0.18\pi])\}, \quad (31)$$

вне которой данные значения практически равны нулю.

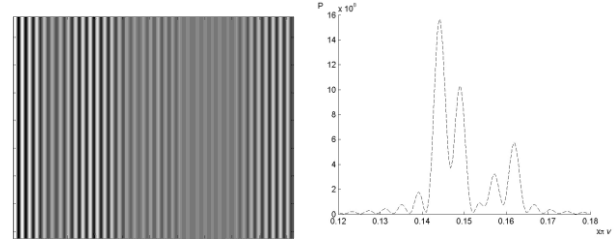


Рис. 2. Модельное изображение и поперечное сечение множества абсолютных значений его трансформанты Фурье

Для выбранного изображения были вычислены субполосные компоненты (10), соответствующие заданным подобластям пространственных частот  $V_1$  и  $V_2$ ,

$$V_1 = \{(x \in [0, \pi]) \cap (y \in [0.14\pi, 0.147\pi])\}, \quad (32)$$

$$V_2 = \{(x \in [0, \pi]) \cap (y \in [0.146\pi, 0.153\pi])\}.$$

Для полученных субполосных компонент были вычислены трансформанты Фурье, поперечное сечение множества абсолютных значений которых, соответствующие области  $D$  (31), отображены на рис. 3 с помощью сплошной линии. На рис. 3 пунктирной линией также отображены соответствующие абсолютные значения трансформант Фурье исходного изображения. Подобласти фильтрации  $V_1$  и  $V_2$  (32) на рис. 3а и рис. 3б отмечены вертикальными линиями. На графиках (рис. 3) видно, что спектр субполосных компонент монотонно убывает за границами подобластей фильтрации.



С целью сравнительного анализа были вычислены результаты КИХ-фильтрации [1] исходного изображения в подобластях  $V_1$  и  $V_2$  (длина импульсной характеристики выбранных КИХ-фильтров составляет 512 значений). Поперечное сечение множества абсолютных значений трансформант Фурье выходных последовательностей КИХ-фильтров (значений, соответствующих области  $D$  (31)) отображены на рис. 3 с помощью «точечной» линии.

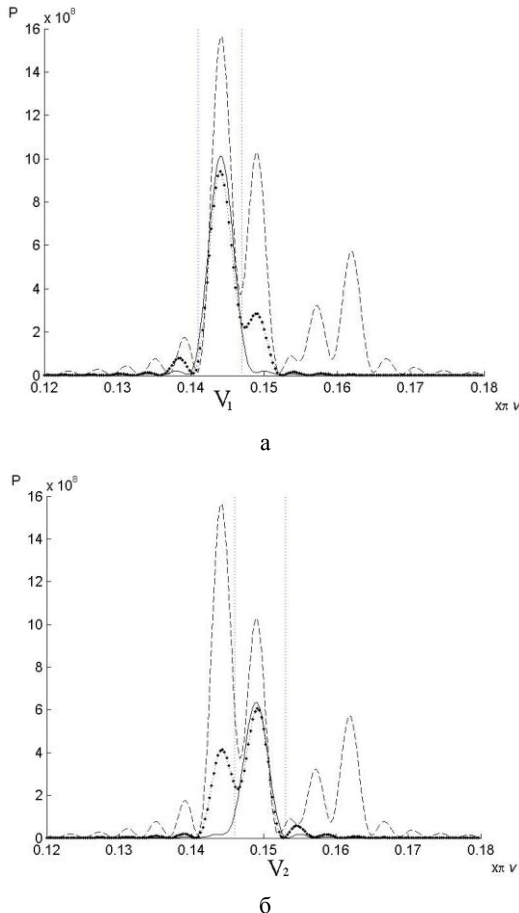


Рис. 3. Абсолютные значения трансформант Фурье в области  $D$  при фильтрации в различных подобластях ПЧ :  
а – в подобласти  $V_1$ , б – в подобласти  $V_2$

На рис. 3 видно, что на графиках спектра выходной последовательности КИХ-фильтра присутствуют значительные подъемы справа (рис. 3а) и слева (рис. 3б) от подобластей фильтрации  $V_1$  и  $V_2$ , определяемые компонентами исходного изображения, которые соответствуют подъемам на графике спектра исходного изображения в смежных подобластях пространственных частот.

Графики, приведенные на рис. 3, демонстрируют тот факт, что, в отличие от КИХ-фильтрации, на свойства субполосных компонент вида (10) не влияет энергия исходного изображения из смежных подобластей пространственных частот.

## 6. ВЫВОДЫ

В настоящей работе разработаны методы вычисления точных значений долей энергии изображения в заданных подобластях пространственных частот и оптимального выделения соответствующих этим подобластям субполосных компонент изображения (оптимального в смысле минимального отклонения трансформант Фурье субполосных компонент от трансформанты Фурье исходного изображения в заданной подобласти пространственных частот и минимального отклонения от нуля вне ее), для реализации которых не требуется вычисление трансформант Фурье. Также показано, что выделяемые субполосные компоненты полностью определяются двумерными отрезками трансформант Фурье исходного изображения из заданных подобластей пространственных частот.

## 7. ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ

Исследование выполнено при поддержке Министерства образования и науки Российской Федерации, гос. контракт № 14.514.11.4088.

## 8. ЛИТЕРАТУРА

- [1] Гонсалес, Р. *Цифровая обработка изображений [Текст] / Р. Гонсалес, Р. Вудс. – М.: Техносфера, 2012. – 1104 с.*
- [2] Жилияков, Е.Г. *Вариационные методы анализа сигналов на основе частотных представлений [Текст] / Е.Г. Жилияков, С.П. Белов, А.А. Черноморец // Вопросы радиоэлектроники. Сер. ЭВТ. – 2010. – Вып. 1. – С. 10-25.*
- [3] Жилияков, Е.Г. *О наилучшем ортогональном базисе для субполосного анализа и синтеза сигналов / Е.Г. Жилияков, С.П. Белов, С.В. Туяков, Д.В. Урсол // Информационные системы и технологии. – 2011. – № 2. – С. 26-33.*
- [4] Жилияков, Е.Г. *Оценка количества значимых долей энергии изображения, определяемых его проекциями на собственные векторы субполосных матриц [Текст] / Е.Г. Жилияков, А.А. Черноморец // Информационные системы и технологии. – № 3 (71). – 2012. – С. 51-58.*
- [5] *Методы компьютерной обработки изображений [Текст] / Под редакцией В.А. Соифера. – М.: ФИЗМАТЛИТ, 2003. – 784 с.*

## Информация об авторах

Жилияков Евгений Георгиевич, Белгородский государственный национальный исследовательский университет, г. Белгород, д. т. н., профессор, заведующий кафедрой информационно-телекоммуникационных систем и технологий, e-mail: zhilyakov@bsu.edu.ru

Черноморец Андрей Алексеевич, Белгородский государственный национальный исследовательский университет, г. Белгород, к. т. н., доцент, заведующий кафедрой прикладной информатики, e-mail: chernomorets@bsu.edu.ru

# Создание универсальных классификаторов текстовых образов на основе сверточных нейросетевых технологий

Кузьмицкий Николай Николаевич  
Кафедра "ЭВМ и системы"

Брестский государственный технический университет, Брест, Беларусь  
knnbrest@yandex.ru

## Аннотация

В данной работе предложен перспективный подход к построению универсальных классификаторов текстовых образов с использованием сверточных нейросетей (CNN). В его основе формирование комитетов CNN, обученных на образах различного масштаба, с последующей селекцией членов. Результаты проведенных экспериментов показали, что созданные классификаторы более эффективны, чем коммерческие системы на рассмотренных базах. При этом была достигнута уникальная точность распознавания тестового MNIST (заглавного NIST): 99.65% (98.174%), что является 3-м (1-м) из лучших опубликованных результатов.

**Ключевые слова:** CNN, комитет, селекция, универсальность.

## 1. ВВЕДЕНИЕ

Технологиям распознавания текстовых образов (OCR) уделяется повышенное внимание с начала развития области машинного зрения. Причиной тому является множество сфер их потенциального применения (практическая мотивация), а задача распознавания – одна из классических в контексте ИИ (творческая). В результате проведенных исследований:

- 1) накоплен большой объем информации в форме моделей классификаторов, методов выделения признаков, баз и др. [6].
- 2) созданы прикладные программные продукты, например, системы обработки банковских документов.
- 3) проведены эксперименты, показавшие возможность достижения точности распознавания, сравнимой с человеческой на отдельных тестовых множествах [2].

Однако, несмотря на большой прогресс в решении OCR задач, по-прежнему не решенной остается главная – создание универсального классификатора текстовых образов. Помимо объективных причин (их высокой вариативности) не менее важное значение имеет и ориентация разработчиков в первую очередь на специализированные решения в рамках предполагаемых ограничений: шрифтовой, неискаженный, рукопечатный текст и др. При этом во многих приложениях данный подход не позволяет достичь приемлемой точности: начиная от обработки рукописных документов, заканчивая естественными сценами и робототехникой.

Анализ литературы выявил значительный недостаток исследований, связанных с решением главной OCR задачи. Большая часть из рассмотренных работ посвящена созданию методов, способных определять тип текстовых блоков (чаще строк или слов), с целью последующего выбора подходящего классификатора. В частности, авторы [14] применяют подход на основе "мешка слов", SIFT-детекторов и SVM, достигая

хороших результатов разделения. Однако, используемые ими типы текстовых образов (машинописный и рукописный) охватывают лишь часть всех возможных, что, как показано в данной работе, может привести к падению точности распознавания на "смешанном" типе. Анализ распределения 822714 образов цифр, проведенный в [12], показал существование устойчивых границ классов, каждый из которых образует несколько крупных центров притяжения (кластеров). Вместе с тем, поиск данных границ затруднен существованием выбросов и аллографов – образов близких сразу нескольким классам. В [10] на примере рукописных баз (MNIST, USPS, DIGITS) и различных методик выделения признаков показана уязвимость классификаторов на основе моделей k-NN, SVM и частично нейросетях, в их способности переносить знания с тренировочного множества на другие. Авторы пришли к выводу, что данная проблема, названная "хрупкостью" (слабостью в ИИ терминологии), присуща всем моделям, основанным на методах машинного обучения.

В представленной работе описан подход к созданию универсальных классификаторов текстовых образов на базе сверточных нейронных сетей (CNN) и их комитетов. При этом основное внимание уделено изучению способности CNN распознавать образы различных типов и эффективному объединению знаний CNN. В ходе проведения экспериментов использовались как известные базы данных, так и новые.

## 2. ОСНОВНЫЕ МОДЕЛИ И ДАННЫЕ

### 2.1 Постановка задачи

Под универсальностью классификатора будем понимать его способность сохранять высокую точность распознавания на произвольной выборке входных данных. В зависимости от способа получения нами были выделены три типа текстовых образов: *машинописные, рукописные и синтезированные*, примеры которых приведены на рис. 1. К последнему типу относятся изображения, полученные в результате применения серий пространственных преобразований к образам первых двух. Так, в описываемом исследовании использовалась техника генерации, основанная на "волновых искажениях", образованных суперпозицией функций вида:

$$f_i(x) = (-1)^i \times a_i \times \cos(\pi / l_i \times x)$$

где  $a_i$  – амплитуда,  $l_i$  – длина  $i$ -ой функции. Подобные искажения позволяют имитировать эффекты, возникающие при неблагоприятных окружающих условиях съемки и дефектах аппаратуры (размытии, боковом ракурсе и др.).

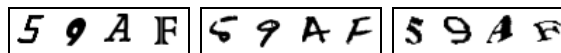


Рисунок 1: Слева направо: машинописные, рукописные и синтезированные текстовые образы.

Представленное исследование было выполнено для модели CNN, которая доказала свою эффективность в решении различных задач машинного зрения. CNN использовались для классификации двух подмножеств текстовых образов: арабских цифр и заглавных букв английского языка, ввиду их большой распространенности в практических приложениях. Оценка универсальности характеристик проводилась по перекрестному распознаванию образов выделенных типов.

## 2.2 Сверточная нейросетевая модель

В качестве базовой для построения сверточной модели была выбрана архитектура LeNet-5 [5], являющаяся наиболее разработанной в данном классе нейронных сетей. Данная архитектура способна выделять признаки образов в режиме "черного ящика" благодаря следующим особенностям:

1) *локальные рецептивные поля*: нейроны получают входной сигнал от окрестностей нейронов предыдущего слоя, за счет чего сеть обучается двумерной структуре входного образа;

2) *разделяемые веса*: нейроны слоя объединены картами, в которых они обладают общими весами, при этом карты генерируют различные признаки и сокращают количество параметров, настраиваемых в ходе обучения;

3) *пространственные подвыборки*: локальное усреднение откликов карт приводит к синтезу высокоуровневых признаков и повышает инвариантность сети к искажениям.

Отметим, что использованная архитектура, изображение которой приведено на рис. 2, отличалась от классической, т.к. перед выходным в ней отсутствовал RBF слой. Данный факт объясняется целесообразностью его применения при большем числе классов, например, для полного алфавита.

Обучение нейронных сетей проводилось модификацией алгоритма обратного распространения ошибки, основанной на стохастическом диагональном методе Левенберга-Марквардта [5], в online-режиме, с параметром обучения равномерно уменьшающимся от 0.001 до 0.000001 в течение 68 эпох. Перед каждой из них образы тренировочного множества подвергались эластичным ( $\sigma = 8$ ,  $\alpha = 36$ ) и аффинным искажениям (поворот на  $\pm 15^\circ$ , для образов '1', '7', 'Г' –  $\pm 7^\circ$ , и масштабирование в пределах  $\pm 15\%$ , для каждой размерности). Отметим, важность искажений в повышении обобщающих свойств сети и предотвращении переобучения.

## 2.3 Базы маркированных образов

Для обучения и тестирования CNN использовались базы, содержащие текстовые образы одного из типов:

1) машинописные: *FONT* – тренировочная часть включает по 1866 шрифтовых образов каждого класса (для цифр и заглавных букв) из [13] и по 508 образов с нормальным и полужирным начертанием из [1] (тестовая часть, содержит по 508 курсивных), *ETL6* – 13830 образов цифр и 38724 заглавных рукопечатных из [3];

2) рукописные: *MNIST* – 60000/10000 образов цифр в тренировочной/тестовой части из [8], *NIST* – 69522/11941 образов заглавных букв (разделены аналогично с [2]), *ETL1* – 14450 цифровых и 37570 заглавных образов из [3];

3) синтезированные: *KNI1* и *KNI2* – по 6000/1000 цифровых и 2500/500 заглавных образов каждого класса в тренировочной/тестовой части, созданных на базе шрифтовых с помощью методики генерации, основанной на волновых искажениях и поворотах с контролем попарного подобия.

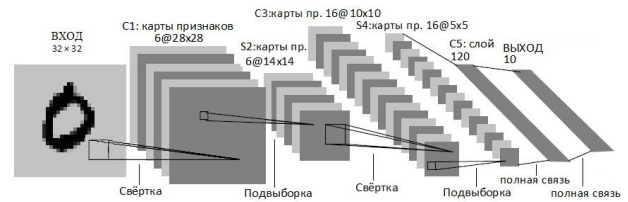


Рисунок 2: Использованная нейросетевая архитектура

Описанные базы по цели их использования можно разделить на две группы: *главные* (FONT, MNIST, NIST, KNI1) – применялись для построения нейронных сетей и анализа универсальности, *контрольные* (ETL, ETL6, KNI2) – для оценки обобщающих свойств итоговых классификаторов. Ввиду различия формата образов баз (размеров, диапазонов яркости), применялась их предобработка по схеме, аналогичной получению MNIST. Заметим также, что в дальнейшем изложении имя базы с префиксами *\_dig/\_big* и *\_test/\_train* трактуется как ее цифровая/заглавная и тренировочная/тестовая части соответственно.

## 3. ПРЕДЛАГАЕМЫЕ РЕШЕНИЯ

### 3.1 Построение "частных" CNN

Под "частной" будем понимать CNN, построенную на основе образов одного из выделенных типов. Так, для создания трех частных цифровых (заглавных) CNN использовались базы FONT, MNIST (NIST) и KNI1: обучение проводилось на их тренировочных, а проверка на тестовых частях. Результаты тестирования нейросетей, представленные в таблице 1, позволяют отметить следующее: 1) CNN имели более 99% точность распознавания образов своего типа (за исключением CNN\_NIST), что подтвердило эффективность выбранной архитектуры сети и методики ее обучения; 2) максимальный уровень ошибок был получен на множестве KNI1, что доказало актуальность рассмотрения синтезированного типа образов; 3) средняя точность сетей 92.58% (91.90%) показала низкую универсальность частных CNN.

### 3.2 Объединение "частных" CNN в комитеты

Имея высокоточные на образах своего типа CNN, перспективным являлся путь объединения их знаний с помощью методов коллективного распознавания. Хорошие предпосылки для формирования комитетов создавала общая архитектура и методика обучения CNN, обеспечивающие нормировку их откликов, как вероятностей классов, в единый диапазон. Одним из факторов эффективности комитета является схема голосования членов. Из существующих были выбраны наиболее универсальные, для которых победителем является класс: *максимальное (MAX)* – с максимальным откликом членов, *усредняющее (AVER)* – с максимальным средним откликом членов, *большинством (MAJOR)* – с наибольшим числом голосов членов в свою пользу.

CNN	MNIST	FONT dig	KNI1 dig	Среднее
CNN_MNIST	99.39	95.23	85.06	93.22
CNN_FONT dig	87.23	99.58	88.06	91.62
CNN_KNI1 dig	86.52	92.97	99.29	92.92
	NIST	FONT big	KNI1 big	
CNN_NIST	97.69	86.75	81.49	88.64
CNN_FONT big	89.41	99.15	89.27	92.61
CNN_KNI1 big	89.26	95.148	99.00	94.46

Таблица 1: Точность (в %) "частных" CNN.

Комитет	MNIST	FONT_dig	KN1_dig	Среднее
MAX_COM_dig	98.23	98.62	98.56	98.47
AVER_COM_dig	97.64	99.11	98.10	98.28
MAJOR_COM_dig	93.82	98.50	98.86	97.06
	NIST	FONT_big	KN1_big	
MAX_COM_big	96.62	98.27	98.21	97.70
AVER_COM_big	96.60	98.45	97.98	97.67
MAJOR_COM_big	95.09	97.26	95.43	95.92

Таблица 2: Точность (в %) комитетов "частных" CNN.

На основе ранее описанных CNN были сформированы три цифровых (заглавных) комитета с разными схемами учета голосов. Результаты тестирования комитетов, отраженные в таблице 2, показали, что схема MAX является наиболее эффективной. Это объясняется низким уровнем корреляции откликов частных CNN, ввиду отличия типа образов используемых при их обучении. Средняя точность MAX комитета превышала аналогичную одиночных CNN на 5.89% (5.80%), что доказало целесообразность объединения их знаний, однако уровень универсальности комитетов все еще оставался не достаточно высоким.

### 3.3 Формирование "частных" комитетов

Комитет позволил объединить знания одиночных CNN, однако их перекрестная точность все еще оставалась на низком уровне. С целью его повышения была исследована возможность формирования "частных" комитетов и их объединения в одном. Для создания комитетов применялась процедура "регулярного масштабирования": выберем диапазон вариации высоты  $h$  и ширины  $w$  образа (учитывая, исходный размер в  $20 \times 20$  использовались [16, 24] для  $h$  и [10, 18] –  $w$ , с шагом 2 пикселя) и выполним его масштабирование к каждому из допустимых размеров (всего возможны 25 комбинаций с фиксацией  $h$  и  $w$ , 10 – с фиксацией одного из них). Таким образом, для любого тренировочного/тестового множества можно построить 35 подмножеств аналогичной мощности, обучить 35 CNN и сформировать из них комитет, схема применения которого отражена на рис. 3.

Преимуществами описанной выше процедуры являются: 1) отсутствие необходимости привлекать дополнительные базы данных; 2) обеспечение высокого разнообразия образов; 3) соответствие естественной природе различия образов: шрифтов, наклонов и др. Кроме того, по сравнению с другими методиками, основная цель процедуры – повышение универсальности характеристик классификатора, а не его точности распознавания на отдельном множестве образов. Отметим также существенное значение при изменении  $h$  и  $w$  выход за пределы  $20 \times 20$  пикселей (в отличие от [2]). Это позволяет настраивать CNN-фильтры фиксированного размера  $5 \times 5$  на детектирование уникальных признаков, за счет варьирования степени приближения образа, наподобие "увеличительного стекла". Недостатком процедуры является значительное время обучения: для одной CNN от 3 до 5 часов на ЭВМ стандартной аппаратной конфигурации.

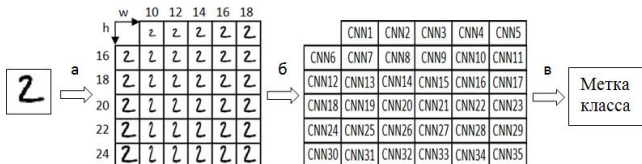


Рисунок 3: Схема применения "частного" комитета CNN: а) регулярное масштабирование; б) распознавание образов; в) объединение решений.

Комитет	MNIST	FONT_dig	KN1_dig	Среднее
COM_MNIST_dig	99.58	96.85	89.77	95.40
COM_FONT_dig	90.82	99.82	90.44	93.69
COM_KN1_dig	90.22	96.31	99.86	95.46
COM_COM_dig	98.95	99.27	99.06	99.09
	NIST	FONT_big	KN1_big	
COM_NIST	97.94	89.28	86.95	91.39
COM_COM_big	96.61	98.43	98.60	97.88

Таблица 3: Точность (в %) "частных" комитетов CNN.

Ввиду указанного недостатка в ходе экспериментов было создано меньше CNN, чем предусматривалось процедурой: построены по 10 сетей для баз MNIST, FONT\_dig, KN1\_dig и NIST, из которых были сформированы частные комитеты (COM), объединенные в общие (COM\_COM). Для частных комитетов использовалось AVER голосование (в связи с высокой корреляцией ошибок), для общих – MAX. Результаты тестирования комитетов, представленные в таблице 3, позволили отметить повышение перекрестной точности частных комитетов по сравнению с одиночными CNN на 2.18%, 2.07%, 2.54%, 2.75% для MNIST, FONT\_dig, KN1\_dig, NIST соответственно. При этом средняя точность общего цифрового (заглавного) комитета возросла на 0.62% (0.18%). Таким образом, регулярное масштабирование показало свою эффективность даже в своей сокращенной реализации, что доказывает перспективность процедуры.

### 3.4 Селекция членов комитетов

Основными факторами создания качественного комитета является точность и разнообразие (степень корреляции ошибок) членов [7]. Результаты экспериментов показали невысокий уровень разнообразия в частных комитетах, построенных хотя и с вариацией, но все же на одном обучающем множестве. Данный факт, наряду с необходимостью уменьшения размеров комитетов с целью их эффективной аппаратной реализации, привел к исследованию вопроса селекции членов, основной задачей которой является выбор подкомитета с точностью не ниже самого комитета.

Для этого использовались по 10 CNN, обученных ранее на базах MNIST и NIST с вариацией размеров: ( $h = 16, 18, 20, 22, 24, w/h$ ), ( $h = 20, w = 10, 12, 14, .$ ), ( $h = 18, w = 18$ ), ( $w = 22, w/h$ ) (где  $w/h =$  пропорционально), имеющие точности 99.14% – 99.44% и 97.39% – 97.79% на тестовых частях баз. Селекция CNN проводилась с помощью алгоритма EPIC [7], в основе которого лежит тезис: члены с большим (меньшим) числом верных (ложных) предсказаний в меньшинстве наиболее (наименее) полезны (вредны) при формировании комитета. Выходом алгоритма является перечень членов в порядке убывания их вкладов в общую эффективность, который трактовался двумя способами: 1) путем пошагового добавления членов строилось десять подкомитетов и выбирался самый точный; 2) подкомитет инициализировался первым членом, добавление остальных проводилось пока мог быть найден член, повышающий общую точность. Алгоритмом были сформированы по два подкомитета (из 5 и 6 членов) с точностью распознавания: тестового MNIST – 99.59% и 99.6%, NIST – 98.02% и 98.08%, превышающей уровень всего комитета, при этом количество членов, по сравнению с ним, было значительно уменьшено.

В базовой реализации EPIC корректность предсказания членами оценивается бинарными величинами, поэтому возникло предположение об их замене вероятностными откликами CNN. Так, модифицированный EPIC позволил

сформировать комитеты с 99.65% точного распознавания тестового MNIST и 98.174% заглавного NIST! Данные результаты являются 3-м и 1-м из лучших для данных точек отсчета, при этом по сравнению с ближайшими из [8] и [2], использовались гораздо менее громоздкие архитектуры CNN.

Отметим также, что комитеты в [2] содержали по 35 членов, в отличие созданных: MNIST – 4 члена ( $h = 16, 24, w / h$ ), ( $h = 20, w = 10, 18$ ), NIST – 6 ( $h = 16, 20, 24 w / h$ ), ( $h = 20, w = 14$ ) ( $w = 22, w/h$ ), ( $h = 18, w = 18$ ). Выбор данных сетей выглядел неоднозначным с позиции точности (10, 1, 9, 6-ая для MNIST) однако с точки зрения разнообразия он не являлся случайным, т.к. данные CNN были созданы при значительно отличных вариациях масштабов символов.

### 3.5 Формирование комитетов "общих" CNN

Кратким итогом описанных выше исследований является: "частные" CNN обладают низкой универсальностью, комитет является хорошим средством объединения их знаний, его размер ограничен практической эффективностью, поэтому необходима селекция членов. "Частные" CNN – наиболее слабое звено исследования, поэтому в его заключительной части был выполнен переход к "общим" CNN. Их обучение проводилось на смешенных выборках исходных баз. В частности, для построения цифрового (заглавного) множества MKF (NKF) использовалось по 20000/5000 (20540/5200) образов тренировочных/тестовых частей баз MNIST (NIST), FONT и KNI1 (количество образов обеспечивало равное представительство классов и их типов).

Результаты тестирования обученных CNN, приведенные в таблице 4, показали, что средняя точность "общих" CNN не только значительно превышала точность "частных", но и их комитетов: MAX\_COM\_dig на 0.65%, MAX\_COM\_big – 0.73% и даже COM\_COM\_dig на 0.03%. Причина данного факта заключается в том, что хотя базы MKF и NKF содержали меньше образов, однако различие их типов заставило CNN выделять более универсальные признаки, т.е. вместо объединения знаний выполнялась их интеграция. Эксперименты по схеме: регулярное масштабирование + селекция членов позволили построить комитеты, CNN в которых обучались на образах размеров: ( $h = 20, 16, 24, w/h$ ), ( $h = 20, w = 10, 18, 14$ ), ( $w = 20, w/h$ ). Их средняя точность, отраженная в таблице 4, была максимальной полученной. Для сравнения в таблице 4, 5 также приведены лучшие результаты систем [9, 11] (при разных значениях OCR/ICR параметров).

## 4. ЗАКЛЮЧЕНИЕ

Исследование показало низкую перекрестную точность CNN, одной из лучших моделей классификаторов, в распознавании текстовых образов с типом отличным от ее тренировочных.

Классификатор	MNIST	FONT_dig	KNI1_dig	Среднее
CNN_MKF	98.82	99.01	99.55	99.12
COM_MKF	99.15	99.56	99.86	99.52
SmartZone	97.65	95.07	89.04	93.92
Nicomsoft OCR	97.50	97.44	85.84	93.59
	NIST	FONT_big	KNI1_big	
CNN_NKF	97.17	98.10	98.07	97.78
COM_NKF	97.68	98.80	98.81	98.43
SmartZone	92.07	93.85	86.46	90.79
Nicomsoft OCR	85.55	97.76	85.26	89.52

Таблица 4: Точность (в %) "общих" CNN, их комитетов и сторонних классификаторов на главных базах.

Классификатор	ETL1	ETL6	KNI2_dig	Среднее
COM_MKF	99.05	99.78	99.25	99.36
SmartZone	96.90	99.43	93.65	96.66
Nicomsoft OCR	95.53	98.06	94.10	95.89
	ETL1	ETL6	KNI2_big	
COM_NKF	98.19	99.51	99.18	98.96
SmartZone	92.55	97.42	87.99	92.65
Nicomsoft OCR	92.69	95.97	89.43	92.69

Таблица 5: Точность (в %) комитетов "общих" CNN и сторонних классификаторов на контрольных базах.

Подход, основанный на объединении знаний "частных" CNN и их интеграции с помощью единой выборки типов в "общих" доказал свою эффективность в повышении универсальности. Методика создания комитетов CNN, обученных на разных масштабах образов, и их селекция позволили достичь уникальной точности распознавания тестового MNIST и заглавных букв NIST. Перспективными направлениями продолжения исследования является регуляризация множеств образов, развитие архитектуры CNN и их комитетов.

## 5. ССЫЛКИ

- [1] T.E. Campos, B.R. Babu *Character Recognition in Natural Images*. VISAPP (2), 2009, pp. 273-280.
- [2] D.C. Ciresan, U. Meier *Multi-column deep neural networks for image classification*. CVPR, 2012, pp. 3642-3649.
- [3] ETL character databases: <http://projects.itri.aist.go.jp/etlcnb/>.
- [4] P.J. Grother *Nist special database 19 – handprinted forms and characters database*. NIST, Tech. Rep., 1995.
- [5] Y. LeCun, L. Bottou *Gradient-Based Learning Applied to Document Recognition*. Proceedings of the IEEE, 86(11), 1998, pp. 2278-2324.
- [6] C.-L. Liu, H. Fujisawa *Classification and Learning Methods for Character Recognition: Advances and Remaining Problems*. Machine Learning in Document Analysis and Recognition, 2008, pp. 139-161.
- [7] Z. Lu, X. Wu *Ensemble pruning via individual contribution ordering*. KDD, 2010, pp. 871-880.
- [8] MNIST database: <http://yann.lecun.com/exdb/mnist/>.
- [9] Nicomsoft OCR: <http://www.nicomsoft.com/products/ocr/>.
- [10] A.K. Seewald *On the Brittleness of Handwritten Digit Recognition Models*. ISRN Machine Vision, vol. 2012, Article ID 834127, 10 pages.
- [11] SmartZone: <http://www.accusoft.com/smartzone.htm>.
- [12] S. Uchida, R. Ishida *Character Image Patterns as Big Data*. ICFHR, 2012, pp. 479-484.
- [13] J.J. Weinman, E. Learned-Miller *Scene text recognition using similarity and a lexicon with sparse belief propagation*. IEEE Trans. on PAMI, 31 (10), 2009, pp. 1733–1746.
- [14] K. Zagoris, I. Pratikakis *Handwritten and Machine Printed Text Separation in Document Images Using the Bag of Visual Words Paradigm*. ICFHR, 2012, pp. 103-108.

## Об авторах

Кузьмицкий Николай Николаевич – аспирант, кафедры "ЭВМ и системы" БрГТУ. Его адрес: [knnbrest@yandex.ru](mailto:knnbrest@yandex.ru)

# Оценка применимости алгоритмов сегментации изображения для решения задачи оконтуривания левого желудочка сердца на УЗИ-изображениях

<sup>1</sup>Анастасия Бобкова, <sup>1</sup>Сергей Поршневу, <sup>1</sup>Василий Зюзин, <sup>2</sup>Владимир Бобков  
<sup>1</sup>Институт радиоэлектроники и информационных технологий - РТФ

Уральский Федеральный Университет имени первого Президента России Б.Н. Ельцина, Екатеринбург, Россия  
[iconismo@gmail.com](mailto:iconismo@gmail.com), [sergey\\_porshnev@mail.ru](mailto:sergey_porshnev@mail.ru), [zvzuzin@gmail.com](mailto:zvzuzin@gmail.com)

<sup>2</sup>Уральский Государственный Экономический Университет – СИНХ, Екатеринбург, Россия  
[btow@yandex.ru](mailto:btow@yandex.ru)

## Аннотация

Проведено исследование работоспособности алгоритмов сегментации изображений при обработке УЗИ-изображений четырехкамерной проекции сердца в рамках задачи по автоматическому оконтуриванию левого желудочка. Результаты сравнения областей, получаемых с помощью алгоритмов сегментации и обозначенных экспертами, обнаруживают существенные расхождения между ними. Таким образом, задача автоматического оконтуривания левого желудочка сердца оказывается не тривиальной и требует разработки специализированных алгоритмов, учитывающих особенности эхокардиографических изображений.

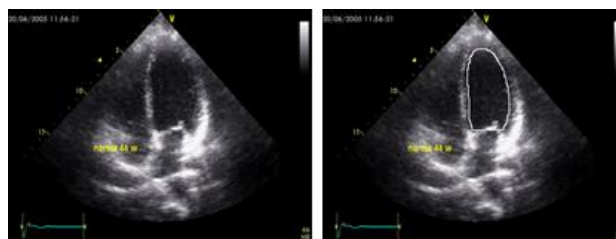
**Ключевые слова:** Левый желудочек, сегментация, поиск образов на изображении, эхокардиография (ЭхоКГ).

## 1. ВВЕДЕНИЕ

Большинство скоропостижных смертей людей, в том числе и не пожилого возраста, связано с различными болезнями сердца. В этой связи необходима качественная и точная диагностика, позволяющая предупреждать возникновение проблем сердечной деятельности задолго до их появления. Сегодня одним из наиболее распространенных методов диагностики являются ультразвуковые исследования (УЗИ) кровеносной системы (эхокардиография – ЭхоКГ), предоставляющие кардиологам визуальные изображения сердечной мышцы в различных ее проекциях. При этом наиболее часто в задачах кардиологии используется четырехкамерная позиция сердца (рис. 1), которая позволяет детально исследовать область левого желудочка (ЛЖ). На основании получаемой информации из контура ЛЖ кардиологи получают возможность рассчитывать различные количественные показатели, характеризующие работу сердечной мышцы и делать на их основе обоснованные выводы о состоянии сердечной мышцы пациента в целом.

Построение контура ЛЖ является одной из важных задач при диагностике сердечных заболеваний. На сегодняшний день контур ЛЖ строится либо вручную, либо полуавтоматически. Большинство современных УЗИ-сканеров имеют программное обеспечение, которое позволяет сделать автоматизировать процесс оконтуривания, но не делает его полностью автоматическим.

В тоже время анализ опыта врачей-практиков показывает, что при проведении контура врачи опираются на свои субъективные убеждения (доставляют недостающие границы областей сердца, сглаживают границу ЛЖ, отбрасывая на изображении, в соответствии со своими представлениями о правильной границе сердечной мышцы, те или иные области повышенной плотности), как правило, сформулировать которые в виде некоторого набора правил не удастся. Таким образом, возникает необходимость создания инструментария для построения контуров ЛЖ на основе строгих правил. Данная задача рассматривается в следующих работах: [1]; [6].



**Рис 1:** ЭхоКГ изображение (слева) и контур, обозначенный экспертом (справа).

В данной работе изложены результаты исследования возможности использования в рассматриваемой задаче известных алгоритмов сегментации изображений: JSEG [8], NCutImage [4], MNCS [7], Mean-Shift [3], Active Contours [5].

## 2. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

Для оценки были использованы данные ЭхоКГ, собранные в Свердловской областной клинической больнице № 1 г. Екатеринбурга в ходе УЗИ сердец пациентов. Оценка результатов оконтуривания проводилась по результатам, полученным у 30 пациентов, среди которых были и пациенты с патологией сердечной мышцы. УЗИ-фильмы для каждого пациента содержали в среднем по 25 кадров.

Типичные результаты применения различных алгоритмов сегментации к УЗИ-кадрам представлены на рис. 2–6, из которых видно, что наилучшие результаты с точки зрения близости формы ЛЖ, построенной экспертом и выделенной автоматически, оказываются алгоритмы JSEG и Active Contours, использованные в дальнейших исследованиях.

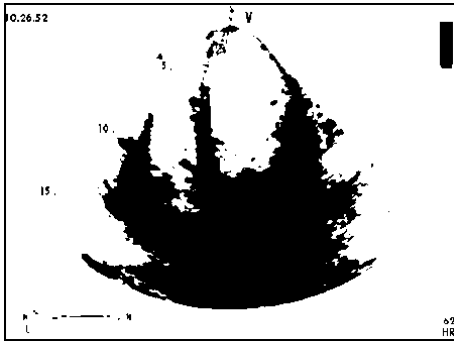


Рис. 2: Алгоритм сегментации Active Contours

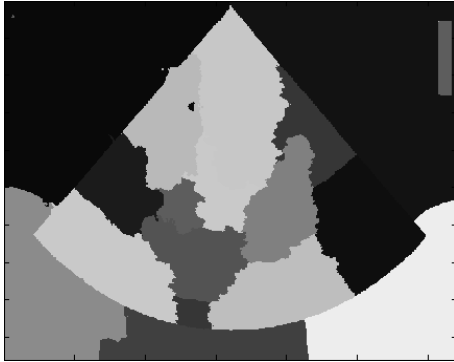


Рис. 3: Алгоритм сегментации MNCS

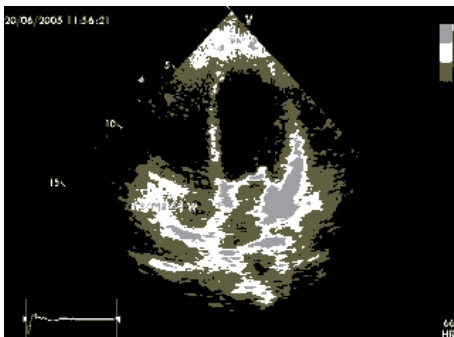


Рис. 4: Алгоритм сегментации Mean-Shift

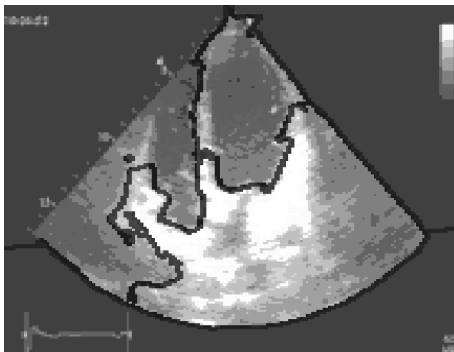


Рис. 5: Алгоритм сегментации NCutImage

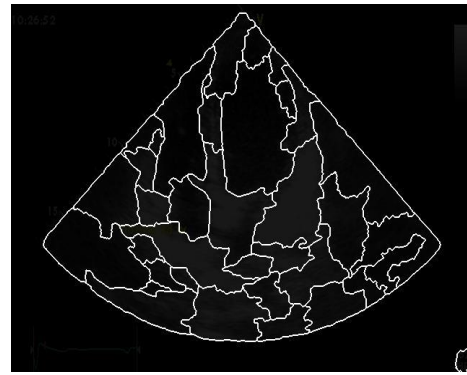


Рис. 6: Алгоритм сегментации JSEG

Для получения количественных показателей близости рассматриваемых контуров ЛЖ на сегментированных изображениях выбиралась область, в которую попал центр масс области экспертного контура. Затем найденная область сравнивалась с соответствующей областью контура, обозначенного экспертом. При оценке качества оконтуривания ЛЖ использовались следующие показатели:

- количество пикселей области, образуемой экспертным контуром ЛЖ;
- количество пикселей области, полученной путем сегментации изображения;
- количество пикселей области, образуемой пересечением двух областей.

Для оценки качества оконтуривания, следуя [2], были использованы следующие критерии:

- точность, отношение площадей правильно найденной области к экспертной области:

$$Precision = \frac{Expert \cap Segment}{Segment}$$

- полнота, отношение площадей правильно найденной области к области, полученной сегментацией:

$$Recall = \frac{Expert \cap Segment}{Expert}$$

Средние значения выбранных критериев представлены в таблице 1.

Таблица 1. Средние значения критериев оценки качества оконтуривания

Алгоритм сегментации	Precision (Точность)	Recall (Полнота)
JSEG	0.71 ± 0.15	0.76 ± 0.15
Active Contours	0.32 ± 0.16	0.80 ± 0.09

Из таблицы 1 видно, что ≈71% пикселей контуров, построенных алгоритмом JSEG, попадают в контур, построенный экспертом (точность), при этом данные пиксели составляют ≈76% от общего количества пикселей области, найденной алгоритмом JSEG (полнота): точность алгоритма Active Contours оказалась равной ≈32%, полнота – ≈80%.

### 3. ЗАКЛЮЧЕНИЕ

Полученные оценки точности нахождения контура ЛЖ на УЗИ-изображениях известными алгоритмами сегментации изображений не удовлетворяют врачей-кардиологов, поэтому для решения задачи автоматического оконтуривания ЛЖ сердца требуется разработка специализированных алгоритмов, адаптированных к низкому качеству УЗИ-изображений.

### 4. БЛАГОДАРНОСТИ

Работа выполнена при финансовой поддержке ФГБУ «Фонд содействия развитию малых форм предприятий в научно-технической сфере» в рамках госконтракта №11475п/20975.

### 5. ССЫЛКИ

- [1] Варламов А.Д., Макарова Е.Е.. Метод автоматического выделения объектов на УЗИ снимках сердца. Available at: <http://www.xn----7sbabbbc7aihkfne7cddf3ak8a.xn--p1ai/downloads/paper8.pdf>
- [2] Левашкина А.О., Поршнева С.В. Методы поиска изображений на основе визуального сходства // Монография, LAP LAMBERT Academic Publishing, 2011, 293 с.
- [3] Jens N. Kaftan, Andre A. Bell, Til Aach. Mean shift segmentation evaluation of optimization techniques. *In Proc. 3rd International Conference on Computer Vision Theory and Applications, VISAPP. 2008*, pp. 365–374.
- [4] Jianbo Shi. *Normalized Cut image segmentation*. Available at: <http://www.cis.upenn.edu/~jshi/>
- [5] Kristian Bredies, Heimo Wolinski. *An active-contour based algorithm for the automated segmentation of dense yeast populations on transmission microscopy images*. SFB-Report No. 2011–028. 2011.
- [6] Moldovanu S., Moraru L., Bibicu D. *Left ventricle segmentation using active contour model*. The Annals of «Dunărea de Jos» University of Galați, Fascicle V, Technologies in Machine Building, 2012
- [7] Timothee Cour, Jianbo Shi. *Multiscale Normalized Cuts Segmentation*. Available at: [http://www.timotheecour.com/software/ncut\\_multiscale/ncut\\_multiscale.html](http://www.timotheecour.com/software/ncut_multiscale/ncut_multiscale.html).
- [8] Yining Deng, B.S.Manjunath. *JSEG –Segmentation of color texture regions in images and video*. Available at: <http://vision.ece.ucsb.edu/segmentation/jseg/>

### Об авторах

Анастасия Олеговна Бобкова – к.т.н., доцент кафедры ИТ, ИРИТ-РТФ, УрФУ имени первого Президента России Б.Н. Ельцина.  
Её адрес: [iconismo@gmail.com](mailto:iconismo@gmail.com).

Сергей Владимирович Поршнева – д.т.н, профессор, заведующий кафедрой РЭИС, ИРИТ-РТФ, УрФУ имени первого Президента России Б.Н. Ельцина.  
Его адрес: [sergey\\_porshnev@mail.ru](mailto:sergey_porshnev@mail.ru).

Василий Викторович Зюзин – аспирант кафедры РЭИС, ИРИТ-РТФ, УрФУ имени первого Президента России Б.Н.

Ельцина.

Его адрес: [zvzuzin@gmail.com](mailto:zvzuzin@gmail.com)

Владимир Валерьевич Бобков – к.пед.н, доцент кафедры Бизнес-информатики, УрГЭУ – СИНХ.

Его адрес: [btow@yandex.ru](mailto:btow@yandex.ru).



# Двухфазная схема решения задачи классификации

Евгений Прохоров, Михаил Кумсков  
Механико-математический факультет

Московский государственный университет имени М.В. Ломоносова  
Москва, Россия

[eugeny.prokhorov@gmail.com](mailto:eugeny.prokhorov@gmail.com), [qsar\\_msu@mail.ru](mailto:qsar_msu@mail.ru)

## Аннотация

Данная работа посвящена решению задачи классификации в общем случае. Результаты получены в терминах математической теории распознавания образов. Предложенные методы решают задачу построения правил отказа от прогноза при просмотре больших баз объектов. Для предложенной в работе двухфазной схемы доказана оценка качества прогнозирования. В частности доказано улучшение качества прогноза при использовании нетривиальных правил отказа от прогноза. Также приводятся результаты практических испытаний предложенного метода решения задачи, подтверждающие эффективность описанного подхода. Полученные результаты могут быть использованы для улучшения качества классификации изображений с помощью распознающих моделей различной природы, а также для эффективного поиска нужных изображений в больших банках данных.

**Ключевые слова:** классификация, распознавание образов, машинное обучение

## 1. ВВЕДЕНИЕ

Работа посвящена в первую очередь задаче бинарной классификации, когда необходимо с использованием ранее обученной модели отнести объект классификации к одной из двух групп (обычно обладающих и не обладающих ключевым свойством объектов). Далее в тексте эти две группы условно обозначаются как «активные» и «неактивные» объекты. Однако полученные результаты легко обобщаются и на задачу с произвольным конечным числом классов. Ключевой особенностью решаемой задачи распознавания является предположение о том, что построенные модели будут использоваться для поиска объектов, потенциально обладающих ключевым свойством, в больших базах данных. Примерами таких задач могут выступать виртуальный скрининг [7] (поиск потенциально активных химических веществ в больших базах соединений) или поиск изображений, сходных с заданным шаблоном.

С учетом этой особенности рассматриваются ограничения допустимости для распознающих моделей [3, 9]. Такой подход к решению задачи классификации может быть рассмотрен, как классификация с отказами [2] или как частный случай применения смесей экспертов для классификации [8].

## 2. ОПРЕДЕЛЕНИЯ И ПОСТАНОВКА ЗАДАЧИ

Пусть обучающая выборка  $LS$  состоит из  $N$  объектов  $x_i$ ,  $i = 1, \dots, N$ , каждому из которых поставлено в соответствие одно из значений: 1 или  $-1$  (1 соответствует условно активным объектам,  $-1$  – неактивным). Вектор, последовательно содержащий активности всех объектов обучающей выборки, обозначим  $y = (y_1, y_2, \dots, y_N)$ ,  $y_i \in \{-1, 1\}$ .

Пусть также построена распознающая модель, решающая исходную задачу классификации, т.е.  $RM_1(x_i) \in \{-1, 1\}$  для любых  $x_i \in LS$ . Назовем  $RM_1$  моделью первого уровня.

Напомним, что процедура скользящего контроля (leave-one-out cross-validation [4]) заключается в следующем: из обучающей выборки последовательно удаляется каждый объект, по оставшимся объектам строится распознающая модель, и с помощью этой модели прогнозируется активность удаленного объекта. В работе везде будет использован функционал качества моделей со скользящим контролем, равный отношению количества верных прогнозов к общему числу спрогнозированных объектов.

Обозначим через  $R_1$  – множество тех объектов обучающей выборки  $x_i$ , для которых полученные в ходе процедуры скользящего контроля значения активности совпадают с действительными:  $RM_1(x_i) = y_i$ , т.е. множество верно классифицированных моделью первого уровня объектов. Через  $W_1$  обозначим множество ошибочно классифицированных моделью первого уровня объектов:  $W_1 = \{x_i \in LS \mid RM_1(x_i) \neq y_i\}$ . Таким образом, функционал качества со скользящим контролем для модели первого уровня равен  $\varphi_1 = |R_1| / N$ .

Определим задачу классификации второго уровня. Всем объектам обучающей выборки, спрогнозированным верно моделью первого уровня (их  $|R_1|$ ), поставим в соответствие значение 1, а соединениям, спрогнозированным неверно (их  $|W_1|$ ), поставим в соответствие значение  $-1$ . Сформируем таким образом вектор  $\hat{y} = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_N)$ ,  $\hat{y}_i \in \{-1, 1\}$ :

$$\hat{y}_i = \begin{cases} 1, & \text{если } RM_1(x_i) = y_i; \\ -1, & \text{если } RM_1(x_i) \neq y_i, \end{cases} \quad i = 1, \dots, N.$$

Возникшую задачу классификации назовем задачей классификации второго уровня.

Пусть построена распознающая модель  $RM_2$ , решающая задачу классификации второго уровня, т.е.  $RM_2(x_i) \in \{-1, 1\}$  для любых  $x_i \in LS$ . Назовем  $RM_2$  моделью второго уровня.

Пусть в ходе процедуры скользящего контроля моделью второго уровня получено  $|R_2|$  верных прогнозов, где  $R_2 = \{x_i \in LS \mid RM_2(x_i) = \hat{y}_i\}$ . Тогда функционал качества модели второго уровня  $\varphi_2 = |R_2|/N$ .

Наконец, определим результирующую распознающую модель  $RM_0$ . Результирующая модель решает исходную задачу классификации, но в отличие от модели первого уровня результирующая модель обладает опцией отказа от прогноза. То есть  $RM_0(x_i) \in \{-1, 0, 1\} \quad \forall x_i \in LS$  и значение  $RM_0(x_i) = 0$  интерпретируется как отказ от прогноза активности объекта  $x_i$ .

Для  $x_i \in LS$

$$RM_0(x_i) = \begin{cases} 1, & \text{если } RM_2(x_i) = 1 \text{ и } RM_1(x_i) = 1; \\ -1, & \text{если } RM_2(x_i) = 1 \text{ и } RM_1(x_i) = -1; \\ 0, & \text{если } RM_2(x_i) = -1. \end{cases}$$

Таким образом, результирующая модель осуществляет отказ от прогноза тогда, когда модель второго уровня предсказывает, что модель первого уровня ошибается, и осуществляет прогноз активности моделью первого уровня в противном случае.

Как и ранее, обозначим через  $R_0 = \{x_i \in LS \mid RM_0(x_i) = y_i\}$  множество верно классифицированных результирующей моделью объектов. Пусть также через  $Reject$  обозначено количество отказов от прогноза. Тогда функционал качества результирующей модели  $\varphi_0 = |R_0|/(N - Reject)$ .

### 3. ОЦЕНКА КАЧЕСТВА РЕЗУЛЬТИРУЮЩЕЙ МОДЕЛИ

**Теорема.** Верна следующая оценка качества результирующей модели.

$$\varphi_0 = \frac{(\varphi_1 + \varphi_2)N - Reject}{2(N - Reject)}.$$

**Доказательство.** По определению  $\varphi_1 N = |R_1|$ , а  $\varphi_2 N = |R_2|$ . Кроме того  $R_0 = R_1 \cap R_2$ . Последнее можно доказать так: если  $x_i \in R_1 \setminus R_2$ , то модель первого уровня осуществляет верный прогноз, однако модель второго уровня (ошибаясь) возвращает значение  $-1$ , в силу чего  $RM_0(x_i) = 0$ , таким образом, происходит отказ от прогноза. Если же  $x_i \in R_2 \setminus R_1$ , то модель первого уровня ошибается, а

модель второго уровня снова возвращает  $-1$ , что опять означает отказ от прогноза. Когда  $x_i \notin R_1$  и  $x_i \notin R_2$ , модель первого уровня ошибается, в то время как модель второго уровня возвращает значение  $1$ , таким образом, осуществляется неверное прогнозирование.

Учитывая выше сказанное, заметим, что отказам от прогноза соответствуют множества  $R_2 \setminus R_1$  и  $R_1 \setminus R_2$ . Следовательно,

$$Reject = |R_1 \Delta R_2|.$$

Таким образом, числитель дроби в формулировке теоремы приобретает вид  $|R_1| + |R_2| - |R_1 \Delta R_2|$ . Далее,

$$|R_1| + |R_2| - |R_1 \Delta R_2| = 2|R_1 \cap R_2| = 2|R_0|,$$

и, сокращая дробь на 2, имеем выражение для функционала качества  $\varphi_0$ . **W**

**Следствие 1.** Пусть  $\varphi_{\min} = \min(\varphi_1, \varphi_2) > 1/2$ , тогда, если  $Reject > 0$ , то  $\varphi_0 > \varphi_{\min}$ .

**Доказательство.** Заметим, что  $Reject < N$ . Действительно, так как  $\varphi_1 > 1/2$  и  $\varphi_2 > 1/2$ , то  $R_1 \cap R_2 \neq \emptyset$  и существуют верные ответы результирующей модели.

Теперь по теореме имеем

$$\begin{aligned} \varphi_0 &= \frac{(\varphi_1 + \varphi_2)N - Reject}{2(N - Reject)}, \Rightarrow \varphi_0 \geq \frac{2\varphi_{\min}N - Reject}{2(N - Reject)} = \\ &= \frac{2\varphi_{\min}N - 2\varphi_{\min}Reject + 2\varphi_{\min}Reject - Reject}{2(N - Reject)} = \\ &= \frac{2\varphi_{\min}(N - Reject) + (2\varphi_{\min} - 1)Reject}{2(N - Reject)} = \\ &= \varphi_{\min} + \frac{(\varphi_{\min} - 1/2)Reject}{N - Reject} > \varphi_{\min}. \end{aligned}$$

**W**

**Следствие 2.** Если  $\varphi_2 \geq \varphi_1 > 1/2$ , то в случае  $Reject > 0$  имеем  $\varphi_0 > \varphi_1$ .

**Доказательство** тривиально вытекает из следствия 1. **W**

**Следствие 3.** Если  $\varphi_2 > \varphi_1 > 1/2$ , то  $\varphi_0 > \varphi_1$ .

**Доказательство** вытекает из доказательства следствия 1. **W**

Таким образом, доказано, что если модель первого уровня классифицировала объекты из обучающей выборки хотя бы чуть лучше, чем случайным образом, и качество модели второго уровня не хуже качества модели первого уровня, то при условии, что количество отказов от прогноза больше нуля, результирующая модель демонстрирует более высокое качество классификации на исходной задаче, чем модель первого уровня. Также, доказано улучшение качества результирующего прогноза в случае, когда качество модели второго уровня превосходит качество модели первого уровня.

## 4. ОПИСАНИЕ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ

Описанный в работе подход тестировался на практике для решения задачи «структура – свойство». Задача «структура – свойство» – актуальная задача приложения математической теории распознавания образов в химии. Построенные классифицирующие модели используются для прогнозирования активности химических соединений. Работа с изображениями также является актуальной областью приложения предложенных методов, однако, на текущий момент тестирование на изображениях не проводилось и является направлением будущей работы.

В таблице приведены результаты использования предложенной двухфазной схемы решения для прогнозирования активности ингибиторов фермента деления клеток (PARP) [1]. Обучающая выборка, предоставленная Высшим химическим колледжем РАН, состояла из 120 соединений. Экспериментальное измерение активности проводилось в Институте химической биологии и фундаментальной медицины СО РАН.

В качестве метода построения распознающих моделей первого и второго уровня выступал метод опорных векторов (Support Vector Machine) [6]. При его использовании применялось ядро многослойного перцептрона (Multilayer Perceptron kernel) [5]. Для задач классификации первого и второго уровня независимо применялся также эволюционный отбор признаков. В таблице столбцы  $D_1$  и  $D_2$  содержат количество признаков, отобранных для решения задач классификации первого и второго уровня соответственно. Строки таблицы соответствуют различным типам описания химической структуры соединений (различным пространствам признаков).

Признаки	$\varphi_1$	$\varphi_2$	Отказы	$\varphi_0$	$D_1$	$D_2$
1	0,942	1	7	1	2	1
2	0,892	0,883	1	0,891	5	1
3	0,908	0,933	13	0,972	3	3
4	0,942	0,925	9	0,982	4	1
5	0,883	0,892	9	0,919	1	1
6	0,875	0,875	8	0,911	3	1
7	0,908	0,925	8	0,946	2	1
8	0,967	0,925	9	0,982	4	1

**Таблица 1.** Качество прогноза моделей, построенных с помощью метода опорных векторов

На основе полученных результатов построен прогноз для соединений с неизвестной активностью. Для наиболее перспективных соединений проводилось экспериментальное оценивание ингибирующей концентрации, в результате чего было найдено как минимум одно активное соединение, являющееся ингибитором PARP. Таким образом, показана практическая применимость двухфазной схемы решения задачи классификации.

## 5. ЗАКЛЮЧЕНИЕ

В работе предложен новый подход к решению задачи классификации, учитывающий такую особенность некоторых прикладных задач классификации, как необходимость осуществлять классификацию только ограниченного числа объектов. Подход может быть применен для решения задач виртуального скрининга и для поиска изображений,

соответствующих заданному шаблону. Доказаны теоретические оценки качества прогнозирования при использовании описанных методов. Приведенные практические исследования показывают эффективность подхода. Тестирование подхода на больших базах изображений является направлением будущей работы.

## 6. БЛАГОДАРНОСТИ

Работа была выполнена при частичной поддержке грантов РФФИ №10-07-00694, №12-03-01036-а и №12-03-92420.

## 7. ССЫЛКИ

- [1] Amours D., Desnoyers S., Silva I. and Poirier G.G. Poly(ADP-ribosylation) reactions in the regulation of nuclear functions // *Biochem. J.* 1999. **342**, № 2. 249–268.
- [2] Herbei R., Wegkamp M. Classification with reject option // *Can. J. Statist.* 2006. **4**, №4. 709–721.
- [3] Prokhorov E.I., Ponomareva L.A., Permyakov E.A., Kumskov M.I. Fuzzy classification and fast rules for refusal in the QSAR problem // *Pattern Recogn. and Image Anal.* 2011. **21**, №3. 542–544.
- [4] Stone M. Cross-validated choice and assessment of statistical predictions. // *J. Roy. Statist. Soc.* 1974. **B**, №36. 111–147.
- [5] Thomas R., Karsten B. Multilayer Perceptron kernel // *Proc. 24th SIBGRAPI Conf. on Graphics, Patterns and Images. Maceio, Alagoas, Brazil*, 2011. 337–343.
- [6] Vapnik V.N. *The nature of statistical learning theory*. N. Y.; London: Springer, 1998.
- [7] Walters W.P., Stahl M.T., Murcko M.A. Virtual screening – an overview // *Drug Disc. Today* 1998. V. 3. P. 160-178.
- [8] Yuksel S.E., Wilson J.N., Gader P.D. Twenty years of mixture of experts. // *IEEE Trans. Neural Networks Learning Syst.* 2012. **23**, №8. 1177–1193.
- [9] Прохоров Е.И. Нейронные сети для построения ограничений допустимости в задаче «структура–свойство» // *Нейрокомпьютеры: разработка, применение*. 2012. №10. 46–56.

## Об авторах

Евгений Прохоров – аспирант кафедры вычислительной математики механико-математического факультета МГУ им. Ломоносова. Электронная почта: eugeny.prokhorov@gmail.com

Михаил Кумсков – доктор физико-математических наук, профессор механико-математического факультета МГУ им. Ломоносова. Электронная почта: qsar\_msu@mail.ru

## Исследование методов удаления спекл-шумов на ультразвуковых изображениях

<sup>1</sup>Анастасия Бобкова, <sup>1</sup>Сергей Поршневу, <sup>1</sup>Василий Зюзин, <sup>2</sup>Владимир Бобков  
<sup>1</sup>Институт радиоэлектроники и информационных технологий - РТФ

Уральский Федеральный Университет имени первого Президента России Б.Н. Ельцина, Екатеринбург, Россия  
[iconismo@gmail.com](mailto:iconismo@gmail.com), [sergey\\_porshnev@mail.ru](mailto:sergey_porshnev@mail.ru), [zvzuzin@gmail.com](mailto:zvzuzin@gmail.com)

<sup>2</sup>Уральский Государственный Экономический Университет – СИНХ, Екатеринбург, Россия  
[btow@yandex.ru](mailto:btow@yandex.ru)

### Аннотация

Выполнено экспериментальное исследование методов удаления спекл-шумов на ультразвуковых изображениях: SVD-преобразование, медианная фильтрация и фильтр Винера. Вне зависимости от параметров алгоритмов лучшие результаты показала медианная фильтрация. При размере окрестности области сглаживания  $8 \times 8$  пикселей медианная фильтрация показывает наилучшие результаты. Соответственно, данный метод целесообразно использовать на предварительном этапе для устранения спекл-шума на ультразвуковых изображениях.

**Ключевые слова:** Левый желудочек, спекл-шумы, поиск образов на изображениях, эхокардиография (ЭхоКГ).

### 1. ВВЕДЕНИЕ

Эхокардиография (ЭхоКГ) – метод исследования структуры и функции сердца, основанный на регистрации отражённых импульсных сигналов ультразвука, генерируемых датчиком с частотой 2,5 – 5,0 МГц. Зондирующая волна, если геометрические размеры исследуемого объекта превышают длину ультразвуковой волны (1 – 1,5 мм), отражается от границ раздела двух сред, имеющих различные акустические плотности, и рассеивается на объектах, геометрические размеры которых не превосходят 1 мм. Последние объекты оказываются слишком мелкими для их распознавания ультразвуковым методом и на снимках ЭхоКГ визуализируются в виде спекл-шума (speckle noise). Данный шум, представляет собой случайно распределенные черно-белые пиксели на всей поверхности ЭхоКГ-кадра.

Спекл-шум ухудшает качество изображения и делает его визуальное и цифровое распознавание достаточно сложной задачей. Как следствие, с точки зрения задач медицинской диагностики, снижается ценность ЭхоКГ-снимков и затрудняется внедрение алгоритмов автоматической диагностики. Таким образом, задача уменьшения влияния спекл-шума на качество изображения, решение которой позволит облегчить интерпретацию и анализ результатов ЭхоКГ исследований, является актуальной.

В настоящее время разработано множество методов удаления шумов: линейное усреднение пикселей по соседям [3], медианная фильтрация [1], математическая морфология [2], гауссовское размытие [2], методы на основе вейвлет-преобразований [2], фильтры Винера [2] и пр.

В работе [4] сделан обзор возможностей применения SVD-преобразования для обработки изображений. Сделаны выводы о том, что данное преобразование является перспективным и актуальным для исследования возможностей его применения в обработке изображений.

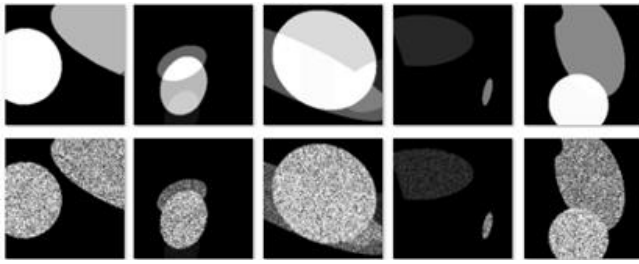
Целью данной статьи является исследованием методов устранения спекл-шумов на ЭхоКГ-изображениях и сравнение SVD-преобразования с известными методами устранения шумов на изображениях: медианной фильтрации и фильтром Винера.

### 2. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

Обозначим  $X^1$  – исходное изображение,  $Y$  – изображение со спекл-шумом. Спекл-шум является мультипликативным, т.е. не добавляется в требуемый сигнал (как в случае аддитивного шума), а умножается на него. Соответственно, связь между исходным и зашумлённым изображениями определяется уравнением  $Y = X^1 + n \cdot X^1$ , где  $n$  – равномерно распределённый случайный шум со средним значением 0 и дисперсией  $v$ . Для уменьшения влияния параметров шума на качество его удаления, дисперсия  $v$  выбиралась случайным образом из выборки со средним значением 0,5 и стандартным отклонением, равным 1.

При тестировании методов удаления шумов обычно проводится сравнение исходного изображения  $X^1$  с изображением, получаемом после удаления ранее наложенного шума  $X^2$ . Однако, при исследовании способов удаления шумов с ультразвуковых изображений, подобное сравнение является очень проблематичным, поскольку они изначально формируются уже с присутствующими в них спекл-шумами. В этом случае представляется возможным для тестирования использовать искусственно сгенерированные изображения, как, например, при анализе медицинских изображений часто используются так называемые фантомные изображения.

В данной работе тестирование проводилось на коллекции из 500 искусственно сгенерированных изображений размером  $200 \times 200$  пикселей (рис. 1, верхний ряд). На каждом изображении присутствуют области, параметры которых задаются случайным образом. Далее, на каждое исходное тестовое изображение накладывался шум со случайными параметрами (рис. 1, нижний ряд).



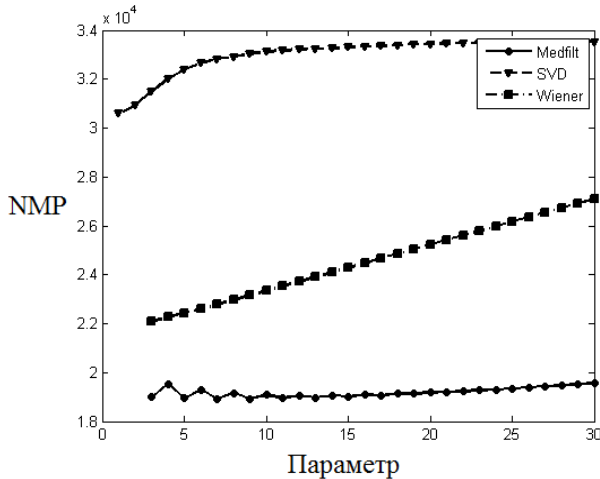
**Рис. 1:** Изображения тестовой коллекции: исходные (верхний ряд), со спекл- шумом (нижний ряд)

Для оценки качества удаления спекл-шума на изображениях тестовой коллекции использовались следующие критерии:

1. Количество несовпадающих пикселей NMP (*number of mismatched pixels*)  $NMP = \sum_{i=1}^M \sum_{j=1}^N Diff(i, j)$ , где  $Diff(i, j) = 0$ , если  $X^1(i, j) = X^2(i, j)$ .

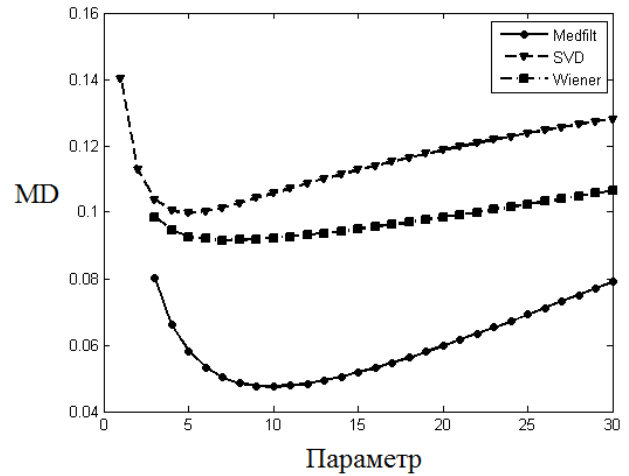
2. Плотность несовпадений MD (*mismatch density*)  $MD = \frac{1}{M \cdot N} \cdot \sum_{i=1}^M \sum_{j=1}^N |X^1(i, j) - X^2(i, j)|$ .

Зависимости выбранных критериев от значения параметра соответствующего алгоритма (для медианной фильтрации и фильтра Винера – размера области сглаживания  $k = k_x \times k_y$  пикселей; для SVD-преобразования – количества главных компонент, использованных при восстановлении изображения) представлены на рис. 2 и 3.



**Рис. 2:** Зависимость среднего количества несовпадающих пикселей от размера области сглаживания (фильтр Винера, медианная фильтрация) и числа главных компонент (метод SVD)

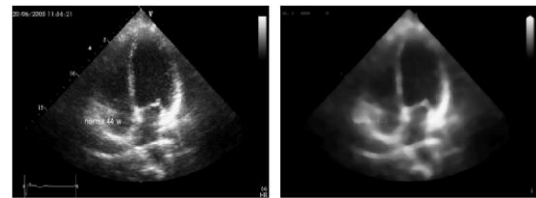
Из рис. 2 видно, что при использовании медианной фильтрации зависимость среднего количества несовпадающих пикселей от параметра алгоритма проявляется в наименьшей степени. При этом наименьшим оказывается и среднее значение несовпадающих пикселей.



**Рис. 3:** Зависимость средней плотности несовпадений от размера области сглаживания (фильтр Винера, медианная фильтрация) и числа главных компонент (метод SVD)

Из рис. 3 видно, что наименьшее значение плотности несовпадений при всех значениях параметров алгоритмов имеет метод медианной фильтрации, у которого изученная зависимость достигает своего минимального значения при размере области сглаживания  $k = 8 \times 8$  пикселей.

Полученные результаты были использованы для обработки ЭхоКГ-снимков. Типичный вид исходного УЗИ-кадра и этого же кадра после устранения спекл-шума медианной фильтрацией с  $k = 8 \times 8$  представлен на рис. 5.



**Рис. 4:** Исходное УЗИ-изображение (слева) и результат удаления шума медианным фильтром при  $k = 8 \times 8$  (справа)

Из рис. 4 видно, что выбранное преобразование, действительно, уменьшает спекл-шум, присутствующий в исходном изображении.

### 3. ЗАКЛЮЧЕНИЕ

Проведен сравнительный анализ возможности использования SVD-преобразования, медианной фильтрации и фильтра Винера в задаче для удаления спекл-шумов на ультразвуковых изображениях. Эксперименты показали, что при любом размере основного параметра алгоритмов (окрестности) наилучшим оказывается медианный фильтр. При этом, оптимальным значением параметра медианной фильтрации оказывается  $k = 8 \times 8$ .

### 4. БЛАГОДАРНОСТИ

Работа выполнена при финансовой поддержке ФГБУ «Фонд содействия развитию малых форм предприятий в научно-технической сфере» в рамках госконтракта №11475p/20975.

## 5. ССЫЛКИ

- [1] Егорова И. Н., Коваленко Е. К. Методика повышения качества изображений с использованием методов фильтрации шумов. Восточно-Европейский журнал передовых технологий, 2011, № 4/2 (52).
- [2] Гонсалес Р. Вудс Р. Цифровая обработка изображений. Москва, Техносфера, 2006. 1072 с.
- [3] Калинина Д., Ватолин Д. Проблема подавления шума на изображениях и видео и различные подходы к ее решению. Компьютерная графика и мультимедиа, 2005, №3(2). Доступно по ссылке: <http://cgm.computergraphics.ru/content/view/74>.
- [4] Sadek R. A. SVD Based Image Processing Applications: State of The Art, Contributions and Research Challenges. *International Journal of Advanced Computer Science and Applications*, 2012, vol. 3, No. 7, pp. 26-34.

### Об авторах

Анастасия Олеговна Бобкова – к.т.н., доцент кафедры ИТ, ИРИТ-РТФ, УрФУ имени первого Президента России Б.Н. Ельцина.

Её адрес: [iconismo@gmail.com](mailto:iconismo@gmail.com).

Сергей Владимирович Поршнеv – д.т.н, профессор, заведующий кафедрой РЭИС, ИРИТ-РТФ, УрФУ имени первого Президента России Б.Н. Ельцина.

Его адрес: [sergey\\_porshnev@mail.ru](mailto:sergey_porshnev@mail.ru).

Василий Викторович Зюзин – аспирант кафедры РЭИС, ИРИТ-РТФ, УрФУ имени первого Президента России Б.Н. Ельцина.

Его адрес: [zvzuzin@gmail.com](mailto:zvzuzin@gmail.com)

Владимир Валерьевич Бобков – к.пед.н, доцент кафедры Бизнес-информатики, УрГЭУ – СИНХ.

Его адрес: [btow@yandex.ru](mailto:btow@yandex.ru).

# Интерактивный выбор объектов трёхмерной сцены

Е.И. Коростелев, М.А. Городилов, Б.С. Долговесов  
Институт Автоматики и Электротехники СО РАН, Новосибирск, Россия  
kore3d@gmail.com

## Аннотация

В статье предлагается метод выбора трёхмерных объектов виртуальной сцены для систем визуализации реального времени. Метод использует один бит буфера трафарета и позволяет определять порядковый номер объекта, которому принадлежит некоторый пиксель итогового изображения. Таким образом, он позволяет определять объекты под курсором мыши, что позволяет интерактивно взаимодействовать с ними. Для этого не требуется дополнительных проходов визуализации, и отсутствуют ограничения на используемые при визуализации материалы (процедурные сетки, полупрозрачность, альфа-тест).

**Ключевые слова:** выбор объектов, буфер трафарета, визуализация реального времени.

## 1. ВВЕДЕНИЕ

Большинство систем виртуальной реальности предоставляют возможность интерактивного взаимодействия с рендеримыми виртуальными объектами. Как правило, взаимодействие осуществляется путём выбора конкретного объекта курсором мыши, а выбранный объект выделяется специальным образом. Кроме этого, в презентационных и обучающих системах [1] существует необходимость рисовать поверх объектов, для чего используется аналогичный механизм выбора.

В системах визуализации, где число доступных для выбора объектов незначительно, как правило, применяется метод выбора объектов путём нахождения ближайшего пересечения луча с набором доступных объектов [2]. Этот алгоритм реализуется на центральном процессоре и не позволяет точно выделять динамически изменяющиеся объекты (например, процедурно-генерируемые сетки).

Одним из методов определения порядкового номера (порядок визуализации) объекта под курсором является: отдельный проход рендеризации всех видимых объектов сцены с упрощёнными материалами и запись в буфер цвета (как правило, одноканальный 32-битный) идентификатора объекта [3]. Этот метод использует графический процессор, но имеет ряд недостатков. Во-первых, он требует повторную рендеризацию всех объектов (списка батчей), что накладно в случае динамически изменяющихся сеток (например, скелетная анимация). Во-вторых, в «простом» материале, используемой на отдельной стадии, необходимо учитывать свойства материалов, которые отбраковывают часть поверхности (например, инструкция discard в шейдере) или дают альфа-канал (альфа-тест).

Авторами статьи предлагается метод, лишенный изложенных выше недостатков и обладающий определёнными преимуществами: простота интеграции в существующую систему, производительность, возможность реализации для видеокарт уровня DirectX 9. Несущественными недостатками

предлагаемого метода является необходимость использовать 1 бит буфера трафарета [4] (stencil buffer), теста трафарета (stencil test) и дополнительной видеопамати (до 32 МБ на разрешение сверхвысокой чёткости 4K UHD TV).



Рис 1: Выбор одного из 1000 виртуальных объектов

Пример использования разработанного метода приводится на рисунке 1. На изображении представлен результат визуализации тысячи трёхмерных объектов (разноцветных сфер), поверхность которых определяется специальной текстурой (отбраковка части поверхности), а цвет задан случайно. Объект, находящийся под курсором, выделен специальным цветом.

## 2. АЛГОРИТМ

Основой метода является использование буфера и теста трафарета. Буфер трафарета используется для перезаписи 1 бита после визуализации каждого объекта (вызов отрисовки). Поскольку требуется перезаписывать только 1 бит, данный метод может быть скомбинирован с другими алгоритмами, использующими трафарет, либо применяться без ограничений. Большинство алгоритмов визуализации не используют тест трафарета и ограничиваются использованием буфера и теста глубины (буфера формата D24SX, где D24 – 24 бита для записи глубины). Таким образом, достаточно использовать буфер формата D24S8.

Дополнительно, для хранения идентификаторов рендеримых объектов, потребуется один буфер, соответствующий размеру кадра (обозначим его *PickBuffer*). Для него достаточно формата R16G16. В красный канал (R16) будет записываться идентификатор объекта.

При использовании предлагаемого метода визуализация всех объектов производится как обычно, без ограничений. Алгоритм поддерживает буферы с несколькими «семплами» и не влияет на поддержку алгоритмов устранения ступенчатости.

Рассмотрим общую схему работы алгоритма:

1. Начало кадра: инициализируем буфер трафарета значением 0.
2. Визуализация объектов сцены: включаем запись в буфер трафарета и пишем 1 бит, если пиксель прошёл тест глубины (либо тест выключен)
- 2.1 После каждого вызова отрисовки временно переключаем буфер кадра на *PickBuffer* и с включенным тестом трафарета на проверку бита с шага 1 рисуем прямоугольник в 1 пиксель (размер области, в которой необходимо определить ближайший объект). При этом, во время проверки сбрасываем этот бит.
3. Конец кадра: пересылаем содержимое *PickBuffer* в системную память.
4. Декодирование области *PickBuffer* (распаковка идентификатора) – получение номера объекта

На шаге 2.1 может быть выбрана область произвольного размера, но её существенное увеличение приведет к падению производительности, поскольку потребуются обновлять больше значений в видеопамяти. Каждая видеокарта имеет ограничение скорости заполнения пикселями (филлрейт).

Содержимое буфера идентификаторов (*PickBuffer*) для полного кадра тестовой сцены представлено на рисунке 2. На изображении для наглядности заполнены все пиксели, но, в случае использования однопиксельной области, в буфере обновляется только 1 пиксель. Кроме этого, для лучшего визуального различия содержимого буфера цвета были переназначены и использован красный цвет для подсветки текущего выбранного объекта.

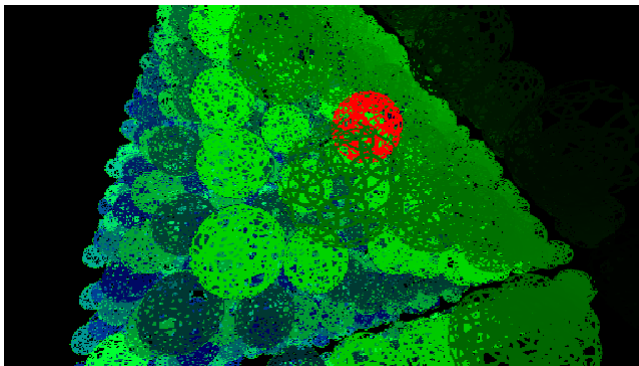


Рис 2: Буфер идентификаторов 3d-объектов полного кадра

При использовании мультисемплирования буфер идентификаторов содержит несколько субпикселей на пиксель. Перед передачей его в системную память производится преобразование субпикселей в пиксель. Данный механизм поддерживается аппаратно, но плохо конфигурируется и, при настройках по умолчанию, даёт усредненные значения по границам объектов, что нельзя интерпретировать как идентификаторы. На большинстве видеокарт 8-летней давности поддерживается возможность задать маску семплирования, что позволяет исключить преобразование значений. В случае отсутствия возможности маскировать какой-то один субпиксель, допускается использование области из 9 пикселей на шаге 2 и отбраковке значений, если в окрестности не найдено 2-3 совпадения.

Копирование *PickBuffer* в системную память (третий шаг алгоритма) следует выполнять с задержкой, как минимум, в 1 кадр, т.е. на текущем кадре считывать предыдущий. При

несоблюдении этого условия возникает точка синхронизации между графическим и центральным процессором, что отрицательно сказывается на производительности. Причина в том, что центральный и графический процессор работают асинхронно и, к моменту выполнения шага 3 центральным процессором, данные будут еще не готовы. Кроме этого, все команды графическому процессору буферизируются и могут задерживаться на несколько кадров.

Небольшое запаздывание выбора объекта под курсором мыши (из-за задержки считывания *PickBuffer*) устраняется путём прогнозирования перемещения курсора. Для этого используются значения ускорения и скорости движения мыши за несколько предыдущих кадров, а для текущего кадра берется прогнозируемая позиция.

### 3. РЕЗУЛЬТАТЫ

Реализация алгоритма тестировалась на различном аппаратном обеспечении. При этом типовым случаем считалось наличие большого набора объектов и использование алгоритмов устранения ступенчатости (мультисемплирование). Приводимые далее результаты были получены в реализации алгоритма на Direct3D11 и исполнение его на видеокarte с аппаратной поддержкой DirectX10 (функциональный профиль 10.0 устройства Direct3D11). Кроме этого, работоспособность алгоритма проверена в рамках функций Direct3D9 (и функционального профиля 9.3 устройства Direct3D11). Результаты замера производительности для реализации Direct3D11 приведены в таблице 1.

Видеокарта \ N	512	1000	2197
NVIDIA 8800 GTX	3 мс	4 мс	5 мс
AMD FirePro W8000	5 мс	13 мс	26 мс

Табл. 1: Производительность при выборе среди N объектов

Как видно, из таблицы наиболее эффективно метод работает на видеокартах NVIDIA. На результат тестирования при этом влияет, как версия драйвера, так и поколение видеочипа. Поскольку в большинстве случаев не требуется выбирать объект из десятков сотен других, производительности достаточно для решения большинства задач, где необходимо интерактивное взаимодействие с трёхмерными объектами.

Разработанный метод также позволяет выбирать объекты последовательно, т.е. выбирать объекты не прошедшие растеризацию или последовательно прятать ближайшие. Такой механизм удобен при необходимости выбирать объекты, находящиеся позади полупрозрачных. Данная функция реализуется посредством составления «чёрного» списка объектов, для которых игнорируется запись в буфер идентификаторов.

Возможность рисования поверх поверхности объекта после выбора некоторого объекта обеспечивается путём дополнительной растеризации этого объекта в конце кадра. Это позволяет определить текстурные координаты выбранной поверхности и реализуется с использованием уже заполненного на этапе подготовки кадра буфера глубины.



## 4. ЗАКЛЮЧЕНИЕ

В статье предложен эффективный алгоритм выбора объектов, использующий аппаратные возможности современных и старых поколений графических процессоров. Разработанная технология применяется в интерактивной системе визуализации реального времени, что позволяет предоставить пользователю возможность выбирать объекты, менять их свойства, а также рисовать на поверхности виртуальных объектов.

## 5. БИБЛИОГРАФИЯ

- [1] Долговесов Б.С. Системы виртуальной реальности для космических тренажерных комплексов и интерактивных презентаций // Сборник материалов конференции «Пилотируемые полеты в космос», ИМБП РАН, 2011. С. 66.
- [2] Brian Hook, Mouse Ray Picking Explained, <http://bookofhook.com>
- [3] D. Cantor et al., WebGL Beginner's Guide, 2012, pp.257-285
- [4] J. Zink et al. Practical Rendering and Computation With Direct3D 11, 2011, pp.241-258

# Interactive object picking by stencil buffer technique

## Abstract

A new approach for fast object picking is presented in the article. Described technique uses a stencil test. No additional rendering passes are needed. So this technique allow to select any meshes (procedural, alpha blended, etc) and works fine with multisample render targets. And also it's implementable for d3d9-capable hardware.

*Keywords: object picking, stencil buffer, real-time rendering.*

## About the author

Evgeny I. Korostelev is a Ph.D. student at Institute of Automation and Electrometry SB RAS. His contact email is [kore3d@gmail.com](mailto:kore3d@gmail.com)

Mikhail A. Gorodilov is a Ph.D. student at Institute of Automation and Electrometry SB RAS. His contact email is [gorodilovm@gmail.com](mailto:gorodilovm@gmail.com)

Boris S. Dolgovesov (Ph.D.) is a head of Synthesizing Visualization Systems Laboratory at Institute of Automation and Electrometry SB RAS. His contact email is [bsd@iae.nsk.su](mailto:bsd@iae.nsk.su)

# Система распознавания символов на изображениях со сложным фоном

Анна Балахонцева, Александр Годоба, Нгуен Тьен  
Институт компьютерных систем

Одесский национальный политехнический университет, Одесса, Украина  
a.c.balackhontseva@gmail.com

## Аннотация

Предложена система сегментации и распознавания символов на изображениях со сложным фоном. Алгоритм сегментации использует метод кратчайшего пути по яркости пикселей для расчета разрезов между символами. Распознавание символов основывается на алгоритме масштабного пространства кривизны (curvature scale space - CSS). Предлагается использовать вейвлеты Гаусса для расчета кривизны кривых в алгоритме CSS для упрощения разрабатываемой системы.

**Ключевые слова:** сегментация, распознавание, сложный фон, вейвлет Гаусса.

## 1. ВВЕДЕНИЕ

Современные технологии распознавания текста работают вполне отлично для печатных документов с высоким разрешением. Распознавание же наложенного текста на изображениях со сложным фоном (когда на изображении кроме текста присутствуют и другие объекты – деревья, дома, люди и т.д.) все ещё является трудно решаемой задачей. Существует специализированные приложения OCR, которые доступны в качестве коммерческого продукта для распознавания отсканированных документов с высоким разрешением, системы распознавания номерных знаков и рукописных символов [6]. Но эти приложения, как правило, не доступны для большинства изображений окружающей среды. Это связано со следующими трудностями, возникающими при распознавании текста на изображении:

- низкое разрешение изображения,
- сложный и структурированный фон, такой, что достоверное различие между символами и фоном не возможно,
- различные текстовые шрифты и размеры.

## 2. ПОСТАНОВКА ЗАДАЧИ

Целью работы является разработка системы распознавания текста на изображениях со сложным фоном, которая использует в качестве алгоритма сегментации метод кратчайшего пути, основываясь на яркости пикселей, а в качестве алгоритма распознавания символов – алгоритм CSS, который разглаживает контур буквы с помощью Гауссовой функции ядра и отслеживает её точки перегиба [1-5]. Для более точной классификации будет представлен расширенный алгоритм CSS для генерации особых точек для вогнутых и выпуклых сегментов контура. Для расчета кривизны кривых будет использоваться вейвлет Гаусса. На вход для обработки подается текстовая область, а на выходе будет генерироваться распознанный текст.

## 3. СЕГМЕНТАЦИЯ СИМВОЛОВ

Шаг сегментации очень важен в данном подходе, т.к. методики классификации, основанные на контурном анализе, терпят неудачу без надёжной сегментации. Алгоритм сегментации будет основываться на модернизированном алгоритме нахождения кратчайшего пути [2] и состоять из следующих этапов:

1 Этап: Перед тем как начать разрез текста на отдельные символы, необходимо задаться начальными точками разрезов. Очевидно, что вертикальная гистограмма текстовой области (рис. 1а) покажет яркие пики в местах, где символ имеет чёткую вертикальную линию, менее яркие пики в местах, где символ имеет небольшие глифовые перемычки (например, горизонтальная перемычка буквы «н»), и низкие значения гистограммы в местах разделения символов (рис. 1б). Определив вогнутые участки гистограммы, методом поиска экстремумов, можно найти области, достаточно похожие на места разделения символов.



Рис 1: Сегментация слова на символы.

2 Этап: Определившись с начальными точками разреза, можно было бы опустить прямые из этих точек и получить сегментированные символы. К сожалению, из-за излишней линейности данного подхода и зашумленности исходного изображения, разрез не всегда может пройти между буквами (рис. 2а). Для этого ограничим путь движения разреза тремя направлениями: на один пиксель вниз, на один пиксель вниз и влево и на один пиксель вниз и вправо. В итоге алгоритм должен принять решение, в каком из трёх направлений двигаться, решение принимается на основе яркости пикселей допустимых путей: выигрывает пиксель с наименьшим значением яркости (рис. 2б).

3 Этап: Для того чтобы уменьшить количество бесполезных движений вправо или влево, связанное с шумами на изображении, пиксели смещающие движение разреза по горизонтали умножаются на коэффициент  $\zeta > 1$  с целью наказания движения в сторону. Хороший результат показал коэффициент  $\zeta = 1.4$ . Все веса точек пути разреза, включая

умножение на  $\zeta$ , аккумулируются в переменной  $S$ , названной стоимостью пути, для дальнейшего использования.

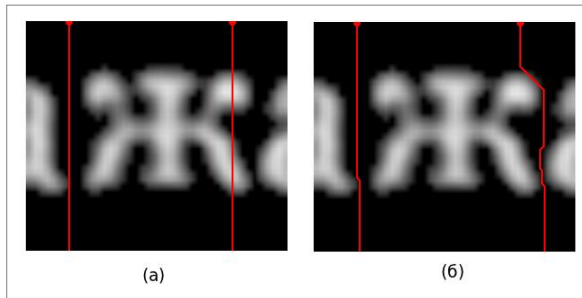


Рис 2: Пути разреза символов.

4 Этап: Несмотря на то, что описанный выше подход позволяет разрезать изображение на наиболее подходящие области, полученные разрезы всё ещё могут проходить через центры символов (рис. 1в) один или более раз. Очевидно, что стоимость пути разреза, прошедшего через начертание глифа, будет существенно большей стоимости разреза, прошедшего между символами. Воспользовавшись методом определения выбросов из математической статистики, приняв множество стоимостей разрезов как выборку, а ложные разрезы как выбросы, можно достаточно точно отфильтровать ложные разрезы. Хороший результат показал фильтр, основанный на межквартильном расстоянии [4] усечённом сверху.

Отфильтрованные таким образом разрезы имеют наиболее удовлетворительный результат (рис. 1г). Основным преимуществом этого подхода является тот факт, что не требуется порога для нахождения разделений между символами.

#### 4. КЛАССИФИКАЦИЯ СИМВОЛОВ

Далее требуется выполнить анализ контура буквы и получить признаки для классификации для каждой из выделенных букв.

Для обучения мы сохраняем инверсное и обычное CSS изображение обучаемой выборки, которое затем сравниваем с получаемым на изображении, поданном на распознавание.

##### 4.1 Стандартный CSS алгоритм

Он состоит из следующих этапов:

1 Этап: Для получения контура каждой из букв применяется алгоритм Кенни [3].

2 Этап: Символы могут оказаться размазанными, и контур, возможно, будет иметь разрывы в некоторых местах. Для устранения этих разрывов применяется морфологическое замыкание. Результат этого шага представлен на рис. 3.



Рис 3: Результат 1 и 2 этапа классификации.

3 Этап: Так как алгоритм CSS работает только с одним контуром, для букв с несколькими контурами оставляем только внешний (рис.4).



Рис 4: Результат 3 этапа классификации.

4 Этап: Имея закрашенные фигуры, нужно получить их контуры, но уже не просто в виде картинок, а в виде функций от одной переменной. Для этого обходим букву по контуру, собирая его координаты, и нормируем по параметру длины дуги  $u$  (рис.5).

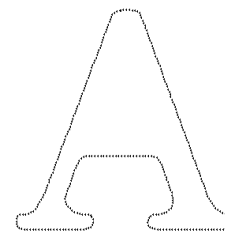


Рис 5: Результат 4 этапа классификации.

5 Этап: После предыдущего этапа у нас будет плоская кривая  $\Gamma(u)$

$$\Gamma(u) = \{(x(u), y(u)) | u \in [0, 1]\}, \quad (1)$$

с нормированным параметром длины дуги  $u$ . Кривая сглаживается одномерной гауссианой с ядром  $g(u, \sigma)$  с шириной  $\sigma$ . Деформация замкнутой плоской кривой представлена в виде

$$\Gamma(u, \sigma) = \{(X(u, \sigma), Y(u, \sigma)) | u \in [0, 1]\}, \quad (2)$$

где  $X(u, \sigma)$  и  $Y(u, \sigma)$  обозначают компоненты  $x(u)$  и  $y(u)$  после свертки с  $g(u, \sigma)$ . Вид кривой представлен на рис.6



Рис 6: Результат 5 этапа классификации.

6 Этап: Рассчитаем кривизну  $\kappa(u, \sigma)$  видоизменённой кривой. В стандартном методе предлагается использование частных производных полученной кривой  $X_u(u, \sigma)$ ,  $X_{uu}(u, \sigma)$ ,  $Y_u(u, \sigma)$  и  $Y_{uu}(u, \sigma)$ . А после предыдущего этапа это функции свертки исходной кривой и функции Гаусса. Так как расчет производных от кривой, представленной дискретными значениями, сложен, и его результат не даст ожидаемой сглаженности, предлагается воспользоваться правилом дифференцирования [7] и вместо этого использовать свертку первую и вторую производные функции Гаусса, т.е. WAVE-вейвлет и MHAT-вейвлет. И тогда по

$$k(u, \sigma) = \frac{X_u(u, \sigma) * Y_u(u, \sigma) - X_u(u, \sigma) * Y_u(u, \sigma)}{(X_u(u, \sigma)^2 + Y_u(u, \sigma)^2)^{3/2}} \quad (3)$$

рассчитаем кривизну  $k(u, \sigma)$  видоизменённой кривой.

7 Этап: Получим CSS изображение  $I(u, \sigma)$  (рис. 7), определяющееся как

$$I(u, \sigma) = \{(u, \sigma) | k(u, \sigma) = 0\}. \quad (4)$$



Рис 7: Стандартное CSS изображение символа «А».

Изображение CSS показывает нулевые пересечения с точки зрения их расположения на контуре и ширины ядра Гаусса.

Во время процесса деформации, нулевые пересекающие сливаются, так как переходы между сегментами контура различной кривизны уравниваются. Следовательно, после определённого количества итераций, точки перегиба исчезают, а форма замкнутой кривой становится выпуклой. Значительные свойства контура, которые видны во время большого числа итераций, выливаются в высокие пики на CSS изображении. Тем не менее, области с быстро меняющейся кривизной, вызванной шумом, выдают только небольшие локальные максимумы.

#### 4.2 Расширенный CSS алгоритм

Основным недостатком стандартного подхода является неполноценное представление выпуклых сегментов контура. Так как CSS изображение представляет расположение точек перегиба, нужны сегменты вогнутого контура. А контур символов без вогнутых сегментов (например «О» рис. 8) не может быть выделен.

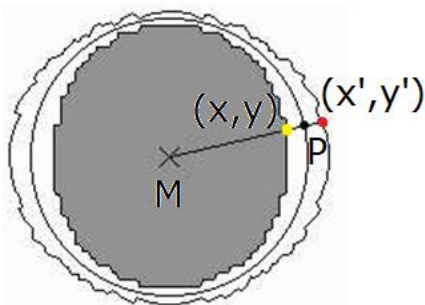


Рис 8: Стандартный и инверсный контур буквы «О».

Мы используем вначале стандартный CSS подход, чтобы получить собственные векторы функций, которые классифицируют вогнутые части контура очень хорошо. Общая идея состоит сейчас в том, чтобы создать второй контур, который обеспечит дополнительные возможности для выпуклых сегментов оригинального контура. Оригинальный контур переходит в новый контур с инвертированной кривизной. Сильновыпуклые сегменты исходного контура становятся вогнутыми сегментами отображенного контура. Значительная кривизна в исходном контуре также существует в отображенном контуре.

Чтобы создать преобразованный контур, мы ограничиваем контур символа окружностью радиуса  $R$  и определяем точку  $P$  окружности, ближайшую к каждому пикселю контура. Контур пикселей зеркально отображается относительно окружности в  $P$ . Сегменты контура, которые имеют сильновыпуклую кривизну, сопоставляются с вогнутыми сегментами. Степень искривления преобразованных контуров зависит от радиуса  $R$  и от кривизны оригинального контура.

Расчёт преобразованного контура достаточно быстр. Каждый пиксель контура в позиции  $u$  замкнутой плоской кривой  $(x(u), y(u))$  отображается на кривой  $(x'(u), y'(u))$ . Центр окружности  $(M_x, M_y)$  с радиусом  $R$  рассчитывается как среднее положение контура пикселей. Результат создания инверсного CSS изображения показан на рис. 9.



Рис 9: CSS изображение символа «А»: а) – стандартное, б) - инверсное.

## 5. ТЕСТИРОВАНИЕ

Система обучалась на выборке букв русского алфавита на 69 изображениях. Все обучающие буквы были написаны шрифтом Calibri с размером 100. Тестирование системы проводилось на выборке из 115 изображений подобранных с из интернета фотографий, а так же сделанных самостоятельно. Диапазон качества изображений был испробован от 300 dpi до 100dpi. В качестве шрифтов для тестируемых изображений использовался не только Calibri, но и Times New Roman, Courier, Tahoma, Verdana, Arial Black. Так же на изображениях присутствовал фон различной сложности (рис. 10).

В качестве критерия оценки распознавания была выбрана ошибка первого рода. В роли гипотезы выступает правильное распознавание символа. Процент распознавания определялся как:

$$Rez = (x * 100\%) / y \quad (5)$$

где  $x$  - общее количество символов на изображении, а  $y$  - количество правильно распознанных. Общий результат вычислялся как среднее значение по всей выборке оттестированных изображений и составил 85%.

## 6. ЗАКЛЮЧЕНИЕ

В результате выполнения описанных алгоритмов на языке программирования Java 1.7 нами была получена система, которая считывает текст с изображения. На вход для обучения подается выборка букв русского алфавита. Распознавание же мы проводили на изображениях с текстом различного шрифта и с различным фоном (рис. 10).

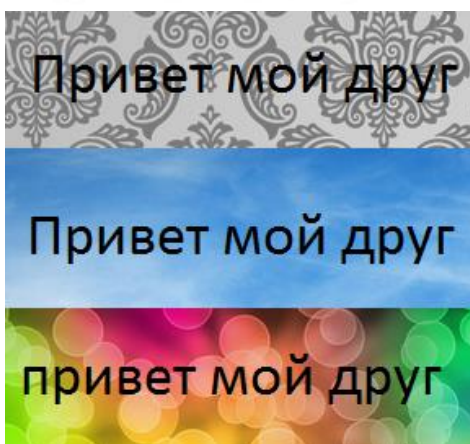
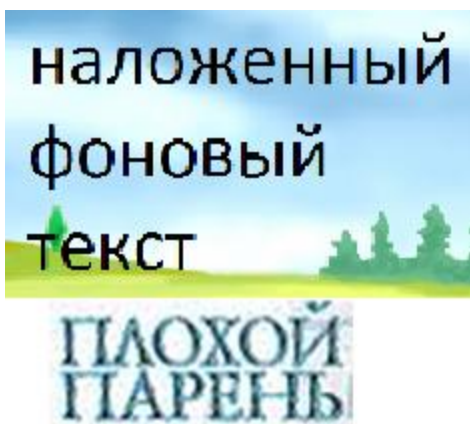


Рис 10: Текстовые изображения на сложном фоне

Результаты показали (рис. 11, 12), что для всех этих шрифтов осуществляется распознавание текста, даже не смотря на то, что обучение проводилось на другом шрифте. Также были проверены различные размеры шрифта в диапазоне от 14 до 30 с курсивным и жирным стилем. Распознавания дают правильный текст в 85% (рис.11, 12), что говорит о неплохой надёжности, но и о потенциальной дальнейшей работе в этом направлении.

```
[
наложенный
фондовый
текст
]
```

```
[
ПЛОХОЙ
ПАРЕНЬ
]
```

Рис 11: Результаты работы системы

```
[
привет Мой друг
]
[
привет Мой друг
]
[
привет Мой друг
]
```

Рис 12: Результаты работы системы

## 7. ССЫЛКИ

- [1] Kopf S., Haenselmann T., Effelsberg, W. *Enhancing curvature scale space features for robust shape classification. Multimedia and Expo, 2005. ICME 2005. IEEE International Conference, 2005.*
- [2] Wang J., Jean J. *Segmentation of merged characters by neural networks and shortest-path, Pattern Recognition, 1994, pp 649-658.*
- [3] Билл Грин. *Алгоритм выделения контуров CANNY, Дрексельская лаборатория автоматизированных систем, 2002.*
- [4] Дэйвид Г. *Порядковые статистики. Москва «Наука»: Главная редакция физико-математической литературы, 1979.*
- [5] Кобзарь Г.А. *Модель межмасштабного пространства кривизны для представления формы геометрических объектов // Искусственный интеллект. – 2008. – с. 92-101.*
- [6] <http://www.cvisiontech.com/reference/general-information/ocr-applications.html>
- [7] [ru.wikipedia.org/wiki/Свёртка\\_\(математический\\_анализ\).](http://ru.wikipedia.org/wiki/Свёртка_(математический_анализ))

## Об авторах

Анна Балахонцева – студентка ОНПУ. Ее адрес: a.c.balackhontseva@gmail.com.

Александр Годоба – студент ОНПУ. Его адрес: hedgecrab@gmail.com.

Тьен Нгуен – аспирант ОНПУ. Ее адрес: ktien85@ukr.net.

# Алгоритм нелокального среднего на основе разложения по функциям Эрмита в задачах компьютерной томографии

Н. В. Мамаев<sup>1</sup>, А. С. Лукин<sup>2</sup>, Д. В. Юрин<sup>3</sup>, М. А. Глазкова<sup>4</sup>, В. Е. Синицин<sup>5</sup>,  
Лаборатория математических методов обработки изображений<sup>6</sup>,  
Факультет вычислительной математики и кибернетики

Московский государственный университет им. М. В. Ломоносова, Россия  
ФГУ «Лечебно-реабилитационный центр» Минздравсоцразвития РФ<sup>7</sup>

<sup>1</sup>[mamaev.nikolay93@mail.ru](mailto:mamaev.nikolay93@mail.ru), <sup>2</sup>[lukin@ixbt.com](mailto:lukin@ixbt.com), <sup>3</sup>[yurin@cs.msu.su](mailto:yurin@cs.msu.su), <sup>4</sup>[mary-ga@yandex.ru](mailto:mary-ga@yandex.ru), <sup>5</sup>[vsini@mail.ru](mailto:vsini@mail.ru),  
<sup>6</sup><http://imaging.cs.msu.su>, <sup>7</sup><http://www.med-rf.ru/stuff/gs/>

## Аннотация

Предлагается алгоритм подавления шума (HeNLM), представляющий собой модификацию метода локальных струй (LJNLM-LR). Суть модификация заключается в разложении локальных окрестностей пикселя по функциям Эрмита, которые образуют ортонормированную базис, вместо производных функции Гаусса. Тестирование алгоритма на изображениях компьютерной томографии в сравнении с методом локальных струй и алгоритмом нелокального среднего показало лучшее сохранение высокодетальных областей, четких границ, и отсутствие потерь важной медицинской информации.

**Ключевые слова:** компьютерная томография, фильтрация шума, алгоритм нелокального усреднения, метод локальных струй, функции Эрмита.

## 1. ВВЕДЕНИЕ

Компьютерная томография (КТ) – метод расчетного восстановления с помощью преобразования Радона изображений срезов объекта по набору рентгеновских снимков, просвечивающих объект в разных направлениях [7]. В получающихся изображениях всегда присутствует шум, уровень которого тем выше, чем ниже мощность излучения. Поэтому эффективное удаление шума в результатах медицинских обследований с помощью КТ позволяет уменьшить дозу облучения пациентов и является актуальной задачей [3,9]. Шум на изображениях КТ близок к Гауссову [4,8].

В последнее время одними из наиболее эффективных методов удаления шума являются алгоритмы, в которых веса пикселей зависят от схожести некоторых величин, характеризующих окрестности пикселей [2], [6], [10]. Первый алгоритм, известный как алгоритм нелокального усреднения (NLM), вычисляет веса в зависимости от евклидова расстояния между блоками вокруг пикселей. Алгоритмы [6] и [10] являются модификациями [2] и позволяют снизить вычислительную сложность за счет замены евклидова расстояния между блоками на расстояние между векторами признаков. Размерность таких векторов, как правило, значительно ниже, чем количество пикселей в блоке. Алгоритм GFNLM [10] основан на признаках, полученных путем разложения окрестности пикселя по функциям Габора. Алгоритмы LJNLM-LR и LJNLM-UR [6] основаны на разложении окрестности пикселя в ряд Тейлора, коэффициенты которого определяются производными функции Гаусса (Local Jets). Достоинством [6] является поиск похожих блоков не только относительно плоско-параллельного сдвига, но и относительно вращения.

В настоящей статье предлагается метод нелокальной фильтрации, основанный на разложении блока пикселей по функциям Эрмита (HeNLM). Наследуя все достоинства подхода [6], метод HeNLM лучше различает текстуры за счет большей независимости компонент вектора-параметра, обусловленной ортонормированностью функций Эрмита и лучшего описания ими высокочастотных компонент локальной окрестности.

## 2. МЕТОД ЛОКАЛЬНЫХ СТРУЙ ДЛЯ АЛГОРИТМА НЕЛОКАЛЬНОГО УСРЕДНЕНИЯ

В алгоритме NLM значение выходного пикселя  $f(x, y)$  является взвешенной суммой значений исходного изображения  $I(x, y)$  по некоторой окрестности  $Q$ :

$$f(x, y) = \frac{1}{\sum_{(\xi, \eta) \in Q} w} \sum_{(\xi, \eta) \in Q} w(x, y, \xi, \eta) I(x + \xi, y + \eta) \quad (1)$$

здесь веса  $w(x, y, \xi, \eta)$  зависят от схожести целых блоков  $v(x, y)$  вокруг пикселя с координатами  $(x, y)$ :

$$w(x, y, \xi, \eta) = \exp\left(-\frac{\|v(x, y) - v(x + \xi, y + \eta)\|_2^2}{2\rho^2}\right) \quad (2)$$

Этот метод обеспечивает высокое качество получаемого изображения, но имеет высокую вычислительную сложность. Кроме того, метод [2] не учитывает поворот блоков пикселей, т.е. пиксели, лежащие на одной границе, но с разным направлением градиента, будут считаться различными и иметь малые веса (1). Это может привести к плохому подавлению шума вдоль границ, где градиент в каждой точке границы имеет разные направления. Так как вычисление весов (2) происходит путем сравнения необработанных значений пикселей, алгоритм NLM сильно чувствителен к шуму.

Метод локальных струй (алгоритмы LJNLM-LR и LJNLM-UR [6]) частично преодолевает недостатки [2]. В LJNLM выходной пиксель также представлен как взвешенная сумма пикселей исходного изображения, но веса зависят от расстояния между векторами признаков, характеризующими некоторую окрестность пикселя. Компонентами вектора признаков в LJNLM являются значения сверток изображения  $I(x, y)$  с производными функции Гаусса на различных масштабах:

$$f_{nm}^{\sigma}(x, y) = I(x, y) * \left( \frac{\sigma^{n+m}}{1+n+m} \frac{d^{n+m} G_{\sigma}}{dx^n dy^m} \right), \quad (3)$$

где функция Гаусса задается выражением:

$$G_{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}, \quad (4)$$

а множитель  $\sigma^{n+m}$  вводится в (3) для того, чтобы отклик фильтра на одинаковые текстуры на разных масштабах был одинаков [11]. Иными словами, если одновременно увеличить  $\sigma$  и во столько же раз растянуть функцию  $I(x, y)$ , то значения (3) не должны изменяться. Знаменатель  $1+n+m$  отражает количество разных производных порядка  $n+m$ .

Для обеспечения инвариантности к повороту полученных признаков они переводятся в систему координат  $(\xi, \eta)$ , где ось  $\xi$  направлена вдоль градиента в точке  $(x, y)$ .

В итоге вектор, характеризующий окрестность пикселя выглядит следующим образом:

$$P = \left\{ \tilde{f}_{nm}^{\sigma}; n+m \leq r, \sigma \in S \right\} \quad (5)$$

где  $\tilde{f}_{nm}^{\sigma}$  – признак в системе координат  $(g, \tau)$ ,  $r$  – максимальный порядок производной,  $S$  – множество масштабов.

### 3. ФУНКЦИИ ЭРМИТА

Функции Эрмита определены следующим образом:

$$\psi_n(x) = \frac{1}{c_n} e^{\frac{x^2}{2}} \frac{d^n (e^{-x^2})}{dx^n} = \frac{(-1)^n}{c_n} H_n(x) e^{-\frac{x^2}{2}},$$

$$\text{где } c_n = \sqrt{\sqrt{\pi} 2^n n!} \quad (6)$$

$$H_n(x) = (-1)^n \frac{d^n e^{-x^2}}{dx^n} e^{x^2} \text{ – полином Эрмита.}$$

Они образуют полную ортонормированную систему в  $L_2(-\infty, +\infty)$  [1, ф.22.11.8,1, ф.22.1.1,22.1.2.,22.2.14]:

$$\int_{-\infty}^{+\infty} \psi_n(x) \psi_m(x) dx = \delta_{nm} \quad (7)$$

Многомасштабные функции Эрмита определим как:

$$\psi_n^{\sigma}(x) = \frac{1}{\sigma} \psi_n\left(\frac{x}{\sigma}\right) \quad (8)$$

Множитель  $1/\sigma$  в (8) приводит к тому, что  $\psi_n^{\sigma}(x)$ , в отличие от (7), оказываются нормированными не на единицу, но обеспечивают более важную нормировку по отклику на текстуры – см. комментарий после формулы (4). Они удовлетворяют дифференциальному уравнению [1, ф.22.6.20]:

$$\psi_n^{\sigma''}(x) + (\sigma^2(2n+1) - x^2)\psi_n^{\sigma}(x) = 0 \quad (9)$$

Некоторые функции Эрмита и производные функции Гаусса представлены на Рис. 1. Видно, что интервалы локализации функций Эрмита и производных функции Гаусса примерно совпадают, однако функции Эрмита позволяют значительно лучше представлять края интервала и не имеют сильного разброса амплитуд. Из уравнения (9) можно получить область локализации функции Эрмита. Действительно,  $\psi_n^{\sigma''} = 0$  в точках  $x = \pm\sigma\sqrt{2n+1}$ , что соответствует последним точкам перегиба функции Эрмита. Область локализации функций Эрмита в основном сосредоточена между крайними перегибами, поэтому носитель функции Эрмита находится между точками  $x = \pm\sigma\sqrt{2n+1}$ . Это выражение позволяет оценить размер окрестности пикселя, по которой происходит разложение по функциям Эрмита, и минимальный размер ядра свертки при численных расчетах.

Двумерные функции Эрмита будем определять [5] как:

$$\psi_{nm}^{\sigma}(x, y) = \psi_n^{\sigma}(x)\psi_m^{\sigma}(y) \quad (10)$$

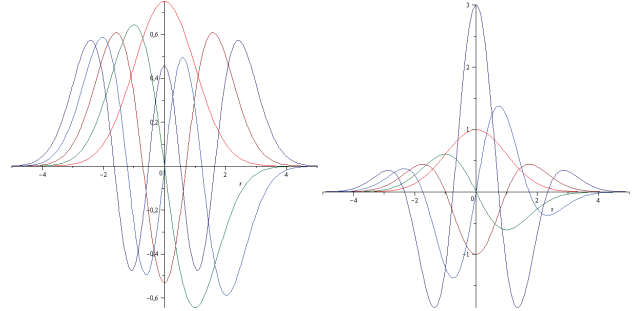


Рис. 1: Функции Эрмита (слева) и производные функции Гаусса (справа).

### 4. АЛГОРИТМ НЕЛОКАЛЬНОГО УСРЕДНЕНИЯ НА ОСНОВЕ ФУНКЦИЙ ЭРМИТА

Мы предлагаем модификацию метода локальных струй [6] с использованием разложения по функциям Эрмита вместо производных функции Гаусса. Элементами вектора признаков теперь являются свертки исходного изображения  $u$  с соответствующими функциями Эрмита:

$$h_{nm}^{\sigma} = u * \psi_{nm}^{\sigma} \quad (11)$$

Так же как и в методе локальных струй, признаки переводятся в систему координат  $(\xi, \eta)$ . Получим явное выражение. Рассмотрим две системы координат, связанные поворотом:

$$\begin{pmatrix} \xi \\ \eta \end{pmatrix} = R \begin{pmatrix} x \\ y \end{pmatrix}, \text{ где } R = \begin{pmatrix} \cos \vartheta & \sin \vartheta \\ -\sin \vartheta & \cos \vartheta \end{pmatrix} \quad (12)$$

Тогда оператор дифференцирования в новых координатах можно выразить через линейную комбинацию операторов в старой системе координат:

$$\frac{d}{d\xi_i} = \sum_{j=1,2} R_{ij} \frac{d}{dx_j}, \text{ где } i=1,2; \xi_1 = \xi; \xi_2 = \eta; j=1,2; x_1 = x; x_2 = y. \quad (13)$$

При этом выражение для производной функции Гаусса из (6) примет вид:

$$\frac{d^{n+m}G_\sigma}{d\xi_1^n d\xi_2^m} = \left( \sum_{j=1,2} R_{1j} \frac{d}{dx_j} \right)^n \left( \sum_{j=1,2} R_{2j} \frac{d}{dx_j} \right)^m G_\sigma \quad (14)$$

Функция Эрмита представляет собой произведение производной функции Гаусса, множителя  $e^{(x^2+y^2)/2}$  и нормировочной константы  $C_i$ . Тогда подставляя (12) в (14), учитывая радиальную симметрию экспоненты и линейность операции свертки (11), нетрудно получить:

$$c_n c_m \tilde{h}_{nm}^\sigma = \sum_{j=0}^{n+m} a_j c_j c_{n+m-j} h_{j,n+m-j}^\sigma, \quad \text{где} \quad (15)$$

$$a_j = \sum_{k=\max(0, j-m)}^{\min(j, n)} (-1)^{j-k} (\cos \vartheta)^{m-j+2k} (\sin \vartheta)^{n+j-2k} C_n^k C_m^{j-k}$$

$$\text{и } C_n^k = \frac{n!}{k!(n-k)!}.$$

Заметим, что (15) могут быть применены и для метода локальных струй (3),(5). Для этого в (15) надо заменить  $h_{nm}^\sigma$  на  $f_{nm}^\sigma$ , а константы  $C_i$  положить равными 1.

Так как новый базис  $\xi, \eta$  выбран так, чтобы ось  $\xi$  совпала с вектором градиента яркости изображения  $\vec{g} = (I(x, y) * dG_\sigma/dx, I(x, y) * dG_\sigma/dy)^T$ , получим, что в (12) и (15)  $\vec{g}/|\vec{g}| = (\cos \vartheta, \sin \vartheta)^T$ .

Использование функций Эрмита вместо производных функций Гаусса представляется лучшим для характеристики окрестности пикселя в силу их ортонормированности, что означает меньшую взаимозависимость компонент  $h_{nm}^\sigma$  (11) и, соответственно, их большую значимость. Также происходит расширение области локализации (Рис. 1), и, следовательно, лучше учитываются периферийные данные локальной окрестности и высокочастотные компоненты.

## 5. РЕЗУЛЬТАТЫ

Исходные, изображения компьютерной томографии (КТ) были предоставлены ФГБУ «Лечебно-реабилитационный центр» Минздрава России и выполнялась на 64-спиральном компьютерном томографе Discovery CT 750 MD (General Electric). На Рис. 2. показан один из срезов КТ, а на Рис. 3. показаны увеличенные фрагменты изображений из рамки на рис. 2, отфильтрованных различными методами.

Для алгоритма нелокального усреднения размер области усреднения был выбран 21x21, размер блока пикселей – 7x7. Для остальных алгоритмов фильтрации область усреднения – 21x21, максимальный порядок производных Гаусса (Local Jets [6]) и функций Эрмита равен 4, значения  $\sigma$  для функции Гаусса равны 1.0 и 4.0. Таким образом, размерность вектора признаков для методов LJNLM-LR и HeNLM равна 20. Пара-

метр фильтрации  $\rho$  выбран вручную так, чтобы PSNR между зашумленным и отфильтрованным изображением во всех случаях был одинаков.



Рис. 2: Исходное изображение компьютерной томографии. Белой рамкой обозначен увеличенный фрагмент изображения, показанный на рис. 3.

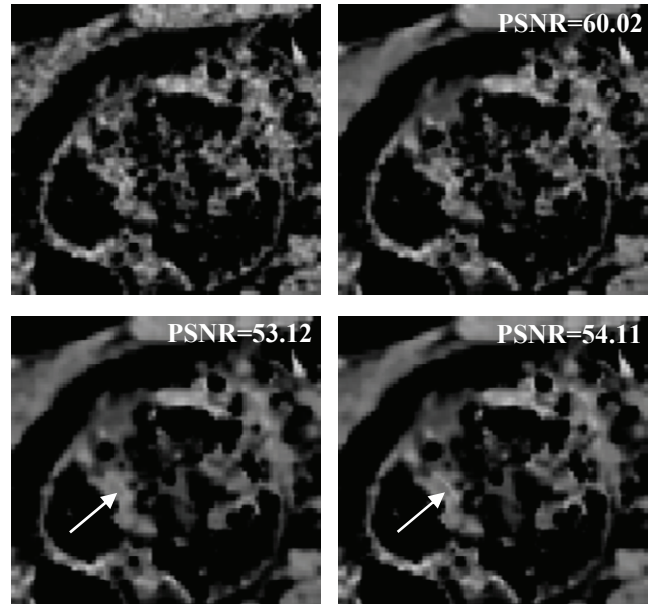
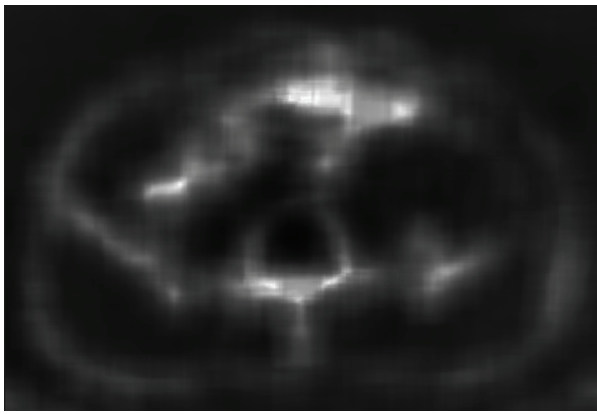


Рис. 3: Слева направо и сверху вниз: увеличенный фрагмент исходного изображения, и результаты работы алгоритмов фильтрации: нелокального усреднения (NLM), локальных струй (LjNLM-LR) и предложенного алгоритма (HeNLM).

На Рис. 3 видно, что метод локальных струй и предложенный нами метод осуществляют фильтрацию лучше, чем алгоритм нелокального усреднения, в местах, где присутствует большое количество мелких деталей. Предложенный метод при фильтрации сохраняет некоторые детали, которые метод локальных струй размывает: на Рис. 3 стрелкой показана линия, которая при фильтрации методом LjNLM-LR размывается, но остается при фильтрации предложенным методом HeNLM.

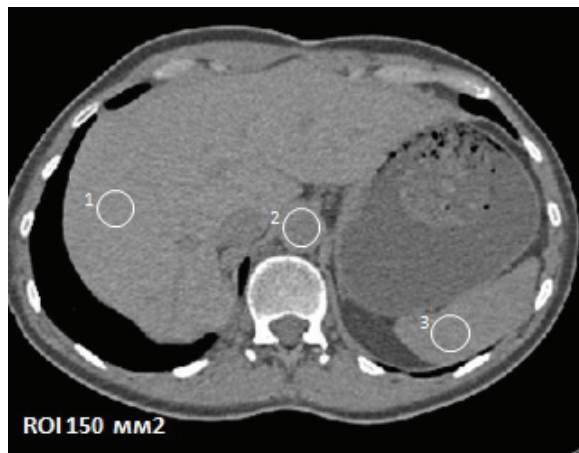
На Рис. 4 показано сравнение PSNR для метода LjNLM-LR и предложенного метода. Белый цвет отображает места, где PSNR выше у предложенного метода, черный – у метода локальных струй. Видно, что предложенный метод удаляет шум сильнее в высоко детализированных областях, а метод локальных струй – в однородных областях.





**Рис. 4:** «Маска» PSNR для предложенного метода и метода локальных струй.

Качество изображений, отфильтрованных различными методами, было оценено двумя врачами-рентгенологами и сопоставлено с исходными данными. Анализ включал в себя измерение уровня шума, локального отношения сигнал-шум (SNR) и субъективную оценку специалистом (визуализация мелких деталей). Для измерения уровня сигнала и шума при исследовании органов брюшной полости была выбрана зона интереса (ROI) 150 мм<sup>2</sup> в пяти различных точках (Рис.5): на уровне брюшной аорты, паренхимы печени, селезенки, поджелудочной железы, почек. Средние значения шума и SNR представлены как медиана (межквартильный размах). Для расчета статистически значимых различий между исходными изображениями и результатами работы алгоритмов фильтрации использовали тест Фридмана.



**Рис. 5:** Измерение уровня сигнала и шума на следующих уровнях: (1) печень; (2) аорта; (3) селезенка. ROI=150 мм<sup>2</sup>.

Полученные данные показали, что благодаря применению различных алгоритмов фильтрации, можно улучшить качество полученных изображений. Так, при сравнении исходных и отфильтрованных изображений было отмечено достоверное снижение уровня шума и повышения SNR ( $p < 0,05$ ) в результате работы алгоритмов фильтрации. При этом статистически значимых различий между уровнем шума и SNR при работе трех разных алгоритмов фильтрации: нелокального усреднения, метода локальных струй и предложенного метода получено не было ( $p = 0,6$  и  $0,55$  соответственно). См. Таб. 1; Рис.6.

Необходимо также отметить, что визуализация мелких деталей была сопоставима с исходными данными.

**Таблица 1**

Параметр	Исходные изображения	HeNLM	LJNLM-LR	NLM	P
Уровень шума	20,6 [18,5-21,3]	4,6 [3,5-5,6]	5,4 [3,6-6,4]	2,7 [2-4,3]	0,005
SNR	2,1 [1,9-3,4]	10,2 [7,5-13]	6,8 [6-16,5]	14,4 [8-31]	0,003



**Рис. 6:** Представлены значения плотности и шума паренхимы левой почки, выбранная область интереса составила 150 мм<sup>2</sup>. Слева направо и сверху вниз: исходное изображение, и результаты работы алгоритмов фильтрации: HeNLM, LJNLM-LR, NLM. Отмечается значительное снижение уровня шума при использовании предложенных алгоритмов фильтрации.

## 6. ЗАКЛЮЧЕНИЕ

В статье предложен метод фильтрации изображений компьютерной томографии на основе функций Эрмита. Использование этого метода позволит улучшить качество полученных в ходе исследования изображений, благодаря снижению уровня шума и повышению соотношения сигнал-шум, без потери диагностической информации.

## 7. БЛАГОДАРНОСТИ

Работа выполнена при поддержке гранта РФФИ 13-07-00584.

## 8. ЛИТЕРАТУРА

- [1] M. Abramowitz, I.A. Stegun. "Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables" // U.S. National Bureau of Standards (1964); Dover, New York 1965.
- [2] A. Buades. "A non-local algorithm for image denoising" // IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, -V. 2, -P. 60-65.

- [3] M.K. Kalra, C. Wittram, M.M. Maher, A. Sharma, G.B. Avinash, K. Karau, T.L. Toth, E. Halpern, S. Saini, J.A. Shepard. "Can noise reduction filters improve low-radiation-dose chest CT images? Pilot study." // *Radiology* 2003; -V. 228, -P. 257–264.
- [4] M.F. Kijewski, P.F. Judy. "The noise power spectrum of CT images" // *Phys. Med. Biol.*, 1987, -V. 32, -No 5, -P. 565–575. Printed in the UK
- [5] A.S. Krylov, A.V. Kutovoi, Wee Kheng Leow. "Texture Parameterization With Hermite Functions" // *Proc. Of The 12<sup>th</sup> International Conference Graphicon'2002*. Russia, Nizhny Novgorod, 2002, -P. 190–194.
- [6] A. Manzanera. "Local Jet based similarity for NL-Means filtering" // *20<sup>th</sup> International Conference on Computer Vision and Pattern Recognition (ICPR)*, 2010, -P. 2668–2671.
- [7] F. Natterer. "The Mathematics of Computerized Tomography" // *SIAM: Society for Industrial and Applied Mathematics*, Philadelphia, PA, USA, July 2001, 222 p.
- [8] T.H. Newton, D.G. Potts. "Radiology of the Skull and Brain. Vol. 5. Technical aspects of computed tomography" // *C. V. Mosby Co*, St. Louis, 1981 585 p., ISBN 0-8016-3662-0.
- [9] M.V. Storozhilova, A.S. Lukin, D.V. Yurin, V.E. Sinitsyn. "Two approaches for noise filtering in 3D medical CT-images" // *Proc. Of The 22-th Int. Conf. on Computer Graphics GraphiCon'2012*. Moscow, Russia, 2012, -P. 68–72.
- [10] S. Wang, Y. Xia, Q. Liu, J. Luo, Y. Zhu, D. Feng. "Gabor feature based nonlocal means filter for textured image denoising" // *Journal of Visual Communication and Image Representation*, -V. 23, Issue 7, Oct. 2012, -P.1008–1018.
- [11] T. Lindeberg. "Scale-Space Theory in Computer Vision" // *Kluwer Academic Publishers*, Dordrecht, 1994

## Об авторах



Мамаев Николай Владимирович, студент кафедры математической физики, факультета вычислительной математики и кибернетики Московского государственного университета им. М.В.Ломоносова.

Nikolay V. Mamaev is a student at Chair of Mathematical Physics in Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University, Russia.

His e-mail is: [mamaev.nikolay93@mail.ru](mailto:mamaev.nikolay93@mail.ru)



Лукин Алексей Сергеевич, к.ф.-м.н., сотрудник лаборатории математических методов обработки изображений факультета вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова.

Alexey S. Lukin (Ph.D.) is a member of scientific staff at the Laboratory of Mathematical Methods of Image Processing in Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University, Russia.

His e-mail is: [lukin@ixbt.com](mailto:lukin@ixbt.com)



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.

Dmitry V. Yurin (Ph.D.) is a senior researcher at laboratory of Mathematical Methods of Image Processing in Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University, Russia.

His e-mail is: [yurin\\_d@inbox.ru](mailto:yurin_d@inbox.ru), [yurin@cs.msu.su](mailto:yurin@cs.msu.su)



Глазкова Мария Александровна, ординатор кафедры лучевой диагностики Российской медицинской академии последипломного образования (РМАПО).

Maria A. Glazkova is an attending physician at Chair of X-Ray diagnostics of Russian Medical Academy of Postgraduate Education (RMAPO).

Her e-mail is: [mary-ga@yandex.ru](mailto:mary-ga@yandex.ru).



Синицын Валентин Евгеньевич, проф., д.м.н. Руководитель центра лучевой диагностики ФГУ «Лечебно-реабилитационный центр» Минздравсоцразвития РФ. Руководитель курса лучевой диагностики и лучевой терапии Факультета фундаментальной медицины МГУ им. М.В.Ломоносова, проф. кафедры лучевой диагностики РМАПО.

Valentin E. Sinitsyn (MD) is a Head of Radiology, Federal Center of Medicine and Rehabilitation, Moscow.

His e-mail is: [vsini@mail.ru](mailto:vsini@mail.ru)

## A Non-Local Means Algorithm Using Hermite Functions Expansion for CT applications

### Abstract

We propose a new algorithm for noise suppression – HeNLM. It is a modification of an existing local jets based method LJNLM-LR. The neighborhood of a pixel is expanded in a series of Hermite functions, which, unlike Gaussian derivatives, form an orthonormal basis. Testing of our method against Local Jets and NLM has shown better preservation of detail-rich areas, sharp edges in CT scans, without loss of medically-important information.

**Keywords:** *computed tomography, noise filtering, non-local means, local jet, Hermite functions.*

# Детектирование жирных линий и древовидных структур на изображениях

А.Е. Левашов<sup>1</sup>, Д.В. Юрин<sup>2</sup>

Лаборатория математических методов обработки изображений<sup>3</sup>,  
Факультет вычислительной математики и кибернетики

Московский государственный университет им. М. В. Ломоносова, Россия  
<sup>1</sup>[alexeylevashov89@gmail.com](mailto:alexeylevashov89@gmail.com), <sup>2</sup>[yurin@cs.msu.su](mailto:yurin@cs.msu.su), <sup>3</sup><http://imaging.cs.msu.su>

## Аннотация

Предлагается новый многомасштабный детектор жирных линий с улучшенным методом неадекватного подавления в пространстве переменных разрешений. Используя результаты работы детектора как предварительную разметку, методом адаптивного выбора порога и заливки производится дополнение линий в точках ветвления, установление верного соединения разорванных линий и уточнение границ жирных линий. Алгоритм тестируется на синтетических, ландшафтных и медицинских изображениях.

**Ключевые слова:** Ridge detection, feature detection, scale space.

## 1. ВВЕДЕНИЕ

Существует несколько основных алгоритмов для нахождения жирных линий. По-видимому, самым первым из них был подход [2], он не использует многомасштабного анализа и, как следствие, детектирует линии только небольшой толщины, что является очень серьезным ограничением. Подход [5] основан на многомасштабном анализе, однако он представляет собой скорее хорошее теоретическое обоснование для многомасштабного подхода, чем практически эффективный алгоритм. За истекшие 15 лет было немного случаев его использования в научных исследованиях. Детектор [3], основанный на обобщении процедуры неадекватного подавления на случай трехмерного пространства переменных разрешений  $(x, y, t)$  представляется более практичным для поиска жирных линий на изображениях. Однако одним из важных недостатков детектирования жирных линий при таком подходе является пропуск точек ветвления таких линий и потеря самих линий в окрестности точек ветвления. Это нарушает связность линий и существенно усложняет анализ древовидных структур на изображениях. Примером таких задач могут быть выделение веток и проводов на любительских фотографиях с целью их ретуширования, выделение дорожной и речной сети на аэрокосмических изображениях с целью актуализации картографической информации, выделение кровеносных сосудов на медицинских изображениях и обработка трехмерных изображений компьютерной томографии с целью диагностики заболеваний. Во всех этих задачах точки ветвления играют важную роль, однако, в окрестности точек ветвления жирных линий детектор часто ошибочно не детектирует такие линии по следующим причинам. Во-первых, для процедуры неадекватного подавления возникает неопределенность выбора направления, т.к. оба собственных значения Гессiana оказываются близки. Во-вторых, в окрестности точки ветвления две (или более) жирные линии близко сходятся, поэтому оператор Лапласа может дать максимальный отклик на существенно большем масштабе (в два и более

раза, что приводит к разрыву линий). В-третьих, отклик может снижаться из-за нарушения математической модели согласованной фильтрации - линия одного цвета в пределах центрального лепестка второй производной функции Гаусса, и фон другого цвета в боковых лепестках. Целью данной статьи является устранение описанных недостатков в подходе [3], что приведет к существенному улучшению качества и упрощению дальнейшего анализа.

## 2. ДЕТЕКТИРОВАНИЕ ЖИРНЫХ ЛИНИЙ

Рассмотрим алгоритм [3], пусть  $I(x, y)$  – яркость исходного дискретного изображения в оттенках серого. Определим

$$L(x, y, t) = G(x, y, t) * I(x, y),$$

$$G(x, y, t) = \frac{1}{2\pi t} e^{-\frac{x^2+y^2}{2t}}, \quad t = \sigma^2, \quad (1)$$

где  $*$  – есть операция свертки, а  $G(x, y, t)$  – функция Гаусса. Под  $L_x, L_y, L_{xx}, L_{xy}, L_{yy}, \dots$  будем понимать свертки изображения  $I(x, y)$  с соответствующей производной функции Гаусса. Непосредственным дифференцированием легко показать [6], что многомасштабное представление изображения  $L(x, y, t)$  удовлетворяет уравнению

$$L_t = \frac{1}{2} (L_{xx} + L_{yy}), \quad (2)$$

где производные взяты на масштабе  $t$ .

## 3. ДЕТЕКТИРОВАНИЕ ЖИРНЫХ ЛИНИЙ С ВЫБОРОМ ОПТИМАЛЬНОГО МАСШТАБА

Рассмотрим алгоритм [3] для детектирования жирных линий с пропуском точек ветвлений. Пусть толщина линии фиксирована и равна  $2\sigma$ , Тогда фильтром для поиска жирных линий может служить оператор Лапласа или наибольшее собственное значение матрицы Гессе  $\mathbf{H}_{22}$  в точке  $(x, y)$

$$\mathbf{H}_{22} = \begin{pmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{pmatrix}, \quad (3)$$

где производные взяты на масштабе  $t = \sigma^2$ . В каждой точке  $(x, y)$  собственные вектора  $(\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2)$  будут показывать

главные направления кривизны: собственный вектор  $V_1$ , соответствующий наибольшему по модулю собственному числу  $\lambda_1$ , будет направлен перпендикулярно касательной к линии в этой точке. Исходя из этого, получается алгоритм детектирования центральных точек жирных линий [3] как нахождение локальных максимумов и локальных минимумов величины  $\lambda_1$  по направлению  $V_1$ . Т.е. двукратным выполнением процедуры немаксимального подавления по окну  $3 \times 3$  пикселя. Если масштаб детектирования  $t$  не вполне соответствует ширине линии, то очевидно, что Лапласиан и  $\lambda_1$  будут меньше, чем при точном совпадении, поэтому анализируя отклик фильтра на соседних масштабах можно определить, детектирована ли линия на своем истинном масштабе. В пространстве переменных разрешений матрица Гессе имеет вид:

$$H_{33} = \begin{pmatrix} L_{xx} & L_{xy} & L_{xt} \\ L_{xy} & L_{yy} & L_{yt} \\ L_{xt} & L_{yt} & L_{tt} \end{pmatrix}, \text{ где с учётом (2)}$$

$$L_{xt} = \frac{1}{2}(L_{xxx} + L_{xyy}), \quad L_{yt} = \frac{1}{2}(L_{xxy} + L_{yyy}),$$

$$L_{tt} = \frac{1}{4}(L_{xxxx} + 2L_{xxyy} + L_{yyyy})$$

В [3] предлагается выполнять модифицированную процедуру немаксимального подавления (NMS) сразу по двум направлениям в трехмерном пространстве переменных разрешений [6]  $(x, y, t)$  и окну  $3 \times 3 \times 3$ . В качестве направлений, по которым будет делаться NMS, выбирались проекции собственных вектора  $\bar{V}_1$  и  $\bar{V}_2$ , соответствующих двум максимальным собственным числам  $(\lambda_1, \lambda_2)$  матрицы  $H_{33}$  на плоскость, задаваемую собственным вектором  $\bar{V}_1$  матрицы  $H_{22}$  и осью  $t$ . Дискретный шаг по масштабам  $\sigma_0, \dots, \sigma_i, \dots, \sigma_n$  в [3] выбирался как и в [7]:  $\sigma_0 = 1, \sigma_i = \sigma_{i-1} \sqrt[4]{2}$ . Для описанной трехмерной процедуры немаксимального подавления достаточно держать в памяти 3 слоя, соответствующих трем последовательным масштабам, шаг за шагом накапливая 3-х мерные локальные максимумы.

Описанный выше метод [3] будет также детектировать еще и особенности типа пятно (blob), которые не являются строго круглыми, и точки контурных линий (edge).

В настоящей работе для отсекация пятен используется векторная обработка линий. После сборки цепочки связанных пикселей (локальных максимумов), они объединяются в граф. Цепочки, не связанные ни с какими другими, в которых число пикселей не велико, отбрасываются. Минимальное количество точек в цепочке определяется пороговой константой, зависящей от ширины линии, что позволяет установить явно минимальную вытянутость пятна, которое считается линией.

Для подавления точек контурных линий в [3] при выполнении немаксимального подавления отбрасываются точки не

прошедшие дополнительный тест на изменение знака скалярного произведения градиентов в соседних точках:

$$(\bar{g}(\bar{x} + \bar{v}_1), \bar{g}(\bar{x} - \bar{v}_1)) < 0. \quad (5)$$

Наши эксперименты показали, что условие (5) может приводить к ошибочному отбрасыванию осевых точек жирной линии в случае разного цвета фона по её разные стороны и в точках сильной извилистости жирной линии. В настояшую работу предлагается модифицированное условие

используется специальная процедура подавления граничных точек. Детектированные на масштабе  $\sigma_i$  точки с координатами  $\bar{X} = (x, y)$ , в которых не происходит смена знака градиента вдоль направления  $V_1$ , остаются, если для них выполняется следующее условие:

$$(\bar{g}(\bar{x} + \sigma_i \bar{v}_1), \bar{g}(\bar{x} - \sigma_i \bar{v}_1)) < T_g \quad (6)$$

где  $\sigma_i$  соответствует текущему масштабу, пороговая величина удовлетворяет условию

$$-1 < T_g < 0, \quad (7)$$

а  $\bar{g} = (L_x, L_y)^T$  – градиент яркости изображения.

#### 4. АДАПТИВНАЯ ЗАЛИВКА

Пусть после предыдущего шага мы получили набор детектированных линий, которые представлены в виде множества точек  $S = \{(\bar{x}_j, \sigma_j, l_j)\}$ , где  $\bar{x}_j$  – координаты центральных точек жирных линий,  $\sigma_j$  – масштаб, на котором была обнаружена точка жирной линии в координатах  $\bar{x}_j$ ,  $l_j$  – величина отклика признака в этой точке (отклик от свертки с лапласианом, взятым на масштабе  $\bar{x}_j$ , нормированный на единицу). По этим точкам построим 3 изображения  $M, I_{\min}, I_{\max}$ . Будем считать, что  $M(\bar{x}) = 1$  если пиксель принадлежит жирным линиям, и  $M(\bar{x}) = 0$  иначе.  $I_{\min}, I_{\max}$  – дополнительные изображения, показывающие допустимые границы изменения интенсивности у жирных линий, по которым будет происходить заливка. Пусть изначально во всех точках  $M(\bar{x}) = 0$ , далее для всех  $\bar{x}_j \in S$  положим

$$\begin{aligned} M(\bar{x}_j) &= 1, \\ I_{\max}(\bar{x}_j) &= I(\bar{x}_j) + 0.5|l_j|, \\ I_{\min}(\bar{x}_j) &= I(\bar{x}_j) - 0.5|l_j|. \end{aligned} \quad (8)$$

Здесь учтено, что как показано в [3], величина Лапласиана на масштабе, соответствующем ширине линии, равна разнице средних яркостей фона и жирной линии (а в случае цветного

изображения по-канальные значения Лапласианов формируют вектор вариации цвета).

Далее будем проводить заливку изображения  $M$ . Предлагаемый алгоритм заливки представляет собой алгоритм flood fill [1] с дополнительными условиями на закраску пикселей.

Рассмотри два соседних пикселя  $\bar{y}_k$  и  $\bar{y}_j$  - такие, что

$$M(\bar{y}_k) = 1, M(\bar{y}_j) = 0. \text{ Если выполняется условие}$$

$$I_{\min}(\bar{y}_k) < I(\bar{y}_j) < I_{\max}(\bar{y}_k), \quad (9)$$

то будем закрашивать пиксель  $\bar{y}_j$  следующим образом:

$$M(\bar{y}_j) = 1,$$

$$I_{\min}(\bar{y}_j) = I_{\min}(\bar{y}_k), \dots$$

$$I_{\max}(\bar{y}_j) = I_{\max}(\bar{y}_k)$$
(10)

а алгоритм адаптивной заливки примет вид:

**Алгоритм AdaptiveFloodFill**( $\bar{x}, \bar{d}$ )

1. **if**  $M(\bar{x}) = 1$  **return**
2. **if**  $NOT(I_{\min}(\bar{x}) < I(\bar{x} + \bar{d}) < I_{\max}(\bar{x}))$  **return**
3.  $M(\bar{x} + \bar{d}) = 1$   
 $I_{\min}(\bar{x} + \bar{d}) = I_{\min}(\bar{x})$   
 $I_{\max}(\bar{x} + \bar{d}) = I_{\max}(\bar{x})$
4. **AdaptiveFloodFill**( $\bar{x} + \bar{d}, (0 \ 1)$ )
5. **AdaptiveFloodFill**( $\bar{x} + \bar{d}, (1 \ 0)$ )
6. **AdaptiveFloodFill**( $\bar{x} + \bar{d}, (0 \ -1)$ )
7. **AdaptiveFloodFill**( $\bar{x} + \bar{d}, (-1 \ 0)$ )

В самом начале алгоритма для всех соседей  $\bar{x}_i + \bar{d}$  точек для всех  $\bar{x}_i : M(\bar{x}_i) \neq 0$  вызываем алгоритм **AdaptiveFloodFill**( $\bar{x}_i, \bar{d}$ ), который отличается от классического [1] строками 2 и 3. В предлагаемом алгоритме дополнительно необходимо обрабатывать ситуацию, когда  $\bar{x} + \bar{d}$  оказывается за границами изображения. Это схема самой простой, не оптимальной, рекурсивной реализации алгоритма заливки. Более быстрый алгоритм можно найти в [1], который также может быть адаптирован для решения поставленной задачи путем введения условия закраски пикселя (9), (10).

## 5. СКЕЛЕТИЗАЦИЯ ЖИРНЫХ ЛИНИЙ

Бинарное изображение  $M$  после предыдущего шага будет сегментировать изображение на пиксели, которые принадлежат жирным линиям, и на фоновые пиксели. В том случае,

когда в точках ветвления жирные линии мало меняют свой цвет по сравнению с фоном, все стыки и точки ветвления будут включены в состав жирных линий, даже если они не были найдены многомасштабным детектором. Далее можно воспользоваться либо непрерывной скелетизацией бинарного изображения [11], и в качестве результата брать скелет как центры жирных линий. Либо воспользоваться пиксельным алгоритмом утончения [9]. В нашем подходе используется последний вариант, т.к. при таком подходе можно оставлять уже найденные многомасштабным детектором линий и добавлять только стыки между ними как результат утончения.

Алгоритм выделения скелета через утончение [9] заключается в последовательном удалении точек, не являющихся осевыми в линиях. Каждая итерация алгоритма [9] состоит из двух проходов по всем пикселям. При каждом проходе точки, которые не являются скелетными, помечаются, но удаляются только после окончания прохода. Вокруг каждой точки  $\bar{x}$  изображения  $M$  берется окно  $3 \times 3$  пикселя  $P_1, P_2, \dots, P_9$  в последовательности, изображенной на Рис. 1, и вычисляются характеристики  $B(\bar{x}) = \sum_{i=2}^9 P_i$  и  $A(\bar{x}) =$  количеству паттернов 01 в последовательности  $P_2, \dots, P_9, P_2$ .

$P_9$ ( $i-1, j-1$ )	$P_2$ ( $i-1, j$ )	$P_3$ ( $i-1, j+1$ )
$P_8$ ( $i, j-1$ )	$P_1$ ( $i, j$ )	$P_4$ ( $i, j+1$ )
$P_7$ ( $i+1, j-1$ )	$P_6$ ( $i+1, j$ )	$P_5$ ( $i+1, j+1$ )

Рисунок 1: Расположение пикселей в окне  $3 \times 3$

На первом проходе пиксель  $\bar{x}$  помечается для удаления из изображения  $M$ , если выполняется все следующие условия:

- a)  $2 \leq B(\bar{x}) \leq 6$
- b)  $A(\bar{x}) = 1$
- c)  $M(P_2) \cdot M(P_4) \cdot M(P_6) = 1$
- d)  $M(P_4) \cdot M(P_6) \cdot M(P_8) = 1$

На втором проходе пункты c) и d) заменяются на следующие:

- c')  $M(P_2) \cdot M(P_4) \cdot M(P_8) = 1$
- d')  $M(P_2) \cdot M(P_6) \cdot M(P_8) = 1$

Все остальное остается без изменений. Итерации делаются до тех пор, пока в двух проходах не будет удалено ни одного пикселя.

Если не удалять пиксели, которые были найдены многомасштабным алгоритмом, тогда все изначально найденные линии останутся, и к ним добавятся только стыки между ними.

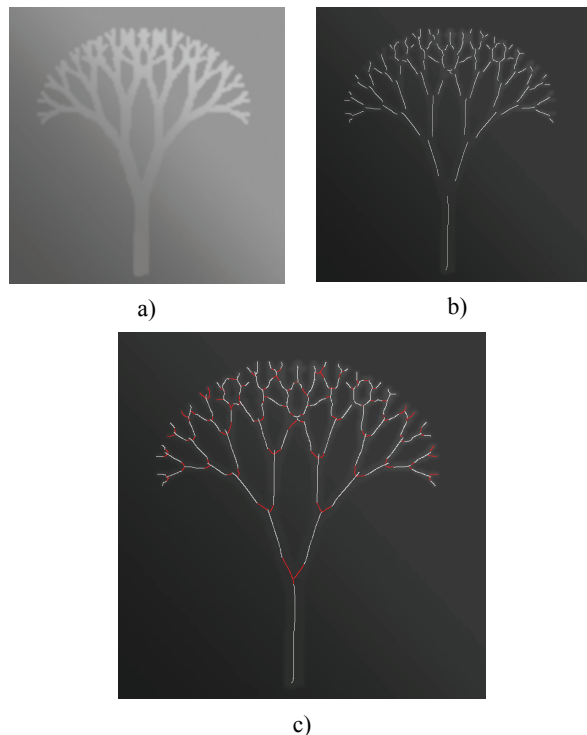
## 6. ДЕТЕКТИРОВАНИЕ ТОЧЕК ВЕТВЛЕНИЯ

После того, как потерянные точки ветвления восстановлены, рассмотрим алгоритм их детектирования. Такой алгоритм был описан в [10] как часть этапа векторизации.

Как и в предыдущем пункте будем делать итеративный проход по всем пикселям, и рассматривать окрестность  $3 \times 3$  каждой точки. Будем считать, что точка  $\bar{X}$  является узлом, если для нее выполняется следующее условие:

$$B(P_1) \geq 5 \vee A(P_1) > 2.$$

Точки помечаются как узел и удаляются вместе с ее окрестностью  $3 \times 3$  из  $M$  сразу при сканировании и запоминаются в другом изображении. Процесс повторяется пока удастся найти хотя бы один узел.



**Рисунок 2: а) Исходное изображение, б) детектирование жирных линий, с) результат сшивки жирных линий**

При сборке пикселей в списки связанных точек [4, 8] определяется их соседство с найденными узлами. Данные векторизуются и хранятся как граф.

## 7. ЗАКЛЮЧЕНИЕ

Предложен алгоритм, позволяющий детектировать жирные линии без разрывов в точках ветвлений и выполнять их векторизацию. Разработан алгоритм адаптивной заливки, который показал хорошие результаты сшивки линий на синтетических и медицинских изображениях, результат работы представлен на Рис. 2. Из недостатков стоит отметить, что адаптивная заливка может не находить точных границ жирных линий. Точные границы планируется находить через минимальный разрез на графе после того, как адаптивная заливка сшила линии одной структуры, что упрощает задачу разметки до двухполюсной задачи с помеченными точками, относящимися к приемнику и источнику максимального потока.

## 8. БЛАГОДАРНОСТИ

Работа выполнена при поддержке гранта РФФИ 13-07-00584.

## 9. ЛИТЕРАТУРА

- [1] S.V. Burtsev, Y.P. Kuzmin. "An efficient flood-filling algorithm" // Computers & graphics, Elsevier, -V. 17, -No 5, -P. 549–561, 1993
- [2] R. Haralick, "Ridges and Valleys on Digital Images" // Computer Vision, Graphics, and Image Processing, -V. 22, -No. 10, 1983
- [3] N. A. Khanina, E. V. Semeikina, D. V. Yurin. "Scale-space color blob and ridge detection" // Pattern Recognition and Image Analysis, -No. 1, -P. 221–227, 2012
- [4] M. Klaiber, L. Rockstroh, Z. Wang, Y. Baroud, S. Simon (2012). "A Memory-Efficient Parallel Single Pass Architecture for Connected Component Labeling of Streamed Images" // Field-Programmable Technology (FPT), International Conference, -P. 159–165, 2012
- [5] T. Lindeberg, "Edge detection and ridge detection with automatic scale selection" // International Journal of Computer Vision, -V. 30, -No. 2, -P. 117–154, 1998
- [6] T. Lindeberg "Scale-Space Theory in Computer Vision" // Kluwer Academic Publishers/Springer, Dordrecht, Netherlands, 1994.
- [7] D.G. Lowe. "Distinctive image features from scale-invariant keypoints" // Int. Journal of Computer Vision, -V. 60, -No. 2, -P. 91–110, 2004
- [8] B. Michael, Dillencourt and Hannan Samet and Markku Tamminen (1992). "A general approach to connected-component labeling for arbitrary image representations" // Journal of the ACM, -V. 39, -N. 2, -P. 253–280, 1992
- [9] T.Y. Zhang, C.Y. Suen "A fast parallel algorithm for thinning digital patterns" // Communications of the ACM, -V. 27, -No. 3, -P. 236–239, 1984.
- [10] А.Е. Левашов, Д.В. Юрин "Система быстрого обнаружения параметрических кривых на серых и цветных изображениях с контролем достоверности" // 21-я международная конференция по компьютерной графике и зрению GraphiCon'2011. Москва, Россия, -с. 212–215, 2011
- [11] Л.М. Местецкий "Непрерывный скелет бинарного растрового изображения" //Труды международной конференции Графикон-1998, Москва, -С. 71-78.

## Об авторах



Левашов Алексей Евгеньевич – аспирант кафедры математической физики, факультета вычислительной математики и кибернетики Московского государственного университета им. М.В.Ломоносова.

[alexeylevashov89@gmail.com](mailto:alexeylevashov89@gmail.com)



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](mailto:yurin_d@inbox.ru)



# Young Scientists School

---

GraphiCon'2013

September 16-20, 2013

Vladivostok, Russia

## Метод вычисления ошибок распознавания частных признаков дактилоскопических изображений

Гудков Владимир Юльевич, Челябинский государственный университет, Челябинск, Россия  
diana@sonda.ru

Дорофеев Константин Андреевич, Челябинский Государственный Университет, Челябинск, Россия  
kostuan1989@mail.ru

### Аннотация

В статье рассматриваются ошибки распознавания частных признаков дактилоскопических изображений. Ошибки первого и второго рода определяют сравнением результатов автоматического распознавания частных признаков с признаками, проставленными экспертом вручную. Для поддержки работы эксперта реализован редактор дактилоскопических изображений.

**Ключевые слова:** дактилоскопия, отпечаток пальца, обработка изображений, частные признаки, шаблон.

### 1. ВВЕДЕНИЕ

В настоящее время методы обработки изображений, как правило, специализированы по степени зашумленности изображений [1-3]. Поэтому, выбор методов обработки зависит от характера изображений. Например, для дактилоскопических изображений (ДИ) качество помехоустойчивого выделения частных признаков узоров [4] оценивают по ошибкам распознавания частных признаков [5, 6].

Целью исследований является решение следующих задач:

- разработка метода вычисления ошибок первого и второго рода распознавания частных признаков ДИ [2];
- вычисление значений ошибок первого и второго рода для реализованного автоматического метода [1] по базе ДИ NIST 14.

### 2. МЕТОД ВЫЧИСЛЕНИЯ ОШИБОК

Метод сравнения результатов автоматического распознавания частных признаков  $A = \{a_i\}$  и эталонных частных признаков  $B = \{b_i\}$ , отмеченных экспертом-криминалистом. Пусть  $B_f$  – число “ложных” частных признаков из множества  $B$ , таких, что ни один из них не находится в заданной окрестности от ближайшего частного признака из множества  $A$ . Ближайший признак находят по евклидовой метрике. Пусть  $A_f$  – число “отсутствующих” частных признаков из множества  $A$ , таких, что ни один из них не находится в заданной окрестности от ближайшего частного признака из множества  $B$ . Тогда ошибку первого рода распознавания частных признаков при сравнении их с эталонными находят в виде

$$\alpha = \frac{B_f}{|B|}$$

Соответственно ошибку второго рода вычисляют как

$$\beta = \frac{A_f}{|A|}$$

Частные признаки имеют признак направления. Проблему анализа направлений частных признаков решают с помощью поля направлений [1, 4]. Для этого выполняют распознавание ориентации локальных окрестностей [3] и поворот этих окрестностей на детектированный угол до совмещения окрестностей.

Частные признаки (окончание и разветвление линии) были отмечены экспертами вручную. Результаты ручной кодировки показаны на рис. 1. Можно считать, что ошибки эксперта распределены по нормальному закону, так как отпечатки пальцев из базы ДИ NIST 14 разные эксперты кодировали независимо. Аналогично автоматическим методом [1] были детектированы и сохранены в шаблоне частные признаки для тех же ДИ. Результаты автоматического распознавания того же ДИ представлены на рис. 2.

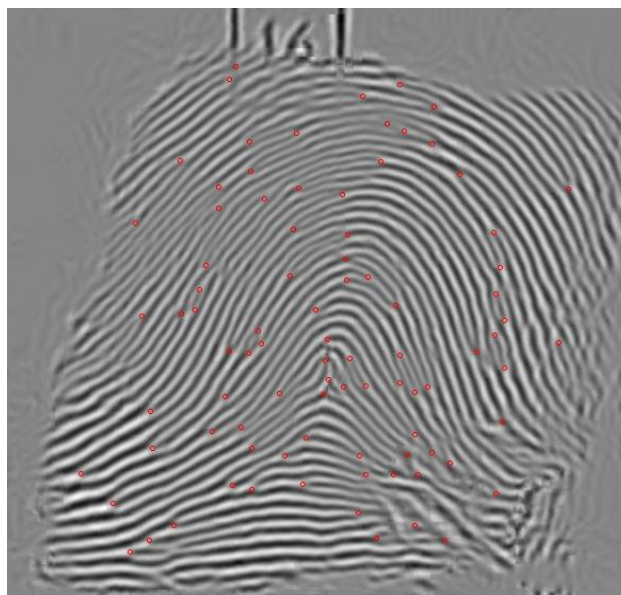


Рис. 1: Эталонные частные признаки



Таблица 1

Окрестность	Ошибки	
	$\alpha$	$\beta$
14x14	0,05	0,07
12x12	0,09	0,1
10x10	0,11	0,15
8x8	0,17	0,35
6x6	0,43	0,51
4x4	0,65	0,75

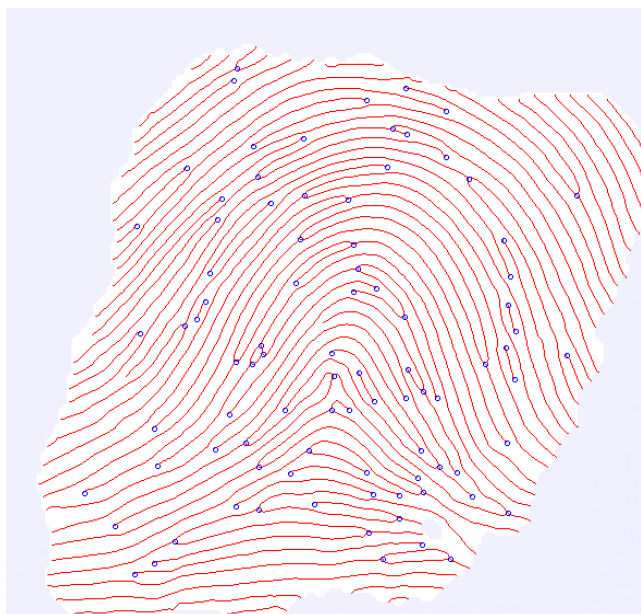


Рис. 2: Детектированные по скелету частные признаки

По результатам сравнения автоматически детектированных и эталонных частных признаков рассчитаны ошибки первого и второго рода, которые представлены в таблице 1. Размер окрестности, задаваемый для поиска ближайшего частного признака из другого множества, заметно влияет на величину ошибки. График зависимости ошибок первого и второго рода от размера окрестности представлен на рис. 3. Ограничение окрестности размером в 14 пикселей связано со средним значением периода папиллярных линий узоров. Превышение этого порога эквивалентно переходу частного признака через линию, что приводит к изменению гребневого счета на 2 единицы и для эксперта-криминалиста недопустимо.

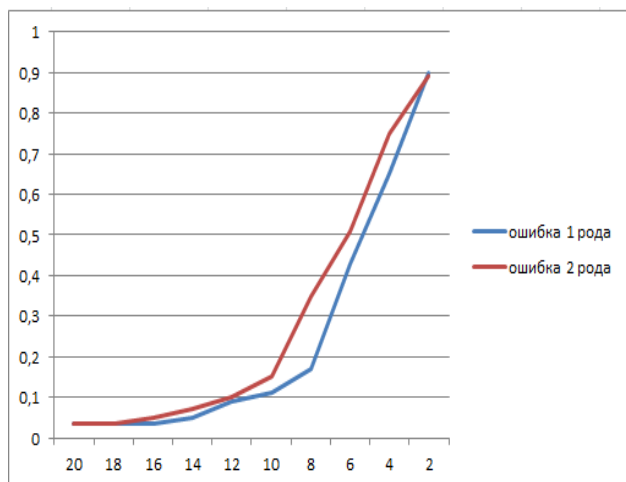


Рис. 3: Зависимость ошибок первого и второго рода от размера окрестности

### 3. ЗАКЛЮЧЕНИЕ

Оценка ошибок распознавания частных признаков ДИ в настоящее время чрезвычайно актуальна [4, 5]. Это определяется сильной зашумленностью ДИ, особенно следов с мест преступлений. Практически не бывает, чтобы автоматически детектированные и эталонные частные признаки полностью совпадали. Это хорошо наблюдается на графике рис. 3. Кроме того, известны трудности, связанные с проблемой выбора признака из подмножества компактно расположенных ложных или истинных частных признаков [1].

В дальнейших исследованиях планируется построение специальной метрики, решающей проблему компактно расположенных частных признаков.

### 4. ССЫЛКИ

- [1] Гудков В.Ю. Методы первой и второй обработки дактилоскопических изображений: монография / В.Ю. Гудков. – Миасс: изд-во “Геотур”, 2009. – 237 с. – ISBN: 978-5-8920-4151-5.
- [2] Гуревич И.Б. Проблемы распознавания изображений: распознавание, классификация, прогноз: математические методы и их применения / И.Б. Гуревич. – Москва: Наука, 1982. – Вып. 1. – 237 с.
- [3] Яне Б. Цифровая обработка изображений / Б. Яне. – Москва: Техносфера, 2007. – 584 с.
- [4] Bazen A.M. Fingerprint identification – feature extraction, matching, and database search / A.M. Bazen, 2002. – 187 с.
- [5] Bolle R.M. Guide to biometrics / R.M. Bolle, J.Y. Connel, S. Pankanti, N.K. Ratha. – New York: Springer-Verlag, 2004. – 368 с.
- [6] Maltoni D. Handbook of fingerprint recognition / D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar. – London: Springer-Verlag, 2009. – 496 p

# Автоматическое преобразование линейных объектов при построении цифровых моделей рельефа (ЦМР)

Шулькин Евгений

Лаборатория ГИС-технологий и моделирования геосистем

Тихоокеанский институт географии ДВО РАН

[evgeny.shulkin@gmail.com](mailto:evgeny.shulkin@gmail.com)

## Аннотация

В данном постерном докладе рассматривается проблема улучшения качества цифровой модели рельефа в условиях недостатка информации о высоте. Предложена реализация метода восстановления z-метрики структурных линий (талъвегов, хребтов и пр.) на основе системы изолиний и отметок высот.

**Ключевые слова:** Цифровая модель рельефа, структурные линии, z-метрика, ArcGIS, Python.

## 1. ВВЕДЕНИЕ

Для решения ряда прикладных задач необходимо уметь создавать высокоточные и гидрологически корректные цифровые модели рельефа (ЦМР). Существует два основных метода получения ЦМР – с помощью стереопары, а также используя данные, которые могут быть получены при обработке картографических источников (отметки высот, изолинии, структурные линии и др.).

Недостаток объема информации о высоте в различных областях карты отрицательно влияет на качество итоговой модели рельефа. Частично этот недостаток можно исправить, восстановив данные о z-метрике структурных линий (в частности, талъвегов, хребтов и пр.). Картографические источники не содержат эту информацию в явном виде, но позволяют ее вычислить на основе системы изолиний и выбранной модели интерполяции.

Цель данной работы заключается в разработке алгоритма нахождения z-метрики структурных линий на основе имеющейся линейной и точечной картографической информации.

## 2. РЕАЛИЗАЦИЯ

В качестве средства реализации был выбран скриптовый язык Python, с целью последующей интеграции в ArcGIS Toolbox. В ArcGIS 9.2 появилась поддержка нового типа данных Terrain Dataset для эффективного представления ландшафтов, на основе нерегулярных триангулированных сетей, которые генерируются с помощью разнородных пространственных данных. Такой подход позволяет оперировать огромными объемами данных при построении ЦМР.

Скрипт интегрирует процесс нахождения структурных линий в общий комплекс задач, связанных с построением цифровых моделей рельефа в ArcGIS, и, в конечном счете, значительно повышает качество итоговой ЦМР.

Результат работы скрипта можно продемонстрировать, сравнив качество ЦМР, построенной с использованием

структурных линий и без них. Использование скрипта, в данном случае, существенно экономит время расчета z-метрики структурных линий.



Рис. 1. Нет структурных линий (явно видны артефакты в долинах водотоков)



Рис. 2. С использованием структурных линий

## 3. ССЫЛКИ

- [1] ArcGIS Help 10.1 – What is a terrain dataset?  
[http://resources.arcgis.com/en/help/main/10.1/index.html#/What is a terrain dataset/005v00000002000000/](http://resources.arcgis.com/en/help/main/10.1/index.html#/What%20is%20a%20terrain%20dataset/005v00000002000000/)
- [2] Paul A. Zandbergen. *Python Scripting for ArcGIS*.

## Об авторах

Шулькин Евгений – аспирант, инженер лаборатории ГИС-технологий и моделирования геосистем ТИГ ДВО РАН. Его адрес: [evgeny.shulkin@gmail.com](mailto:evgeny.shulkin@gmail.com).

# Особенности использования регистровой памяти GPU в OpenCL на примере текстурного компрессора

Илья Перминов

Факультет компьютерных технологий и управления

Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, Санкт-Петербург, Россия  
studentikispam@gmail.com

## Аннотация

В статье представлена модификация алгоритма текстурной компрессии LSDxt [2] для работы на графическом процессоре. Кратко описана реализация алгоритма с использованием локальной разделяемой памяти. Рассмотрены проблемы, возникающие при реализации алгоритмов с использованием только регистровой памяти. Приведены результаты тестирования производительности.

**Ключевые слова:** OpenCL, GPGPU, сжатие текстур, производительность, регистровая память

## 1. ВВЕДЕНИЕ

Алгоритмы и задачи, обладающие высокой степенью внутреннего параллелизма, при выполнении на графическом процессоре могут иметь существенно большее быстродействие по сравнению с традиционными процессорами. Но для получения максимальной производительности необходимо существенно модифицировать алгоритм для эффективного использования ресурсов GPU. У разных моделей GPU архитектура может сильно различаться, но основные рекомендации состоят в следующем:

- Разбиение тредов на группы, кратные размеру аппаратно выполняемых групп (warp для GPU компании Nvidia или wavefront для AMD)
- Работа с глобальной памятью с учетом её блочной организации. Если все треды одной аппаратной группы обращаются к одному и тому же блоку памяти, то этот запрос может быть удовлетворён за одно обращение в память.
- Уменьшение количества конфликтов банков памяти при использовании локальной разделяемой памяти. Если все адреса, по которым обращаются в память треды одной аппаратной группы, относятся к разным банкам памяти, то такой запрос может быть выполнен за одно обращение в память.
- Аккуратное использование условных переходов. Желательно, чтобы все треды одной аппаратной группы выбирали один и тот же переход. Иначе обе ветви условия будут выполняться последовательно.
- Поддержание высокого значения показателя загрузки тредов (Occupancy), чтобы скрыть задержки при обращении к памяти. Если ресурсы мультипроцессора позволяют разместить на нём несколько групп тредов, то во время ожидания

операций с памятью для одной группы, мультипроцессор может исполнять инструкции другой готовой группы.

Однако наибольшим быстродействием доступа обладают именно регистры и для некоторых алгоритмов достаточно объема регистровой памяти. Тем не менее, такой подход не всегда может увеличить производительность.

## 2. СЖАТИЕ ТЕКСТУР

Сжатие текстур используется для экономии пропускной способности памяти, так как текстуры передаются в графический процессор в сжатом виде и распаковываются аппаратно между блоками кэш-памяти L1 и L2. Так же это позволяет уменьшить объем памяти, занимаемый текстурами. Поскольку в 3D-графике крайне важна эффективность произвольного доступа к отдельным текселям, все современные форматы сжатия текстур обладают фиксированным уровнем сжатия и, следовательно, осуществляют сжатие с потерями. Как правило, это блочные кодеки.

Наиболее распространенным форматом является DXT/S3TC и его вариации BC1-BC5. При использовании DXT1 (Рис.1) для каждого блока исходного изображения размером 4x4 сохраняется 2 ключевых цвета в формате RGB565 и таблица индексов, определяющая в какой пропорции смешивать ключевые цвета при восстановлении блока. Кроме этого, взаимным расположением ключевых цветов неявно кодируется еще один бит, используемый при распаковке.

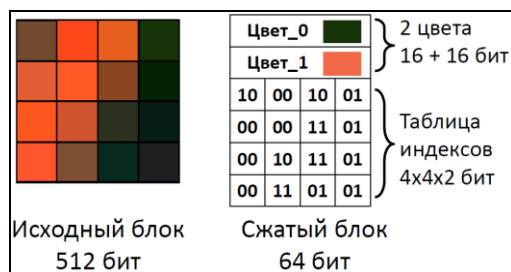


Рис 1: Формат блока DXT1/BC1.

Распаковка такого блока довольно проста, но качество восстановленного изображения сильно зависит от выбора ключевых цветов. Компрессия с высоким качеством очень ресурсоёмка и, как правило, для неё используются методы последовательного перебора.

### 3. АЛГОРИТМ СЖАТИЯ LSDXT

Алгоритм сжатия LSDxt позволяет варьировать скорость и качество сжатия и состоит из трёх основных шагов:

1. Получение ключевых цветов. Для этого отдельно в каждом цветовом канале берутся уникальные значения цвета, и к ним применяется линейная регрессия.
2. Улучшение результата. Воспроизводится сжатый блок, и линейная регрессия применяется к ошибкам отдельно в каждом цветовом канале, что позволяет уточнить результат.
3. Перебор значений в малой окрестности полученных результатов с целью минимизации ошибки.

Шаг 2 является итеративным. Чем больше итераций проводится на шаге 2 и более широкая окрестность выбирается на шаге 3, тем выше качество итогового изображения.

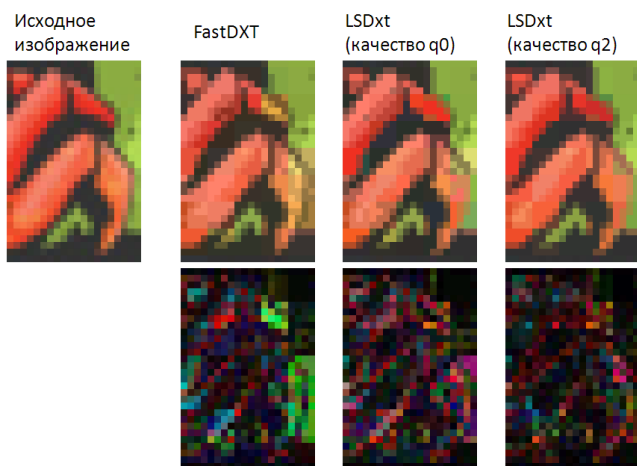


Рис 2: Пример сжатия различными кодеками (верхний ряд) и ошибки сжатия (нижний ряд).

### 4. АЛГОРИТМ С ИСПОЛЬЗОВАНИЕМ ЛОКАЛЬНОЙ ПАМЯТИ

Рабочая группа (workgroup) была выбрана размером 64 так как она кратна и размеру warp (32) и размеру wavefront (64). Каждый блок 4x4 исходного изображения может обрабатываться независимо, и можно было бы использовать отдельный тред (workitem) для каждого блока. Но это потребовало бы выделения 32 KiB локальной памяти на рабочую группу для хранения значений цветов и ошибок, что означало бы очень низкую Occupancy и простой вычислительных ресурсов на каждой операции с памятью. Поэтому для сжатия используется один тред на каждый текстель. То есть одна рабочая группа сжимает 4 блока, что уменьшает требования к локальной памяти в 16 раз. Такой подход позволяет вычислять ошибку каждого текстеля параллельно и использовать параллельную сортировку на шаге 1.

Так как шаги алгоритма независимы, используются отдельные ядра (kernels) для каждого шага. Это делает необходимым загрузку данных из глобальной памяти в

локальную 3 раза (на каждом шаге), но затрачиваемое на это время более чем на два порядка меньше времени работы самого алгоритма.

Размещение данных в локальной памяти в естественном порядке (Рис.3 а) приводит к конфликтам банков памяти при одновременном доступе нескольких тредов. Ликвидировать конфликты можно используя варианты расположения b или c.

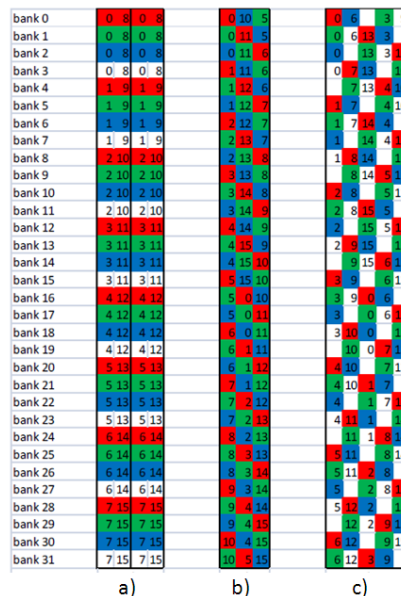


Рис 3: Расположение данных в локальной памяти с конфликтом банков памяти (a) и без (b, c)

При использовании прочих оптимизаций, такой вариант компрессора способен обеспечить 30-кратный прирост производительности по сравнению с CPU вариантом (Табл. 3).

### 5. ИСПОЛЬЗОВАНИЕ РЕГИСТРОВ

Дальнейшее увеличение быстродействия возможно при использовании только регистровой памяти, потому что она обладает наибольшей пропускной способностью и наименьшей задержкой (Табл. 1).

	Global	Local	Reg
Peak Bandwidth	264 Gb/s	3789 Gb/s	22733 Gb/s
Peak Bandwidth /Stream Core	~0.14 bytes/cycle	8 bytes/cycle	12 bytes/cycle
Size	3Gb	64Kb / CU	256 Kb /CU

Таблица 1: Характеристики различных типов памяти AMD Radeon HD 7970 [3].

Но использование только регистровой памяти затрудняется по следующим причинам:

- Невозможность обмена данными между тредами внутри рабочей группы без использования глобальной или локальной разделяемой памяти.

- Как следствие, необходимость использовать один тред на один блок, что значительно увеличивает расход регистров и снижает Ossurancy
- Увеличивается размер исполняемого бинарного кода
- Увеличивается степень дивергенции тредов
- Большие накладные расходы при использовании массивов, если индекс элемента неизвестен на момент компиляции

Фактически, это может привести даже к снижению производительности.

### 5.1 Обмен данными между тредами

Современные графические процессоры имеют аппаратную возможность обмена данными между тредами без использования глобальной или локальной разделяемой памяти. Примерами могут служить инструкция DS\_SWIZZLE\_32 для архитектуры AMD GCN (Southern Islands) [5] или инструкция SHFL для архитектуры NVIDIA SM\_30 [7]. Однако в OpenCL 1.2 такая возможность не предусмотрена и подобный обмен не может быть использован.

### 5.2 Уровень Ossurancy

Использование одного треда на блок заставляет держать целиком весь блок и весь массив с ошибками в регистрах данного треда, что значительно увеличивает расход регистров на треда и на всю рабочую группу, снижая тем самым уровень Ossurancy. Но высокий уровень Ossurancy используется только, чтобы скрыть задержку при обращении к памяти. На графических процессорах NVidia наличие нескольких warp'ов на мультипроцессоре так же позволяет скрыть задержку, вызванную зависимостями инструкций по данным [6].

Поэтому в подобной ситуации уровень Ossurancy в диапазоне 10%-20% может быть оптимальным[1] и дальнейший рост не приводит к увеличению производительности.

### 5.3 Размер исполняемого кода

Так как теперь отдельный тред выполняет большее количество работы, размер исполняемого кода может ощутимо увеличиться. Производительность может снизиться, если размер бинарного кода превысит размер кеша инструкций GPU. Для архитектуры AMD GCN он составляет 32 KB [4]. И хотя, как правило, это не является проблемой, для уменьшения размеров кода можно:

- разбить ядро (kernel) на несколько более мелких ядер и выполнить их последовательно
- уменьшить unroll-factor или полностью убрать директивы разворачивания циклов #pragma unroll для больших циклов. В некоторых ситуациях это может увеличить производительность

### 5.4 Дивергенция

Так как все треды сжимают отдельные блоки, увеличивается шанс, что разные треды одной аппаратной группы пойдут по разным ветвям условного перехода. Впрочем, рекомендации по оптимизации ветвлений не отличаются от обычного случая.

Например, код вида:

```
If (A>B) {
C += D;
} else {
C -= D;}
```

Следует заменять на:

```
int factor = (A>B) ? 1:-1;
C += factor*D;
```

Так как во втором случае не используются инструкции условных переходов.

### 5.5 Адресация элементов массива

Компилятор старается размещать массивы, объявленные без ключевого слова \_\_global, в регистрах. Если все обращения идут по заранее известным константным индексам, то такой массив может рассматриваться просто как набор переменных, которые будут отображаться на фиксированные регистры. И хотя в архитектурах GPU имеются некоторые механизмы для адресации регистров (например, аппаратный регистр M0 в AMD GCN [5]), адресация по неизвестному индексу выполняется гораздо менее эффективно.

Поэтому, если в теле программы осуществляется доступ к элементам массива по неизвестному на момент компиляции индексу, то в большинстве случаев, компилятор будет вынужден разместить такой массив в глобальной или локальной памяти или использовать большое количество дополнительных инструкций. Особенно сильно это сказывается на производительности в архитектуре AMD GCN. О наличии подобных проблем можно судить по результатам профилирования и показаниям аппаратных счётчиков производительности по следующим признакам:

- Наличие memory stall при невысокой интенсивности обмена данными с памятью и блочному обмену с глобальной памятью и отсутствию конфликтов банков локальной памяти.
- Общий объём записанных и считанных из памяти данных значительно превышает ожидаемое значение. Одно обращение в исходном коде к массиву может порождать несколько десятков KiB трафика с памятью.
- В некоторых ситуациях «раздувание» бинарного кода.

На шаге номер 2 рассматриваемого алгоритма сжатия используется 2 таких массива. Для каждого текселя блока вычисляется, к какому из 4-х цветов, доступных в сжатом блоке, он ближе всего (bestInd), и в элементы массивов used и Bucket заносится информация об ошибке:

```
used[bestInd] = 1;
Bucket[bestInd] += MinError;
```

Так как bestInd не известен на момент компиляции, массивы used и Bucket размещаются в глобальной памяти. Но малый размер данных массивов позволяет произвести перебор по всем индексам:

```
#pragma unroll
for(int i=0; i<4; i++){
char is = (bestInd == i) ? 1 : 0;
used[i] |= is;
Bucket[i] += is * MinError;
}
```

Данная оптимизация позволила значительно ускорить выполнение шага 2 (Табл. 2).

	AMD Radeon HD7970	NVidia GTX560
Local Memory	6.05 ms	18.1 ms
Registers, non compile-time constants indexing	12.37 ms	4.34 ms
Registers only, compile-time constants indexing	0.86 ms	3.88 ms

**Таблица 2:** Время выполнения шага 2 для тестовой текстуры 512x512

## 6. РЕЗУЛЬТАТЫ

Результаты тестирования производительности рассматриваемого алгоритма сжатия при выполнении на CPU и GPU представлены в Табл.3. Для запуска на CPU использовался оригинальный многопоточный компрессор от автора алгоритма [2]. Тестирование проводилось на максимальных настройках качества на следующем оборудовании:

- CPU: Intel Core i3 660 (2 cores, 4 threads, 3.33 GHz)
- GPU1: Nvidia GeForce GTX560 (336 CUDA cores, 1.6 GHz)
- GPU2: AMD HD7970 (2048 PE cores, 925 MHz)

	Time	Speedup
Intel Core i3 660	1523 s	-
NVidia GTX560 (local memory)	50.8 s	30 x
NVidia GTX560 (registers)	30.7 s	50 x
AMD HD7970 (local memory)	16.4 s	92 x
AMD HD7970 (registers)	6.8 s	224 x

**Таблица 3:** Скорость сжатия тестовой текстуры 512x512 на максимальном качестве

## 7. ЗАКЛЮЧЕНИЕ

Несмотря на то, что использование только регистровой памяти приводит к существенным проблемам при обмене данными между тредами и сложностям при работе с массивами, такой подход может дать ощутимый выигрыш по скорости.

Однако даже для алгоритмов, активно использующих локальную или глобальную память, использование массивов, относящихся к private области памяти, может служить причиной снижения производительности. Зачастую это временные массивы небольшой размерности. Для повышения быстродействия в подобных ситуациях можно попробовать использовать подход, описанный в разделе 5.5, или другой способ, уменьшающий количество обращений по индексам неизвестным на момент компиляции.

## 8. ССЫЛКИ

- [1] Volkov V., *Better Performance at Lower Occupancy*, 2010, [http://www.nvidia.com/content/GTC-2010/pdfs/2238\\_GTC2010.pdf](http://www.nvidia.com/content/GTC-2010/pdfs/2238_GTC2010.pdf)
- [2] [Release] LSDxt DXT Compressor, 2012, <http://lspiroengine.com/?p=516>
- [3] AMD Accelerated Parallel Processing OpenCL Programming Guide, 2012, [http://developer.amd.com/download/AMD\\_Accelerated\\_Parallel\\_Processing\\_OpenCL\\_Programming\\_Guide.pdf](http://developer.amd.com/download/AMD_Accelerated_Parallel_Processing_OpenCL_Programming_Guide.pdf)
- [4] AMD Graphics Cores Next (GCN) Architecture, 2012, [http://www.amd.com/la/Documents/GCN\\_Architecture\\_whit epaper.pdf](http://www.amd.com/la/Documents/GCN_Architecture_whit epaper.pdf)
- [5] AMD Southern Islands Series Instruction Set Architecture, 2012, [http://developer.amd.com/wordpress/media/2012/12/AMD\\_Southern\\_Islands\\_Instruction\\_Set\\_Architecture.pdf](http://developer.amd.com/wordpress/media/2012/12/AMD_Southern_Islands_Instruction_Set_Architecture.pdf)
- [6] NVidia OpenCL Best Practices Guide, 2011
- [7] NVidia Parallel Thread Execution ISA version 3.0, 2012

## Об авторе

Илья Перминов – аспирант кафедры вычислительной техники в НИУ ИТМО (Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики)

Email: studentikispam@gmail.com

# Навигация с применением RANSAC фильтрации точечных особенностей

Машенцев Владимир Юрьевич, Институт автоматки и процессов управления ДВО РАН, Владивосток, Россия,  
v.mashentsev@gmail.com

## Аннотация

Предложен метод визуальной навигации подводного аппарата по потоку видеоизображений, синхронно захватываемому двумя камерами. Метод основан на использовании оптического потока, прослеживании точечных особенностей среды и обработке соответствующих 3D облаков с применением RANSAC алгоритма. Приведены результаты вычислительных экспериментов на модельных данных.

**Ключевые слова:** Визуальная навигация, оптический поток, RANSAC, точечные особенности среды, 3D облака.

## 1. ВВЕДЕНИЕ

Задача восстановления положения подводного аппарата (ПА) на основе характерных особенностей среды, получаемых с изображений с фото или видеокамер, установленных на аппарате, активно исследуется на протяжении длительного времени. На данный момент, предложены алгоритмы с использованием различных техник, относящихся как к обработке исходных данных (оптический поток, характерные особенности среды, выделение объектов и т.д.), так и непосредственно к решению задачи навигации, основанных на оптимизационных методах, робастных схемах семейства «генерация-тестирование», преобразовании Хафа и др.

Извлечение и последующее отслеживание характерных особенностей среды накладывает некоторые ограничения на используемые методы. Одной из причин таких ограничений является наличие ошибок в отслеживании особенностей от кадра к кадру, причём ошибки могут иметь разную природу – ошибка локализации и ошибка измерения. Ошибка измерения (особенность на кадрах, полученных в разные моменты времени, отслежена верно, но некоторые параметры вычислены с ошибкой) позволяет нам использовать такие особенности при решении, однако ошибка локализации (некорректное сопоставление между кадрами – выбросы – outliers) приводит к неверному решению.

Одной из распространённых и успешно применяемых техник, которые позволяют отсеять «плохие» особенности, являются методы, основанные на устойчивых схемах типа «генерация-тестирование». Первой работой, в которой был предложен алгоритм RANSAC (Random Sample Consensus), является работа [1]. Принцип работы данного алгоритма заключается в итеративной выборке минимального подмножества элементов, необходимого для вычисления параметров модели, оценке данных параметров с помощью некоторого критерия и по завершению всех итераций – выборе наилучшего набора параметров и соответствующего им подмножества. Данный метод снижал большую популярность благодаря сочетанию простоты и эффективности, позволяющему даже при большом количестве выбросов давать приемлемые результаты. С тех пор данный метод регулярно используется для решения различных задач, где существует необходимость поиска

«хороших» подмножеств из всего набора входных данных, наиболее соответствующий некоторой модели.

Для решения задачи навигации RANSAC использовался в различных работах. Например, в работе [2] было предложено улучшение работы RANSAC алгоритма путём предварительной генерации фиксированного числа гипотез и выбора наиболее значимых на ограниченном наборе данных. Такой подход позволяет увеличить производительность и может применяться в системах реального времени, однако потенциально может давать худшие результаты по сравнению с классическим подходом в случае, когда отношение «хороших» данных ко всему набору достаточно большое. В работах [3] и [4] использовались более сложные зрительные подсистемы (LIDAR устройства, трифокальные установки) для извлечения особенностей использовались. В работе [5] вводились ограничения на неголономность механической системы, что позволяло уменьшить размер пробных подмножеств до одной точки. Также RANSAC схемы применялись как часть более сложной системы с использованием фильтра Калмана [4].

Несмотря на то, что во многих работах были получены приемлемые результаты, они имеют под собой ряд ограничений, а значит, своей актуальности проблема не потеряла и требует исследования новых подходов в зависимости от практического приложения.

В данной работе предложен алгоритм восстановления положения ПА по последовательности изображений, получаемых со стереопары, и предназначенный для работы в режиме реального времени. В основе алгоритма лежит модифицированная версия метода RANSAC с использованием 3D облаков особенностей. Для извлечения и прослеживания 2D множества особенностей среды используется KLT-трекер.

## 2. ПОСТАНОВКА ПРОБЛЕМЫ

Ставится задача восстановления траектории ПА с использованием видеоинформации, поступающей с бортовых камер. Одним из возможных подходов к решению данной задачи является применение алгоритмов вычисления оптического потока с использованием характерных особенностей среды (features).

Ключом к решению поставленной задачи является получение преобразования между 3D облаками особенностей в моменты времени  $t_i$  и  $t_j$ , и использование полученного преобразования для вычисления координат ПА. Для обнаружения и отслеживания особенностей используется алгоритм KLT-трекер, с помощью которого можно получать множество особенностей и отслеживать его на последовательности изображений. 3D координаты особенностей вычисляются с использованием геометрии стереопары. Вопросы, связанные с работой стереопары, в данной работе не рассматриваются.

Итак, ставится задача нахождения значений дискретно заданной функции  $F(t_i)$ .

$$F(t_i) = \vec{p}_i = M_i \vec{p}_0,$$

где  $p_i$  – положение ПА в  $i$ -й момент времени,  $p_0$  – положение ПА в начальный момент времени,  $M_i$  – преобразование, связывающее положение ПА в момент времени  $i$ , и начальное положение.

Положение  $\vec{p}_0$  берётся положением первого кадра. Для получения преобразования в произвольной мировой системе координат (МСК) необходимо знание преобразования из начала координат МСК в положение  $\vec{p}_0$ .

Траектория ПА рассматривается в виде последовательности серий изображений.

$$t_0 < \dots < t_i < t_j < \dots < t_N, \text{ где } i < j.$$

Матрица преобразования  $M_i$  получается путем комбинирования преобразований каждой серии.  $M_i = M_{10}M_{21}M_{32} \dots M_{ji}$ . Каждая из матриц  $M_{ji}$  находится способом, предложенным в работе [6] и связывает 3D облака особенностей  $C_i, C_j$ .

$$C_i = M_{ji}C_j,$$

Поскольку особенности, отслеживаемые с помощью KLT-трекера зашумляются, то, по сути, мы имеем  $C'_j$  – облако особенностей в моменты времени  $t_j$ , но вычисленные с ошибкой (особенности на первом кадре каждой серии будем считать идеальными).

$$C_i = M'_{ji}C'_j + E, \text{ где } M'_{ji} = M_{j,i} + \theta, C'_j = C_j + \varphi$$

Таким образом, необходимо найти преобразование  $M'_{ji}$  наиболее близкое к идеальному  $M_{ji}$ , т.е. с минимальной ошибкой  $\theta$ .

### 3. ВЫЧИСЛЕНИЕ ОПТИЧЕСКОГО ПОТОКА

В данной работе для вычисления оптического потока используется KLT-трекер, предложенный в работе [7, 8]. С помощью данного алгоритма производится поиск (извлечение) и прослеживание характерных особенностей среды на изображениях с левых камер стереопары. Прослеживание осуществляется только в пределах серии кадров. В работе исследовались два возможных способа использования KLT-трекера применительно к последовательности изображений. В пределах одной серии трекер можно запускать:

1. на каждом кадре серии (период выборки 1 кадр);
2. выборочно (период выборки 2, 4 и т.д. кадра).

Преимуществом покадрового подхода является минимизация вероятности ошибки локализации, поскольку отслеживание особенностей происходит на малых смещениях. Однако недостатком является время обработки всех кадров серии, которое может превысить реальное время движения ПА.

При выборочном подходе количество запусков трекера снижается и, соответственно, количество ресурсоёмких вычислений снижается, поэтому данный подход более простой для выполнения в реальном времени, однако при больших значениях смещений проекций особенностей на изображениях повышается вероятность появления выбросов и уменьшается количество особенностей в облаках, что будет отрицательно сказываться на дальнейших результатах.

### 4. ПРЕДЛАГАЕМЫЙ ПОДХОД

Поскольку взаимно-однозначное соответствие особенностей облаков  $C_i, C'_j$  известно (результат работы KLT-трекера), то для получения преобразования  $M'_{ji}$  с минимальной ошибкой  $\theta$  необходимо исключить из  $C'_j$  особенности-выбросы.

Для решения проблемы поиска «хороших» подмножеств  $G_i \subseteq C'_i, G'_j \subseteq C'_j$  в данной работе используется алгоритм RANSAC. Работа данного алгоритма основывается на случайных выборках из исходных множеств, построение модели по выборкам и оценке полученной модели относительно исходных множеств.

Под моделью будем понимать преобразование  $M'_{ji}$ . Критерием оценки модели будем считать среднее отклонение особенностей 3D облака  $G_i$  и особенностей 3D облака  $G'_j$ , преобразованного в соответствии с моделью.

Кратко схему алгоритма можно записать в следующем виде:

1. На вход подаются 3D облака  $C_i, C'_j$ .
2. Производится выборка «пробных» подмножеств (облака-гипотезы) соответствующих особенностей  $X_i, X_j$  из  $C_i, C'_j$  соответственно.
3. Вычисляется модель-гипотеза  $H: X_i = HX_j$  методом, описанным в работе [6].
4. Вычисляется оценка соответствия особенностей модели: на основе  $H$  вычисляется облако  $C'_i = HC'_j$  и проверяется соответствие особенностей  $C_i, C'_i$  гипотезе на основе функции оценки  $\|c_{i,l} - c'_{i,l}\|_{L_2} < \xi, c_{i,l} \in C_i, c'_{i,l} \in C'_i, l = \overline{1, A}$ , где  $\xi$  – порог задаваемый пользователем,  $A = |C_i - C'_i|$ . В результате все особенности помечаются либо «хорошими»:  $c_{i,l} \in G_i, c'_{i,l} \in G'_i$ , либо выбросами:  $c_{i,l} \notin G_i, c'_{i,l} \notin G'_i$ .
5. На основе количества особенностей в облаках  $G_i, G'_i$  и оценки критерия  $\sigma = \frac{1}{B} \sum \|g_{i,l} - g'_{i,l}\|_{L_2}, g_{i,l} \in G_i, g'_{i,l} \in G'_i, l = \overline{1, B}, B = |G_i - G'_i|$  гипотеза помечается лучшей и вычисляется обновленная модель по всем особенностям  $G_i, G'_j$ , в противном случае отбрасывается.
6. Повторяем пункты 2-5  $k$  раз,  $k$  – количество итераций, задаваемое пользователем.

После выполнения данного алгоритма на выходе получаем наилучшую модель  $M'_{ji}$  с минимальной ошибкой  $\theta$ , полученную за ограниченное время.

### 5. ПАРАМЕТРЫ RANSAC АЛГОРИТМА

Слабым местом RANSAC подхода в приложениях реального времени, является эвристический поиск параметров количества итераций и количества точек в облаке, на основании которого мы будем строить модель-гипотезу.

При первоначальном анализе проблемы возникают следующие рассуждения: чтобы уменьшить время работы алгоритма необходимо уменьшить общее количество всевозможных выборок, что должно позволить получить оптимальный по времени результат, и нехитрые рассуждения приводят к следующему факту – для того, чтобы уменьшить количество выборок, нужно брать облако минимального размера [1]. Однако даже при минимальном количестве особенностей в облаке, число всевозможных подмножеств достаточно большое, чтобы можно было все их «попробовать» в режиме реального времени. А также немаловажен тот факт, что чем меньше точек в облаке, тем больший вклад вносят их погрешности в модель-гипотезу [9]. С другой стороны при увеличении размера облака повышается вероятность включения точек с большой



погрешностью и их значительный вклад в ошибку модели-гипотезы. Многочисленные эксперименты показали корректность данных предположений. При 10 независимых запусков, размер облака в 3 точки давал неустойчивые результаты вне зависимости от допустимого количества итераций. Однако при увеличении размера облака расхождение результатов всех запусков уменьшалось при допустимом количестве итераций.

На конечный результат также влияет выбор порогового значения  $\xi$ . На первый взгляд, для получения лучшей модели, нужно уменьшать порог  $\xi$  (в идеальном случае мы можем получить  $|\sigma - \varepsilon| = 0$  для любого бесконечно малого  $\varepsilon > 0$ ), однако, такое утверждение неверно и полученные результаты это подтверждают.

Проблема заключается в том, что облако  $C'_i$  получено с ошибкой. Распишем функцию оценки (далее не будем указывать норму, будем подразумевать  $L2$ ):

$$\|C_i - C'_i\| = \|C_i - HC'_j\| = \|C_i - HC_j - H\varphi\| < \xi$$

$\Rightarrow H = M_{ji}$ , где  $M_{ji}$  – идеальная матрица. Перепишем полученную оценку:

$$\|C_i - M_{ji}C_j - M_{ji}\varphi\| = \|M_{ji}\varphi\| < \xi$$

Поскольку шум  $\varphi$  в рамках одного момента времени является постоянным  $\varphi = Const \neq 0$ , то для выполнения оценки, при изменении порога  $\xi$ , нужно изменять  $M_{ji}$ , отдаляясь от идеальной, что противоречит исходной задаче.

Малые значения  $\xi$  отсеивают большую часть точек, что приводит к уменьшению размера 3D облака и увеличению значения вклада погрешности каждой особенности в модель-гипотезу. Однако большие значений порога  $\xi$  допускают наличие особенностей-выбросов в 3D облаке или особенности с большой погрешностью, которые дадут значительный вклад в смещение модели-гипотезы.

## 6. РЕЗУЛЬТАТЫ

При проведении экспериментов были рассмотрены различные комбинации параметров таких как: размер облаков-гипотез  $S$ , порог  $\xi$ , период выборки кадров  $T$ , количество итераций алгоритма  $RANSAC$   $k$ . Каждый набор параметров запускался 10 раз, затем считалось среднее значение для каждого момента времени. Во всех запусках длина серии была принята в 16 кадров. Эксперименты проводились на процессоре Intel Core2Duo E6550 на сцене с «песчаной» текстурой (рис. 3), с перепадами расстояния от центров проекций изображений до поверхности сцены 2-10 метров и размерами изображений 600x400 точек.

В ходе экспериментов в качестве значений  $\xi$  брались 0.01, 0.03, 0.05, 0.10. Значения 0.01, 0.03 давали достаточно большой разброс результатов, а количество «хороших» особенностей могло уменьшаться в несколько раз относительно общего количества. При значениях 0.05 и 0.10 результаты получались лучше, однако вследствие того, что при значении 0.10 количество особенностей в «хороших» облаках практически совпадало с общим числом особенностей (попадали особенности с большой ошибкой измерения), результаты при  $\xi = 0.05$ м были наилучшими.

Параметры  $k$  и  $S$  явным образом связаны между собой. В качестве параметра  $S$  использовались различные значения: от минимального  $S = 3$ , до  $S = 50$ , а параметр  $k$  непосредственно отвечает за соответствие времени работы алгоритма в реальном времени. При  $S = 3$  работа алгоритма оказывалась неустойчивой даже при максимальных значениях параметра

$k$ , однако, тенденция к уменьшению разброса результатов при увеличении параметра  $S$  позволила уменьшать параметр  $k$ . Наилучшие результаты были достигнуты при значениях  $k = 100, S = 50$ .

Для параметра  $T$  также пробоваались несколько значений 1, 4, 8 (рис. 1). При значении  $T = 8$  результаты получались наихудшими, поскольку общее количество точек в облаках уменьшалось, и значительно возрастало количество особенностей с ошибкой локализации. Значения  $T = 1$  и  $T = 4$  дали интересные результаты: при значении  $T = 4$  показатели улучшились по сравнению с  $T = 1$ . Основными факторами, которые повлияли на поведение алгоритма в данной ситуации, являются характер сцены и движения ПА. При минимальном перемещении особенность может попадать в тот же пиксел изображения камеры, что и на предыдущем шаге, и таким образом давать дополнительную погрешность. При  $T = 4$  особенности становятся более «различимы» на изображениях, но при этом расстояние между одной и той же особенностью на кадрах с соседних шагов находится в рамках корректной работы трекара.

	T = 1		T = 4	
	S = 50	Без RANSAC	S = 50	Без RANSAC
Ошибка положения, м.	0.0901	0.18	0.0750	0.1298
Ошибка поворота, м.	0.0991	0.2642	0.0963	0.1381
Время работы (реальное = 128 сек.)	114.7312	94.3912	58.9953	39.1953

Таблица 1: средние величины полученных результатов.

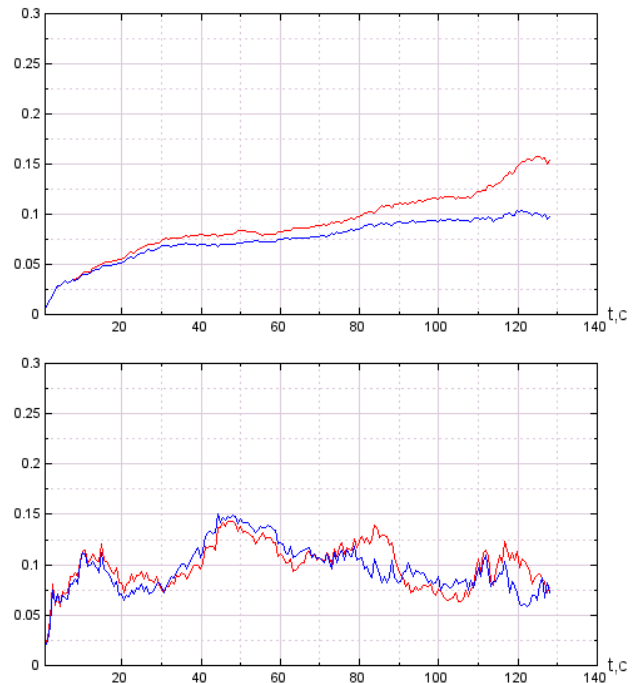
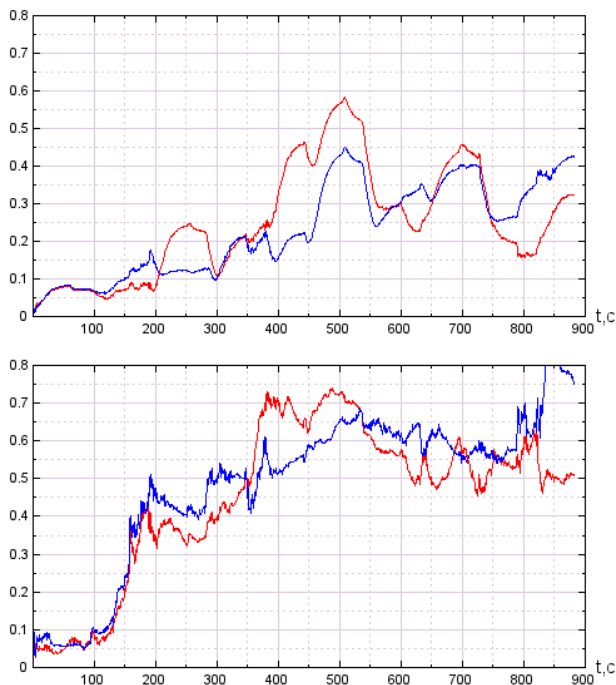


Рис. 1: Ошибка положения ПА в метрах (вверху) и ошибка ориентации в градусах (внизу) с параметрами  $S=50, T=1$  (красный график),  $T=4$  (синий график) и длиной траектории 128 с.

Эксперименты на более длинной траектории (рис. 2) на той же сцене и с теми же значениями параметров показали, что алгоритм дает отклонение траектории в пределах 20 см. и 0.5 гр. на протяжении 350 сек., что может быть приемлемым для решения задачи навигации в условиях локального маневрирования.



**Рис. 2:** Ошибка положения ПА в метрах (вверху) и ошибка ориентации в градусах (внизу) с параметрами  $S=50$ ,  $T=1$  (красный график),  $T=4$  (синий график) и длинной траектории 882 с.

## 7. ЗАКЛЮЧЕНИЕ

Полученные в вычислительных экспериментах оценки эффективности предложенного метода визуальной навигации показали перспективность его применения для целей локального маневрирования ПА. В дальнейшем планируется повышения вычислительной эффективности метода за счет реализации параллельных вычислений с использованием технологии CUDA, что в полной мере обеспечит его работу в режиме реального времени. Также возможным направлением повышения точности вычисления траектории, является учёт показаний бортовой навигационной системы ПА. Также планируется проведение натурных экспериментов с использованием разработанного программного комплекса.

## 8. ССЫЛКИ

- [1] M.A.Fischler, R.C.Bolles. Random Sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography. CACM 24.381-395. 1981.
- [2] D. Nister. Preemptive RANSAC for Live Structure and Motion Estimation, IEEE International Conference on Computer Vision, pp. 199-206, 2003.
- [3] D. Fontanelli, L. Ricciato, and S. Soatto. A fast ransac-based registration algorithm for accurate localization in unknown environments using lidar measurements, in Proc. IEEE Int. Conf. on Automation Science and

Engineering, Scottsdale, AZ, USA, 22-25 September 2007, pp. 597-602.

- [4] B. Kitt, A. Geiger and H. Lategahn. Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme, IEEE Intelligent Vehicles Symposium (IV), pp. 486-492. 2010.
- [5] D. Scaramuzza, F. Fraundorfer and R. Siegwart. Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC, Proc. IEEE Int. Conf. Robot. Autom., pp.4293 -4299 2009.
- [6] K.Berthold, P.Horn. Closed-form solution of absolute orientation using unit quaternions. 1987.
- [7] Carlo Tomasi and Takeo Kanade. Detection and Tracking of Point Features. Carnegie Mellon University Technical Report CMU-CS-91-132, April 1991.
- [8] Jianbo Shi and Carlo Tomasi. Good Features to Track. IEEE Conference on Computer Vision and Pattern Recognition, pages 593-600, 1994.
- [9] O.Chum, J.Matas, J.Kittler. Locally Optimized RANSAC. DAGM 2003.



**Рис. 3.** Кадр с бортовой камеры с изображением морского дна.

# Система материалов и эффектов для реалистичной визуализации трехмерных сцен

А.Н. Артиков, Т.Н. Артиков

Новосибирский Государственный Университет

Институт автоматизации и электрометрии СО РАН, Новосибирск, Россия

{a.artikov, t.artikov}@gmail.com

## Аннотация

В работе предлагается способ реализации материалов и эффектов для систем визуализации реального времени. Ключевыми особенностями данного решения являются использование автоматизированной генерации шейдеров для материалов и представление эффектов в виде графов фильтров, обрабатывающих изображение. Большая часть расчетов, связанных с визуализацией, выполняется на графическом ускорителе, что обеспечивает высокую производительность системы.

**Ключевые слова:** реалистичная визуализация, подготовка шейдеров, обработка изображений на видеокарте.

## 1. ВВЕДЕНИЕ

В современных системах, предназначенных для визуализации интерактивных трехмерных сцен, важным требованием является реалистичность визуализации. Эти системы применяются в виртуальных тренажерах, компьютерных играх, различных обучающих демонстрациях, приложениях научной визуализации. Для достижения реалистичности визуализация сцены осуществляется с учетом материалов объектов (визуальных свойств поверхности), а также применяются эффекты обработки изображений [1].

Видеокарты предоставляют широкие возможности для управления конвейером визуализации с помощью шейдеров [8]. Материал объекта задается функцией, выполняемой на графическом процессоре. В ней происходит текстурирование объекта, расчет освещения, микрорельефа, прозрачности и других свойств. Эффекты обработки изображений также представляют собой функции, вычисляемые на видеокарте, но, в отличие от материалов, эффекты можно комбинировать. Для этого выходы одних эффектов подаются на входы другим. Эффекты могут быть применены как к изображению, полученному при визуализации сцены, так и к входным изображениям (например, к изображениям, считанным из видео-файлов).

Существующие системы материалов разрабатываются для использования с конкретными типами сцен (ландшафты, интерьеры) и ограничены заранее заготовленными свойствами и эффектами. В отличие от них, разработанная авторами система материалов и эффектов лишена этих недостатков.

## 2. МАТЕРИАЛЫ

Материалом трехмерного объекта будем называть совокупность визуальных свойств его поверхности. К этим свойствам относятся цвет поверхности, рельефность, прозрачность, самосвечение, способность объекта отражать и преломлять свет.

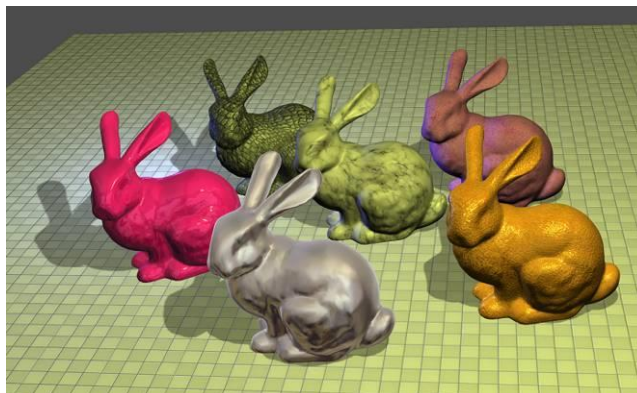


Рис 1: Один и тот же объект с различными материалами

В системах визуализации свойства материала задаются вещественными числами и текстурами. Во время визуализации видеокарта вычисляет изображение объекта с помощью некоторого шейдера на основе информации о геометрии объекта, его материала и свойств сцены (освещение, туман). В зависимости от того, какие свойства имеет материал и сцена, нужно использовать тот или иной шейдер. Количество различных свойств, поддерживаемых системой, велико (несколько десятков), соответственно количество возможных сочетаний свойств составляет несколько тысяч, т.е. написание всех необходимых шейдеров вручную не представляется возможным. Для решения этой задачи в системе реализована автоматизированная генерация кода шейдеров. Генерация кода осуществляется при помощи комбинации двух подходов: условная компиляция и высокоуровневые шейдеры.

Условная компиляция основана на использовании директив препроцессора для генерации кода шейдеров. Ниже приведен пример директивы препроцессора, в котором в зависимости от наличия у материала диффузной текстуры, осуществляется либо считывание из этой текстуры, либо берется цвет, заданный вектором вещественных чисел.

```
#ifdef HAS_DIFFUSE_MAP
    diffuse = tex2D(diffuseMap, tc);
#else
    diffuse = diffuseColor;
#endif
```

Шейдер, содержащий директивы препроцессора, называется убершейдером [4]. Один и тот же убершейдер компилируется несколько раз с разными флагами компиляции, в результате чего получаются разные шейдеры. Помимо булевых флагов так же реализована поддержка целочисленных констант, задаваемых на этапе компиляции, что может быть использовано для задания количества источников света. За счет использования условной компиляции код шейдера

получается максимально производительным: в нем отсутствуют лишние вычисления, считывания из текстур и операции условного перехода, а границы циклов имеют константные значения, за счет чего циклы лучше оптимизируются компилятором. Однако чтобы сделать систему расширяемой, одного этого подхода недостаточно.

Необходима возможность добавлять в систему новые способы расчета характеристик поверхности объекта и новые модели освещения. Для этой цели предлагаются высокоуровневые шейдеры: шейдер поверхности и шейдер освещения. В шейдере поверхности вычисляются следующие характеристики поверхности: диффузный и зеркальный цвет, прозрачность, нормаль, самосвечение.

Пример шейдера поверхности, в котором цвет поверхности берется из текстуры:

```
SurfaceOutput surface(SurfaceInput in)
{
    SurfaceOutput out;
    out.diffuse = tex2D(diffuseMap, in.tc);
    out.normal = in.normal;
    out.specular = float3(1.0, 1.0, 1.0);
    out.emission = float3(0.0, 0.0, 0.0);
    return out;
}
```

Шейдер освещения описывает некоторую модель освещения, то есть то, как поверхность взаимодействует с источниками света. Этот шейдер принимает на вход нормаль поверхности, направление взгляда и на источник света, а возвращает силу диффузной и зеркальной составляющих освещения.

Пример шейдера освещения по Фонгу:

```
float sP : SPECULAR_POWER;
LightingOutput lighting(LightingInput in)
{
    LightingOutput out;
    float3 R = reflect(in.lDir, in.normal);
    out.diffK = max(0, dot(in.normal, -in.lDir));
    out.specK = pow(max(0, dot(R, -in.vDir)), sP);
    return out;
}
```

На основе высокоуровневых шейдеров генерируется код вершинного и пиксельного шейдеров по определенному шаблону, в котором реализована поддержка нескольких источников света, теней и тумана. В коде высокоуровневых шейдеров могут содержаться директивы препроцессора, т.е. получаемые из них шейдеры являются убершейдерами. Далее убершейдер подается в модуль условной компиляции. Этот модуль выставляет нужные флаги и производит шейдер, с помощью которого будет осуществляться визуализация объекта.

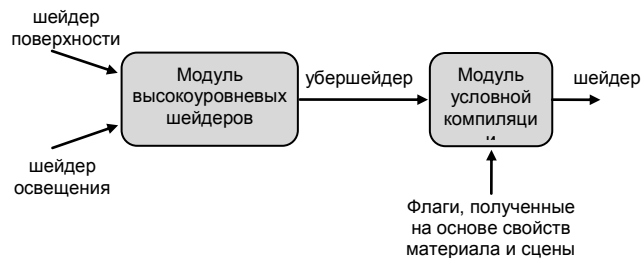


Рис 2: Генерация шейдеров

Для добавления новых возможностей пишутся только высокоуровневые шейдеры, весь остальной код шейдеров генерируется системой. Шейдеры поверхности могут комбинироваться с любым из шейдеров освещения, что дает широкие возможности для настройки внешнего вида объектов. Условная компиляция позволила повысить производительность визуализации в 1.5-2 раза по сравнению с использованием шейдеров, в которых вместо директив препроцессора применяются операции условного перехода. С помощью описанной системы был реализован стандартный материал, поддерживающий до 8 различных текстурных карт и использующий освещение по Фонгу. Так же реализованы трипланарное текстурирование [9] и модель освещения Кука-Торренса [6].

### 3. ЭФФЕКТЫ

Эффекты в системе – это преобразования, которые можно назначить на изображения. В простейшем случае эффект представляет собой фильтр – алгоритм, принимающий на вход несколько изображений и возвращающий одно в качестве результата. Для обеспечения высокой производительности алгоритмы фильтров реализуются для графических ускорителей с использованием пиксельных шейдеров [2].

Выходы одних фильтров можно подавать на входы другим, соединяя их в ориентированный граф, что позволяет реализовать более сложные эффекты из нескольких простых. Процесс обработки изображений графом фильтров заключается в последовательном применении фильтров в порядке, определяемом зависимостями по данным.

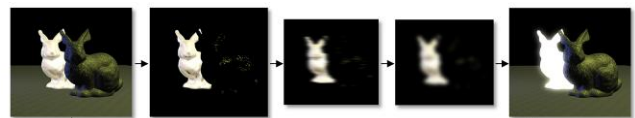


Рис 3: Граф фильтров для эффекта «сияние»

Эффекты, назначенные на окно вывода сцены, могут использовать не только изображение сцены, но и дополнительные изображения, в пикселях которых хранится некоторая информация о сцене: расстояние до камеры, нормаль к поверхности, позиция в пространстве. Эти изображения получают путем повторной визуализации объектов сцены, при этом пиксельные шейдеры материалов заменяются специальным шейдером, который вместо цвета возвращает нужную информацию о сцене.

Кроме этого, реализована поддержка временной фильтрации изображений – фильтр может принимать на вход несколько последовательных кадров одного динамического изображения. Данная возможность необходима для эффектов обработки видео.

Эффекты в системе описываются на декларативном скриптовом языке. Формат описания позволяет задать фильтры, содержащиеся в эффекте, указать каждому фильтру пиксельный шейдер, соединить выходы одних фильтров с входами других, задать свойства выходных изображений фильтров – абсолютный или относительный размер, цветовое пространство, формат пикселей.

Приведем пример описания для эффекта «сияние»:

```
EFFECT.glow = {  
  PASS = {  
    brightPass = SHADER.brightPass {  
      img = IN  
    }  
    blurX = SHADER.blurX {  
      img = PASS.brightPass  
      OUT.SCALE = (0.5, 0.5)  
    }  
    blurY = SHADER.blurY {  
      img = PASS.blurX  
      OUT.SCALE = (0.5, 0.5)  
    }  
    final = SHADER.add {  
      img1 = IN  
      img2 = PASS.blurY  
    }  
  }  
  OUT = final  
}
```

Система считывает описание эффекта и строит по нему граф фильтров. На основе топологии графа автоматически определяется порядок выполнения фильтров, а так же последовательность создания и удаления временных изображений, необходимых для хранения промежуточных результатов вычислений.

С помощью разработанной системы реализованы эффекты обработки видео: преобразование форматов, устранение чересстрочности (deinterlacing), выделение актера на монохромном фоне [3]; а так же эффекты постобработки изображений трехмерных сцен: сияние, имитация рисунка карандашом, глубина резкости [7], туман, глобальное освещение [10] и др.

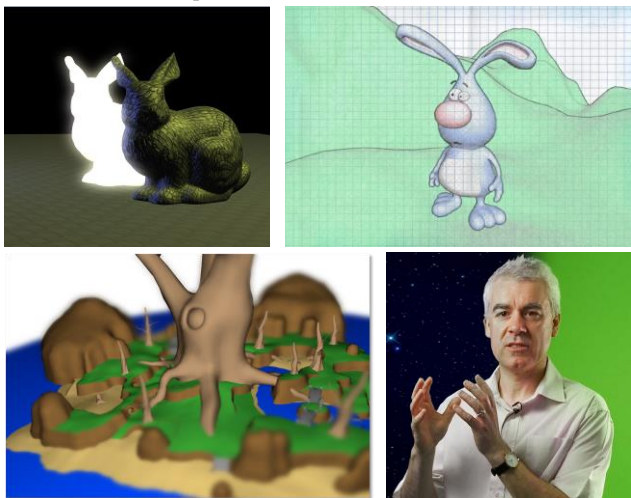


Рис 4: Примеры эффектов

## 4. ЗАКЛЮЧЕНИЕ

Авторами разработаны подходы для реалистичной визуализации трехмерных сцен и реализована расширяемая система материалов и эффектов, позволяющая достичь высокого качества визуализации. Система обладает высокой производительностью и может быть использована для широкого класса интерактивных графических приложений. С помощью системы разработан набор примеров материалов и эффектов, демонстрирующих её возможности.

## 5. ССЫЛКИ

- [1] Городилов М. А., Коростелев Е. И. Программируемые материалы и спецэффекты для систем визуализации реального времени. // Труды конференции МНСК-2011. 2011.
- [2] Дьячков В. Реализация процессора эффектов постобработки. // Статья на сайте [www.uraldev.ru](http://www.uraldev.ru). 2008.
- [3] Ковальков М.А. Разработка и реализация алгоритма рипроекции на базе современного графического акселератора. // Труды конференции Графикон-2006. 2006.
- [4] Сладков Д. Промышленные убершейдеры. // Труды конференции КРИ-2008. 2008.
- [5] Akenine-Moller T. Real-time rendering. 2008.
- [6] Cook R., Torrance K. A Reflectance Model for Computer Graphics. 1982.
- [7] Hammon E. Practical Post-Process Depth of Field. // GPU Gems 3 Chapter 28. 2008.
- [8] Kessenich J. The OpenGL® Shading Language. 2010.
- [9] Nicholson K. GPU Based Algorithms for Terrain Texturing. 2008.
- [10] Ritschel T. Approximating Dynamic Global Illumination in Image Space. 2009.

## Об авторах

Артиков Артиков Неъматжанович – студент факультета информационных технологий Новосибирского Государственного Университета, инженер-программист лаборатории синтезирующих систем визуализации Института Автоматики и Электротри СО РАН.  
e-mail: [a.artikov@gmail.com](mailto:a.artikov@gmail.com)

Артиков Тимур Неъматжанович – студент факультета информационных технологий Новосибирского Государственного Университета, инженер-программист лаборатории синтезирующих систем визуализации Института Автоматики и Электротри СО РАН.  
e-mail: [t.artikov@gmail.com](mailto:t.artikov@gmail.com)

## Моделирование позвоночника

Дмитриев Сергей Валерьевич

Новосибирский Государственный Технический Университет

sdmitriew@gmail.com

### ВВЕДЕНИЕ

Оперативное лечение деформаций позвоночника остается актуальной задачей современности и одной из наиболее трудных проблем травматологии и ортопедии поскольку количество этих деформаций не имеет тенденций даже к уменьшению, а меры профилактики деформаций позвоночника и консервативное их лечение или малоэффективны или невозможны.

Наряду с этим все больше ортопедов и травматологов устремляют свои усилия на разработку наиболее эффективных способов коррекции деформаций позвоночника. Причем, если на заре своего становления вертебрология решала вопросы коррекции умеренно выраженных деформаций, то в настоящее время делаются попытки коррегировать очень грубые и сложные деформации, лечение которых раньше было неосуществимо, или лечебные пособия носили паллиативный характер.

Изучение патологии позвоночника методами моделирования, требует постоянного сопоставления получаемых результатов с нашими представлениями об уже изученных сторонах этой проблемы, а так же подтверждения выявляемых закономерностей или другими экспериментальными методами или на практике.

Дифференцированный биомеханический анализ патологических состояний позвоночника диктует необходимость разработки классификаций различных видов деформаций позвоночника на основе количественных показателей.

### 1 СТРОЕНИЕ ПОЗВОНОЧНИКА

Позвоночный столб (позвоночник) образован последовательно накладывающимися друг на друга позвонками, которые соединены между собой при помощи межпозвоночных дисков, связок и суставов.

Формируя осевой скелет, позвоночный столб выполняет опорную функцию, служит гибкой осью туловища, участвует в образовании задней стенки грудной и брюшной полостей и таза и является вместилищем для спинного мозга. В позвоночном канале находится спинной мозг. Таким образом, позвоночник принимает участие в защите спинного мозга и внутренних органов от повреждений. В вертикальном положении позвоночный столб образует опору для головы, органов грудной и брюшной полостей. Сила тяжести, воспринимаясь позвоночным столбом, увеличивается сверху вниз, поэтому размеры образующих его сегментов (позвонков) в нижнем отделе столба больше, чем верхних. В позвоночном столбе выделяют пять отделов: шейный, грудной, поясничный, крестцовый и копчиковый. Только крестцовый отдел позвоночного столба является неподвижным, остальные его отделы обладают различной степенью подвижности.

### 2 ОСНОВНЫЕ ДИАГНОСТИЧЕСКИЕ ПАРАМЕТРЫ

Длина позвоночника по кривизне

$$L_s = Hb_2 + Hb_3 + \dots + Hb_{23} + Hd_2 + Hd_3 + \dots + Hd_{23}$$

Угол наклона тела позвонка в сагиттальной плоскости:

$$Fb_i = \arg(x^b_i - x^n_i; y^b_i - y^n_i)$$

Угол наклона тела позвонка во фронтальной плоскости:

$$Fbz_i = \arg(z^b_i - z^n_i; y^b_i - y^n_i)$$

Угол наклона диска в сагиттальной плоскости:

$$Fd_i = \arg(x^n_i - x^b_{i+1}; y^n_i - y^b_{i+1})$$

Угол наклона диска во фронтальной плоскости:

$$Fdz_i = \arg(z^n_i - z^b_{i+1}; y^n_i - y^b_{i+1})$$

Длина хорды дуги:

$$L_c = \sqrt{(x^b_0 - x^n_{nc}) + (y^b_0 - y^n_{nc})}$$

$$L_t = \sqrt{(x^b_{nc} - x^n_{nc+nt}) + (y^b_{nc} - y^n_{nc+nt})}$$

$$L_t = \sqrt{(x^b_{nc+nt} - x^n_{nc+nt+nl})^2 + (y^b_{nc+nt} - y^n_{nc+nt+nl})^2}$$

Модель движения позвонка

Основываясь на формализованном описании анатомического строения позвонков в сагиттальной плоскости, нами разработана математическая модель анализа перемещений позвонка при сгибании и разгибании. Общая схема представлена на рисунке 1.

1. Перемещение вышележащего позвонка в плоскости суставной поверхности верхних суставных отростков нижележащего позвонка;
2. Поворот вышележащего позвонка вокруг верхушки верхнего суставного отростка нижележащего позвонка.

Граничными условиями для каждой фазы является перемещение точки каудо-вентрального угла тела позвонка по-вертикали не более 1/2 высоты межтелового промежутка, и не более 3.0 мм. по-горизонтالي.

$L_i$  - длина верхней суставной поверхности нижележащего позвонка в сегменте;

$L_s$  - длина нижней суставной поверхности верхнего в сегменте;

$C$  - точка каудо-вентрального угла перемещаемого позвонка;

$H_d$  - высота вентрального отдела межпозвоночного диска;

$S_b$  - сагиттальный размер базальной замыкательной пластинки перемещаемого позвонка;

$S_h$  - сагиттальный размер покровной замыкательной пластинки нижележащего позвонка.

Угол  $AL$  - угол наклона суставной поверхности верхнего суставного отростка нижележащего позвонка;

Рассчитываемые величины:

$\alpha$  - угловое смещение между отрезками  $L_i$  и  $L_s$ .

$X_c$  - линейное перемещение точки  $C$  по оси  $X$ ;

$Z_c$  - линейное перемещение точки  $C$  по оси  $Z$ ;

Перемещение точки  $a$  в новое положение с координатами  $X_c, Z_c$  (рисунок 2) происходит за две фазы: сначала за счет скольжения по суставной поверхности точка  $a$  перемещается в положение точки  $a'$ , а затем происходит поворот вокруг точки с координатами  $X_0, Z_0$  по окружности радиусом  $R$ . Отрезок  $a'-a$  параллелен суставной поверхности. Таким образом, отрезок  $a'-a$  равен смещению по суставной поверхности, а угол  $\alpha$  повороту тела позвонка.

Необходимо решить следующую задачу: по заданным значениям  $X_c, Z_c$  найти угол  $\alpha$  и длину отрезка  $a'-a$ , при этом считаются известными координаты  $X_0, Z_0, \varphi$  и положение точки  $a$  т.е. исходная высота межпозвонкового диска (величина  $h$ ). Для решения этой задачи необходимо решить систему уравнений:

$$\begin{cases} z - h = kx \\ (X - X_2)^2 + (Z - Z_2)^2 = R^2 \end{cases}$$

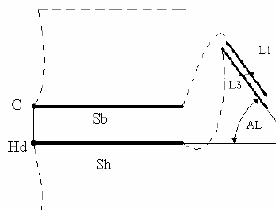


Рисунок 1: Общая схема модели перемещения позвонка в сагиттальной плоскости

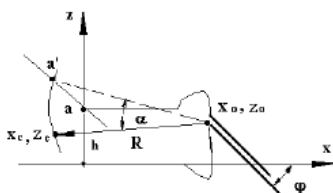


Рисунок 2: Расчетная схема перемещения позвонка в сагиттальной плоскости

## ЗАКЛЮЧЕНИЕ

Основываясь на основании теории [2] и приведенном выше математическом аппарате разрабатывается система, позволяющая спрогнозировать компенсаторные реакции в

позвоночнике человека при различных механических патологиях. На рисунке 3 изображен пример механической деформации с учетом смещения позвонков в сагиттальной плоскости.

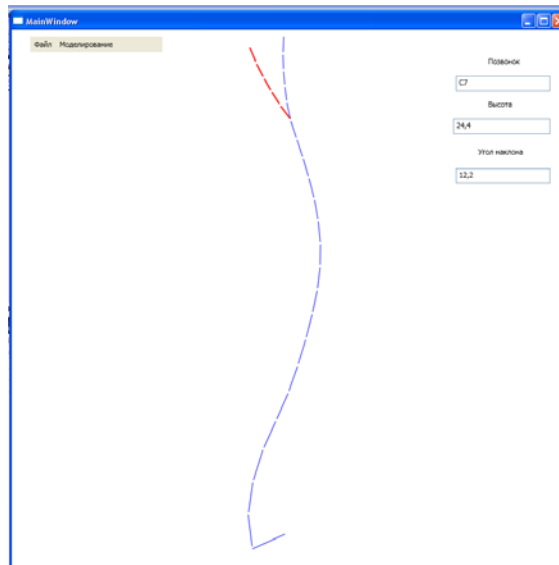


Рисунок 3: Пример патологической деформации, со смещением вышележащих позвонков

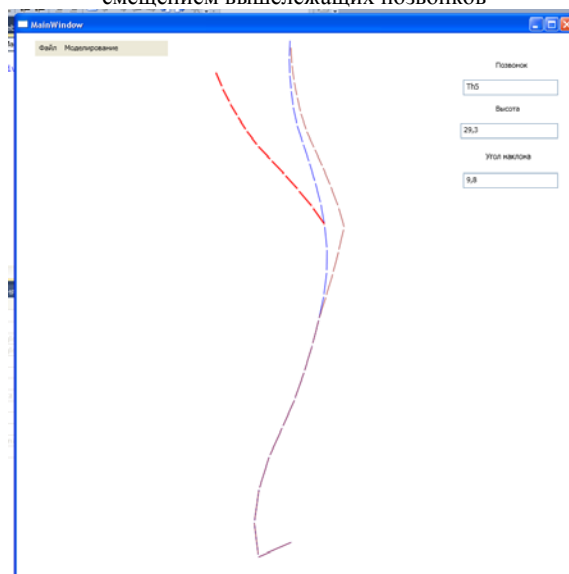


Рисунок 4: Пример скомпенсированной деформации

## ССЫЛКИ

- [1] Хвсюк Н.И. Кинематика позвоночного столба в сагиттальной плоскости в норме и при дегенеративной нестабильности // Ортопедия травматология и протезирование.- 1978.-№ 9.-С.29-34.
- [2] Гладков А.В. Создание системы клинко-биомеханического анализа состояния позвоночника при различной патологии: Диссертация д-ра мед. наук.- Новосибирск, 1994.-320с.

# ОБНАРУЖЕНИЕ И СЛЕЖЕНИЕ ЗА ОБЪЕКТОМ НА ОСНОВЕ СОВМЕСТНОГО ИСПОЛЬЗОВАНИЯ АЛГОРИТМОВ ВИОЛЫ-ДЖОНСА И АНАЛИЗА ТРАЕКТОРИЙ

Мария Николаевна Гурьева

Факультет автоматике и информационных технологий в управлении  
Рязанский Государственный Радиотехнический Университет, Рязань, Россия  
maria\_guryeva@mail.ru

## Аннотация

Настоящая работа посвящена задаче обнаружения и слежения за наземным объектом, наблюдаемым в последовательности изображений. Предложенный алгоритм решения задачи и программное обеспечение, реализующее его, обеспечивают надежное обнаружение и слежение за наземным ориентиром, в качестве которого выступает взлетно-посадочная полоса (ВПП), при наблюдении с борта летательного аппарата (ЛА).

### 1. ВВЕДЕНИЕ

Автоматическое обнаружение и слежение за объектами, наблюдаемыми в последовательности изображений, является актуальной задачей, часто возникающей при разработке навигационных комплексов пилотируемых и беспилотных ЛА. При этом одной из важных проблем остается обеспечение обнаружения и слежения при неопределенности взаимного пространственного расположения видеокамеры и объекта. В настоящей работе рассмотрена задача обнаружения и слежения за аэродромом с борта ЛА в условиях, когда начальное взаимное расположение аэродрома и ЛА неизвестно. Существенным требованием, предъявляемым к разрабатываемому алгоритму, является возможность его реализации в бортовых системах обработки информации.

### 2. АЛГОРИТМ ОБНАРУЖЕНИЯ И СЛЕЖЕНИЯ ЗА ОБЪЕКТОМ

Достаточно эффективным подходом к автоматическому обнаружению объектов в условиях неопределенности их расположения и ориентации является применение алгоритма Виолы-Джонса [1]. Его недостатком применительно к поставленной задаче является достаточно высокий уровень ложных обнаружений, связанный с тем, что обнаруживается сразу несколько областей изображения, похожих на искомым объект. Для преодоления этого недостатка предлагается списки областей изображения, сформированные алгоритмом Виолы-Джонса, подвергать дополнительной обработке с помощью алгоритма анализа траекторий [2].

Рассмотрим основные особенности алгоритмов Виолы-Джонса и анализа траекторий, имеющие значение применительно к решаемой задаче.

#### 2.1 Алгоритм Виолы-Джонса

Для применения алгоритма Виолы-Джонса необходимо выполнить предварительное обучение каскада классификаторов, используя эталонное изображение объекта. Обучение идет очень медленно, но обнаружение объекта осуществляется очень быстро, именно поэтому был выбран данный метод обнаружения.

Результатом обнаружения на каждом кадре является список параметров областей изображения, содержащих искомым объект.

#### 2.2 Алгоритм анализа траекторий

Алгоритм анализа траекторий основан на разбиении двудольного графа. На входе алгоритма имеются данные измерений - список параметров областей изображения, выделенных с помощью обработки каждого изображения в видеопоследовательности. На выходе - список найденных объектов и их свойств, в котором каждый элемент списка обозначен уникальным номером [2].

Каждый найденный объект имеет следующие свойства:

- Идентификатор - уникальный номер, присваиваемый новому обнаруженному объекту. Номер закрепляется за данным объектом до тех пор, пока не будет принято решение о пропаже объекта;
- Координаты левого края объекта;
- Координаты верхнего края объекта;
- Ширина;
- Высота;
- Время жизни - количество кадров видеопоследовательности, на которых присутствовал данный объект с момента первого обнаружения;
- Тип объекта.

Первый шаг алгоритма заключается в построении двудольного графа, первой группой вершин которого является множество объектов, известных на текущем кадре, а второй - множество связанных областей, найденных на кадре. Между парами вершин графа вычисляется мера соответствия. Граф содержит все дуги, для которых эта мера превышает заданный порог.

Для принятия решения о появлении и исчезновении объектов граф разделяется на связанные компоненты. Для каждой компоненты дуга с минимальным весом удаляется. Рекурсивная процедура повторяется для каждого полученного подграфа. На основе списка компонент, полученных после разбиения двудольного графа, выполняется оценивание координат объектов с помощью фильтра Калмана.

Целесообразность использования этого алгоритма обуславливается наличием такого параметра обнаруженного объекта, как время жизни. На этапе предварительного обнаружения искомым принимается объект с наибольшим временем жизни и таким образом снижается вероятность ложного обнаружения.

#### 2.3 Алгоритм обнаружения и слежения за объектом

На основе совместного применения метода Виолы-Джонса и алгоритма анализа траекторий был разработан алгоритм, осуществляющий обнаружение и слежение за объектом на видеопоследовательности.

Разработанный алгоритм включает в себя следующие этапы:



- В течение первых 50 кадров алгоритм выполняется предварительное обнаружение. На каждом кадре видеопоследовательности производятся следующие действия:
  - Объекты обнаруживаются с помощью каскада классификаторов, обученного на основе метода Виолы-Джонса;
  - С помощью алгоритма анализа траекторий анализируется список найденных на предыдущем шаге объектов.
- По прошествии 50 кадров происходит следующее:
  - Анализируется список найденных объектов и их свойств и выявляется объект с наибольшим временем жизни;
  - Запоминается идентификатор этого объекта для последующего слежения за ним на видеопоследовательности.
- Далее разработанный алгоритм переходит в режим слежения. На каждом кадре видеопоследовательности происходит следующее:
  - Объекты обнаруживаются с помощью каскада классификаторов, обученного на основе метода Виолы-Джонса;
  - С помощью алгоритма анализа траекторий анализируется список найденных на предыдущем шаге объектов.
  - Результатом работы алгоритма слежения за объектом принимаются координаты объекта, имеющего запомненный ранее идентификатор.

Координаты найденного объекта могут в дальнейшем использоваться для корректировки курса ЛА.

### 3 Экспериментальные исследования

Для проведения экспериментальных исследований было разработано программное обеспечение на языке C++.

В качестве объекта для поиска было принято спутниковое изображение аэродрома, представленное на Рис.1. Изображение получено с ресурса GoogleMaps.



Рис.1 – Изображение аэродрома со спутника

На основе этого объекта был обучен каскад классификаторов. Обучение производилось с использованием 5000 позитивных образцов, созданных при помощи утилиты `opencv_createsamples`, и 2500 негативных. Количество этапов обучения было принято равным 14. Весь процесс обучения занял приблизительно 63 часа 22 минуты. Процесс поиска объекта на кадре размером 1160x656 при этом выполняется в среднем за 0,4 секунды.

Также при помощи программы «Google планета Земля» смоделирован полет самолета над аэродромом.

Пример работы программы в режиме предварительного обнаружения представлен на рис.2. Несмотря на то, что

искомый объект найден, присутствует некоторое количество ложных обнаружений.



Рис.2 – Режим предварительного обнаружения

Пример работы программы в режиме слежения представлен на рис.3. Видно, что все ложные обнаружения проигнорированы.

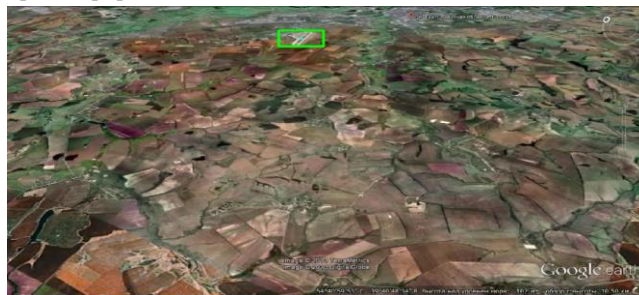


Рис.3 – Режим слежения

Разработанный алгоритм эффективно работает также в присутствии шума на изображении.

### 3.ЗАКЛЮЧЕНИЕ

В данной работе был предложен алгоритм обнаружения аэродрома на основе метода Виолы-Джонса. При помощи алгоритма анализа траектории был значительно снижен процент ложных обнаружений. Анализ результатов компьютерных экспериментов позволяет говорить об успешной работе созданного алгоритма и программы при обнаружении и слежении за объектом.

### 4. СПИСОК ЛИТЕРАТУРЫ

1. P. Viola and M.J. Jones, «Rapid Object Detection using a Boosted Cascade of Simple Features», proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2001), 2001
2. Boris Alpatov and Pavel Babayan, «multiple object tracking based on the partition of the bipartite graph», Proc. SPIE 8186, 81860B (2011).
3. Р.Гонсалес, Р.Вудс, «Цифровая обработка изображений», ISBN 5-94836-028-8, изд-во: Техносфера, Москва, 2005. – 1072 с.
4. Местецкий Л. М., «Математические методы распознавания образов», МГУ, ВМиК, Москва, 2002–2004., с. 42 – 44

# Использование контрольных кластеров для корректирования деформации антропоморфных полигональных моделей

Елена Гридчина, Дмитрий Заставной  
Факультет математики, механики и компьютерных наук  
Южный федеральный университет  
Ростов-на-Дону, Россия  
helen.gridchina@gmail.com, dzast@sfnedu.ru

Александр Букатов, Борис Заставной  
Южно-Российский региональный центр  
информатизации, Южный федеральный университет,  
Ростов-на-Дону, Россия  
{baa, zast}@sfnedu.ru

## Аннотация

В работе представлен новый метод для создания анимации антропоморфных персонажей, основанный на выполнении корректировки вершин модели после применения базовой трансформации вершин на основе скелетной анимации LBS (Linear Blend Skinning). Корректируемые вершины объединяются в группы, называемые контрольными кластерами, для которых задаются эталонные формы. Вычисление трансформаций для прочих вершин модели выполняется на основе интерполяции при помощи кардинальных сплайнов.

**Ключевые слова:** Скелетная анимация, Контрольный кластер, Кардинальные сплайны.

## 1. ВВЕДЕНИЕ

В настоящее время в сфере компьютерной графики продолжают активные исследования в области разработки методов реалистичной деформации трёхмерных моделей антропоморфных персонажей [4, 7], что, в частности, объясняется некоторыми недостатками традиционно используемых методов анимации [1].

Базовым способом деформации полигональной поверхности модели является традиционный метод скелетной анимации Linear Blend Skinning (LBS). Эффективным методом, компенсирующим нежелательные артефакты LBS, является технология Pose Space Deformation (PSD) [6], которая сочетает создание анимации модели при помощи традиционной LBS и блендинга; данный метод широко используется в настоящее время, но является весьма трудоёмким [5].

В настоящей статье рассматривается метод корректировки скелетной анимации, который можно рассматривать как дальнейшее развитие идеи PSD. Предлагаемый метод основан на использовании предложенных авторами контрольных кластеров. В данной статье этот метод рассмотрен для двумерного случая и может быть обобщён для случая трёхмерных полигональных моделей.

## 2. МОДЕЛЬ ПЕРСОНАЖА В LBS

Модель антропоморфного персонажа (далее — модель) в двумерном пространстве представляет собой многоугольник  $P$ .  $V_P$  — множество всех вершин модели  $P$ . С моделью  $P$  связан скелет, который представляет собой иерархическую структуру костей  $S = \{\vec{b}_i\}$ . С каждым узлом структуры  $\vec{b}_i$  связана своя локальная система координат и двумерное преобразование. Двумерное преобразование сустава  $i$  может быть описано матрицей  $W_i$   $3 \times 3$ , описывающей трансформацию (вращение, перемещение и масштабирование). Кости скелета образуют иерархию: каждая

дочерняя кость наследует трансформации родительской кости. В процессе привязки модели  $P$  к скелету, каждой вершине модели  $\vec{v} \in V_P$  ставится в соответствие набор весов всех костей скелета  $\{w_i\}$ ,  $\sum_i w_i = 1$ , где  $w_i$  — вес  $i$ -ой кости. Вес кости определяет степень влияния кости на перемещение вершины модели. Положения (собственные трансформации) костей скелета в позе привязки будем обозначать  $\{B_i\}$ . Таким образом, трансформированное положение отдельной вершины при произвольной конфигурации скелета  $\{W_i\}$  рассчитывается по формуле:

$$\vec{v}' = LBS(\vec{v}) = \sum_i w_i W_i B_i^{-1} \vec{v} \quad (1)$$

При высокой вычислительной эффективности и универсальности этот традиционный подход скелетной анимации LBS характеризуется рядом нежелательных артефактов (дефектов), в том числе, например, потерей объёма, что приводит к значительному снижению реалистичности получаемой анимации. Один из типичных нежелательных артефактов LBS традиционно называется “collapsing elbow” (буквально — “сплюсывающийся локоть”) (см. Рис. 1).

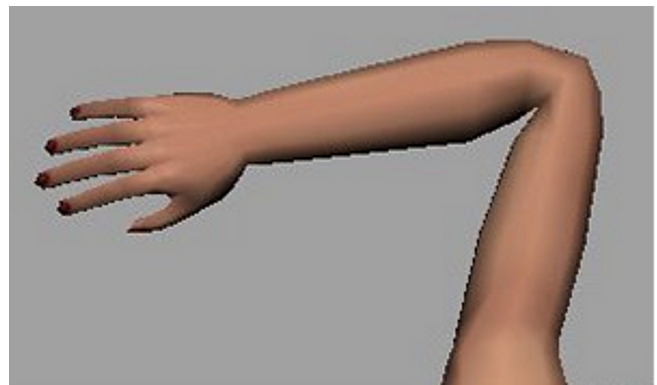


Рис 1: Некорректная деформация модели при сгибе в локте.

## 3. МЕТОД КОНТРОЛЬНЫХ КЛАСТЕРОВ

Для придания требуемой реалистичности модели, деформированной на основе метода LBS, предлагается выполнять для вершин, трансформации которых являются неудовлетворительными, некоторые дополнительные корректирующие трансформации. Для этой цели авторами статьи разработан оригинальный метод контрольных кластеров (МКК). Предлагаемый метод принадлежит к введённой группе корректирующих методов и использует информацию о контрольных кластерах, состоящих из групп

вершин модели, для корректировки модели после применения базовых трансформаций LBS.

### 3.1 Понятие контрольного кластера

Кластером (двумерной) модели  $s_{ij}$  будем называть пару вершин  $\langle \vec{v}_i, \vec{v}_j \rangle$ ,  $i \neq j$ ,  $\vec{v}_i, \vec{v}_j \in V_P$ . Для моделей с оснасткой (“rigging”) (т.е. с заданным скелетом  $S = \{\vec{b}_i\}$  и весами) будем рассматривать такие кластеры  $s_{ij}$ , что в позе привязки модели  $\exists \vec{b}_k, \vec{b}_l \in B \mid (\vec{v}_i \vec{v}_j, \vec{b}_k \vec{b}_l) = 0$ , где кость  $\vec{b}_k$  является дочерней по отношению к кости  $\vec{b}_l$  в иерархической структуре скелета  $S$ .

Контрольные кластеры создаются для тех вершин модели, для которых трансформация на основе LBS демонстрирует нежелательные артефакты; для таких кластеров задаются эталонные положения вершин. Для корректировки LBS используются пары  $\langle X^i, S^i \rangle$ , где  $X^i$  — конфигурация скелета, а  $S^i$  — набор контрольных кластеров модели. Вершины модели, целиком привязанные к одной кости (то есть  $\exists \vec{b}_k \in S \mid w_k = 1$ ) помещаются в контрольный кластер автоматически.

Вершины модели, не входящие в контрольные кластеры, будем далее неформально называть *свободными*.

Метод контрольных кластеров должен производить корректировку положений всех вершин модели для каждого кадра анимации. Сначала корректируются координаты вершин контрольных кластеров, затем положения свободных вершин модели рассчитываются на основе полученных координат вершин контрольных кластеров.

### 3.2 Расчёт текущего положения вершин контрольных кластеров

Для вершин контрольных кластеров вычисляются сдвиги в локальной системе координат кости  $\vec{b}_j$  по формуле  $\vec{d}_j = LBS^{-1}(\vec{v}, \vec{b}_j) B_j^{-1} \vec{v}_0$ , где  $\vec{v}_0$  — положение вершины в позе привязки,  $\vec{v} \in s \in S^i$  — эталонное положение вершины в позе  $X^i$ . Если текущая конфигурация скелета совпадает с конфигурацией  $X^i$ , то текущее положение вершины контрольного кластера  $\vec{v}'$  вычисляется по формуле

$$\vec{v}' = LBS(\vec{v}_0, \vec{d}) = \sum_j w_j W_j (B_j^{-1} \vec{v}_0 + \vec{d}_j). \quad (2)$$

Вычисление сдвига для вершины контрольного кластера для промежуточной конфигурации скелета  $X \neq X^i$  проводится аналогично технологии PSD:

$$\vec{d}_i = \sum_{i=0}^{N-1} \omega_i \varphi(\|X^i - X\|), \quad (3)$$

где  $\|X^i - X^j\|$  — евклидово расстояние между конфигурациями модели  $X^i$  и  $X^j$ ;  $\varphi(x)$  — гауссова радиальная базисная функция;  $\omega_i$  — веса соответствующих конфигураций  $X^i$  наборов контрольных кластеров.

### 3.3 Вычисление текущего положения свободных вершин

Трансформация свободных вершины выполняется на основе данных о её текущем LBS-положении с учётом информации о ближайших контрольных кластерах. Координаты вершины, полученные с помощью LBS, корректируются таким образом, чтобы они находились в соответствии с координатами вершин контрольных кластеров, что обеспечивает корректирование нежелательных артефактов LBS.

Для интерполяции трансформаций свободных вершин используются кардинальные сплайны, которые являются обобщением сплайнов Катмулла-Рома [3]. Данный вид сплайнов подходит для решаемой задачи, так как кардинальный сплайн имеет непрерывную первую производную и, вообще говоря, разрывную вторую. Непрерывность по первой производной обеспечивает гладкость модели, разрывность по второй производной даёт достаточную гибкость в использовании сплайна. Кроме того, кардинальный сплайн является локальным — влияние каждой контрольной точки распространяется лишь на некоторую окрестность кривой. Данные свойства сплайна позволяют вычислять деформацию приемлемым образом для таких традиционно проблемных с точки зрения анимации областей модели, как сгибы в локте.

#### 3.3.1 Выбор типа параметризации

Поведение кривой, построенной при помощи кардинального сплайна, в значительной степени зависит от типа её параметризации [8]. При выборе параметризации для настоящего метода были рассмотрены равномерная параметризация, параметризация по длине хорды, а также центростремительная параметризация (параметризация квадратного корня).

При использовании равномерной параметризации могут проявиться такие нежелательные артефакты, как узловые точки, что недопустимо при работе с анимируемыми моделями. При параметризации по длине хорды могут возникать самопересечения и точки возврата, которые исключены в случае применения центростремительной параметризации. К достоинствам центростремительной параметризации также относится то, что она даёт кривую, расположенную ближе к управляющему многоугольнику, чем кривые, полученные с помощью двух других способов параметризации.

В описываемом методе используется параметризация по длине хорды, что мотивируется следующим. Во-первых, кривизна сегмента кривой между контрольными точками при использовании данного типа параметризации больше в сравнении с другими типами, что позволяет достичь большей реалистичности при построении моделей органических объектов, так как не будет приводить к потере объёма. Во-вторых, хордовая параметризация доставляет наилучшую обусловленность линеаризованной системе уравнений продолжения, и в данном смысле считается лучшей из рассмотренных [2]. Таким образом, в методе используется параметризация  $t_{i+1} = t_i + |\vec{p}_{i+1} - \vec{p}_i|$ , где  $\vec{p}_i, \vec{p}_{i+1}$  — опорные точки сплайна.

#### 3.3.2 Определение параметров сплайна

Значение кардинального сплайна в текущей точке  $\vec{p}_i$  зависит от значений функции в четырёх соседних, опорных точках (двух точках слева и двух точках справа) и от значения параметра натяжения. Обозначим их  $\vec{p}_{i-2}, \vec{p}_{i-1}, \vec{p}_{i+1}, \vec{p}_{i+2}$  и  $q$  соответственно. Таким образом, производная в опорной точке будет рассчитываться по следующей формуле

$$m_k = (1 - q) \frac{\vec{p}_{k+1} - \vec{p}_{k-1}}{t_{k+1} - t_{k-1}}. \quad (4)$$

Рассмотрим также множество контрольных вершин, полученных на предыдущем шаге алгоритма,  $V_C = \{\vec{v}_{j_k}\}_k \mid j_{k_1} < j_{k_2}, k_1 < k_2, V_C \subset V_P$ . Для соседних контрольных вершин  $\vec{v}_{j_n}, \vec{v}_{j_{n+1}} \in V_C \mid j_{n+1} \neq j_n + 1$

необходимо вычислять с помощью интерполяции сплайном положение всех промежуточных вершин. В этом случае  $\vec{p}_{i-1} = \vec{v}_{j_n}, \vec{p}_{i+1} = \vec{v}_{j_{n+1}}$ . При этом выбор опорных точек сплайна  $\vec{p}_{i-2}, \vec{p}_{i+2}$  не является столь же определённым. Как правило,  $\vec{p}_{i-2} = \vec{v}_{j_{n-1}}, \vec{p}_{i+2} = \vec{v}_{j_{n+2}}$ . Но некоторые вершины модели  $P$  могут представлять собой специальный случай. Например, в случае работы с частью модели, обе вершины одного крайнего сечения модели не должны влиять на построение кривой (быть  $\vec{p}_{i+1}, \vec{p}_{i+2}$  соответственно), так как это приведёт к нежелательным деформациям модели. В случае если контрольная вершина  $\vec{p}_{i+1}$  является специальной, т.е. в модели нет вершины  $\vec{p}_{i+2}$ , дополнительная опорная вершина строится методом отражения  $\vec{p}_{i+2} = \vec{p}_{i+1} + (\vec{p}_{i+1} - \vec{p}_{i-1})$ .

С помощью параметра натяжения  $q \in [-1, 1]$  контролируется кривизна сегмента сплайна. В методе контрольных кластеров в качестве значения по умолчанию принимается  $q = -0.5$  (подробнее см. раздел 5).

#### 4. СРАВНЕНИЕ МЕТОДА КОНТРОЛЬНЫХ КЛАСТЕРОВ С LBS И PSD

Разработанный метод контрольных кластеров, с одной стороны, является универсальным, так как в его основе лежит универсальный метод LBS. С другой стороны, метод позволяет добиться реалистичности деформаций, компенсируя нежелательные артефакты LBS за счёт использования эталонных положений вершин кластеров. В общем случае описанный метод даёт аниматору меньший контроль над получаемой анимацией, чем метод PSD. Однако, при необходимости, можно задать контрольные кластеры таким образом, чтобы все вершины модели входили в какой-либо из кластеров. При таких условиях метод контрольных кластеров будет работать аналогично PSD, предоставляя аниматору полный контроль над создаваемой анимацией. Следовательно, в определённом смысле описанный метод можно считать обобщением PSD. Однако, по сравнению с PSD, метод контрольных кластеров позволяет указывать эталонные положения не для всех вершин модели, а лишь для проблемных областей модели, что позволяет существенно уменьшить объём работ, выполняемых аниматором. Кроме того, описанный подход позволяет использовать малое количество пар <конфигурация скелета, набор эталонных вершин>, что также способствует снижению трудоёмкости процесса подготовки модели к анимации.

#### 5. ПРИМЕР ИСПОЛЬЗОВАНИЯ МЕТОДА

В качестве демонстрационного примера рассматривается плоская модель цилиндра (упрощенная модель “руки”) с оснасткой (см. рис. 2). Очевидно, проблемной зоной модели является область сгиба цилиндра (т.е. “локтя”). В качестве контрольного кластера выбираются точки, расположенные в области “сочленения” двух костей, точнее говоря, с формальной точки зрения, в области оси вращения, в данном случае совпадающей со второй костью; вершины в этой области имеют веса привязки к костям (0.5, 0.5).

Далее задаются пары <конфигурация скелета, набор контрольных кластеров модели>. Для данной демонстрационной модели достаточно задать всего одну такую пару. Для этого можно воспользоваться эталонным положением вершин кластера, взятым из PSD-формы модели.

Далее необходимо задать значение параметра натяжения  $q$ . При значениях  $q > 0$  сгиб локтя выглядит “угловатым” и неестественным (см. рис. 3.1а). При отрицательных значениях параметра  $q$  сегменты кривой характеризуются большей

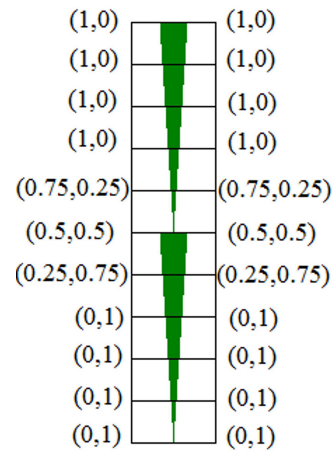


Рис 2: Модель цилиндра со скелетом из двух костей. В скобках указаны веса костей относительно каждой вершины модели по схеме (<вес родительской кости>, <вес дочерней кости>).

кривизной, что позволяет добиться большей реалистичности (см. рис. 3.1б,в). При этом для значений параметра  $q$  близких к  $-1$  модель демонстрирует потерю объёма при определённых положениях дочерней кости (см. рис. 3.2в). Как следствие,  $q = -0.5$  принимается в качестве значения параметра натяжения по умолчанию.

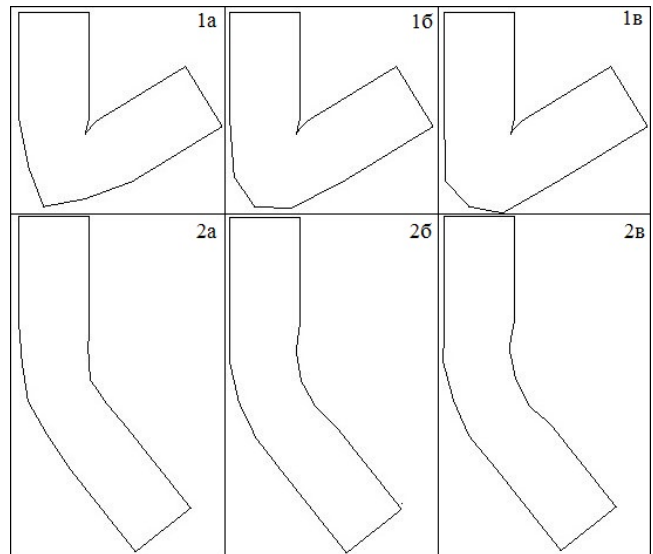
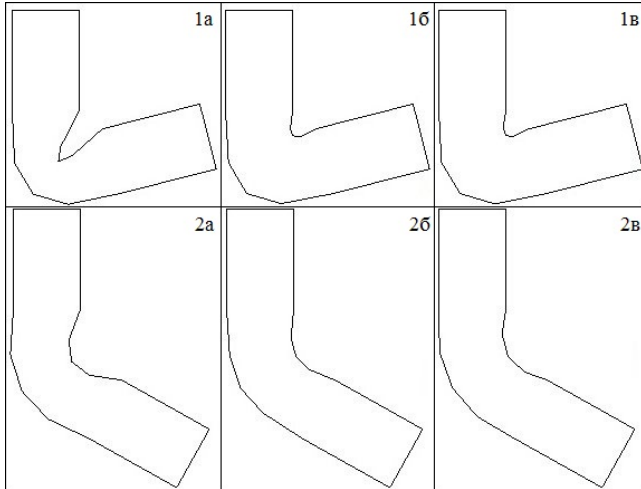


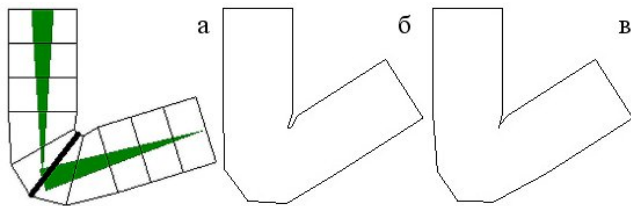
Рис 3: Метод контрольных кластеров с различными значениями параметра натяжения  $q$ : 1а, 2а –  $q = 0.5$ ; 1б, 2б –  $q = -0.5$ ; 1в, 2в –  $q = -1.0$ .

Для сравнения результатов работы LBS, PSD и метода контрольных кластеров была создана анимация с вращением дочерней кости модели. Рассматривается модель с двумя конфигурациями скелета модели (см. рис. 4). Потеря объёма при отработке сгиба методом LBS видна как в первом, так и во втором случае. В то же время технология PSD и метод контрольных кластеров не демонстрируют подобных искажений. При этом при угле сгиба, превышающем угол сгиба в конфигурации скелета, соответствующей PSD-форме

модели, PSD будет проявлять нежелательные артефакты, наследуемые от LBS. Для их устранения будет необходимо смоделировать ещё одну PSD-форму, соответствующую большому сгибу. Метод контрольных кластеров (МКК) при таких условиях будет проявлять большую гибкость: в месте сгиба будет создаваться *точка возврата* (точка, в которой кривая изменяет своё направление), что характерно для антропоморфных моделей, на которые рассчитан метод (см. рис. 5).



**Рис 4:** Результаты работы методов скелетной анимации при обработке сгиба: 1а, 2а — LBS; 1б, 2б — PSD; 1в, 2в — метод контрольных кластеров.



**Рис 5:** Результаты поворота дочерней кости на больший угол. а — эталонная PSD-форма с контрольным кластером, выделенным жирным; б — PSD; в — метод контрольных кластеров.

Таким образом, метод контрольных кластеров позволяет достичь результатов, сравнимых с результатами PSD, используя меньший объём вводимых аниматором данных. Для получения приемлемой анимации рассматриваемой модели методом PSD требуется задать минимум 2 PSD-формы и, соответственно, эталонное положение 6 вершин для каждой из 2 конфигураций скелета; в то время как при использовании МКК достаточно задать эталонное положение 2 вершин для одной конфигурации скелета. То есть, при использовании МКК для данной демонстрационной модели аниматору требуется задать в 6 раз меньше данных, чем при использовании PSD.

## 6. ЗАКЛЮЧЕНИЕ

Предложенный метод контрольных кластеров (МКК) для анимации моделей антропоморфных персонажей является корректирующим методом деформации моделей, полученным на основе LBS, и развивающим идеи PSD.

Введено оригинальное понятие контрольного кластера, на основе которого разработан способ компенсации дефектов трансформации, получаемой при применении базового метода LBS. Метод контрольных кластеров требует, в отличие от PSD, создания и хранения эталонных форм кластеров только для проблемных областей модели. Практическая значимость разработанного метода заключается в существенном сокращении объёма работ, выполняемых аниматором. За счёт большей гибкости по сравнению с PSD метод контрольных кластеров позволяет получить анимацию, сравнимую по качеству с результатами PSD, при использовании меньшего количества вводимых аниматором данных.

Предложенный метод разработан для двумерных моделей. Обобщение данного подхода для случая трёхмерных полигональных моделей, а также способы задания контрольных кластеров для таких моделей требуют дополнительных исследований.

## 7. ССЫЛКИ

- [1] Букатов А. А., Гридчина Е. Е., Заставной Д. А. Методы скелетной анимации для трансформации полигональных поверхностей трёхмерных моделей [Электронный ресурс] // *Инженерный вестник Дона*. — 2012. — № 3. — С. 59–74. [Электронный журнал]. URL: [http://www.ivdon.ru/uploads/article/pdf/2012\\_3\\_11.pdf\\_897.pdf](http://www.ivdon.ru/uploads/article/pdf/2012_3_11.pdf_897.pdf) (дата обращения 10.06.2013).
- [2] Кузнецов Е. Б., Якимович А. Ю. Наилучшая параметризация в задачах приближения кривых и поверхностей // *Ж. вычисл. математ. и математич. физики*. — 2005. — Т. 45, № 5. — С. 760–774.
- [3] E. Catmull and R. Rom, “A class of local interpolating splines,” *Computer Aided Geometric Design*, New York: Academic Press, 1974, pp. 317–326.
- [4] A. Jacobson, I. Baran, L. Kavan, J. Popović, and O. Sorkine, “Fast automatic skinning transformations,” *ACM Trans. Graph.*, vol. 31, no. 4, pp. 77:1–77:10, Jul., 2012.
- [5] G. S. Lee and F. Hanner, “Practical experiences with pose space deformation,” in *ACM SIGGRAPH ASIA 2009 Sketches*, New York: ACM, 2009, pp. 43:1–43:1.
- [6] J.P. Lewis, M. Cordner, and N. Fong, “Pose space deformation: a unified approach to shape interpolation and skeleton driven deformation,” in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, New York: ACM Press/Addison-Wesley Publishing, 2000, pp. 165–172.
- [7] R. Vaillant, L. Barthe, G. Guennebaud, M. Cani, D. Rohmer, B. Wyvill, O. Gourmel, and M. Paulin, “Implicit skinning: real-time skin deformation with contact modeling,” in *Proc. of ACM SIGGRAPH 2013*. Forthcoming. Jul., 2013.
- [8] C. Yuksel, S. Schaefer, and J. Keyser, “On the parameterization of Catmull-Rom curves,” in *ACM Joint Conference on Geometric and Physical Modeling (SPM '09)*, New York: ACM, 2009, pp. 47–53.

# Сжатие геометрической информации и оценка близости сложных объектов на основе порождающих моделей

Грудинин Сергей Николаевич, Фроловский Владимир Дмитриевич  
Новосибирский государственный технический университет, Новосибирск, Россия  
sn\_grudinin@mail.ru

## Аннотация

В статье рассматриваются вопросы моделирования компьютерных манекенов. Для моделирования сложных поверхностей используется каркасная модель с параметризацией по сечениям. В качестве параметров выступают продольные и поперечные диаметры. Рассмотрен метод извлечения и представления знаний о сложном геометрическом объекте.

**Ключевые слова:** параметризация, манекен, порождающие модели

## 1. ВВЕДЕНИЕ

Под геометрической информацией моделируемого объекта понимается [3]: набор параметров  $\{\rho\}$ , характеризующих положение объекта в пространстве, набор параметров  $\{\mu\}$  – метрических характеристик формы объекта и метод формирования изображения объекта по параметрам  $\{\mu\}$ .

После определения геометрической информации, становится возможным построение множества моделей объекта, изменяя параметры  $\{\mu\}$  и  $\{\rho\}$  в установленных границах. Этот подход называется концепцией порождающих моделей [4], при этом геометрическую информацию объекта называют базовой моделью, а множество полученных моделей – производными моделями.

Определение базовой модели для объектов сложной формы, таких как манекен, проходит в два этапа: выбор топологии модели (например, каркасная, точечная, объемная и т.п.) и ее параметризация. При этом параметрическое описание базовой модели с одной стороны, должно наилучшим образом отражать особенности формы объекта, с другой стороны содержать разумное число легко определяемых параметров [5].

В связи с этим существует задача сжатия геометрической информации – наиболее компактного и удобного параметрического описания базовой модели.

Таким образом, для моделирования объектов методом порождающих моделей необходимо:

- 1) на основе данных о форме объекта разработать представление и параметризацию базовой модели;
- 2) заложить в параметризацию возможность деформации формы при изменении параметров для дальнейшего проектирования производных моделей;
- 3) для определения качества моделирования задать критерий сравнения моделей как по параметрам, так и с учетом особенностей формы.

В работе рассматривается модель женского манекена. Форма модели – портновский манекен – правильный торс без головы и конечностей.

## 2. БАЗОВАЯ МОДЕЛЬ

### 2.1 Исходные данные

На старте процесса моделирования входными данными является трехмерная сканированная модель объекта. В качестве исходных сечений для моделирования базовой модели выставляются координаты сканированных точек на заданных уровнях  $l_i$ ,  $i = \overline{1, N}$ . Выбор уровней сечений осуществляется экспертом на основе антропометрических стандартов или методов проектирования промышленных манекенов [2].

Так как по природе сканирование – автоматический процесс, возможен избыток или искажение информации, поэтому каждое сечение проходит предобработку: удаление повторных и лежащих на одной прямой точек, сглаживание В-сплайновой кривой.

### 2.2 Представление

Представлением базовой модели выбрана трехмерная каркасная модель с топологией: множество вершин  $\{v_{ij}\}$ , множество соединяющих их вертикальных  $\{(v_{ij}, v_{ij+1})\}$  и горизонтальных ребер  $\{(v_{ij}, v_{i+1j})\}$ ,  $j = \overline{1, M}$ ,  $M$  – количество точек в сечениях. Вершины располагаются на уровнях  $l_i$ . Множество вершин одного уровня образуют сечение  $h_i = \{v_i\}$ . Сечение содержит одинаковые количества вершин, отсортированных по полярному углу и равноотстоящих друг от друга.

Выбранное представление накладывает два ограничения. Во-первых, параметризация базовой модели сводится к параметризации отдельных сечений. Во-вторых, для получения в любой момент моделирования равноотстоящих друг от друга вершин целесообразнее рассматривать сечение в параметризованном виде, как зависимость от суммарной длины. То есть табличную функцию  $y(x)$ , описывающую сечение, выразить как  $x(s)$  и  $y(s)$ , где  $s$  выступает накопленная (суммарная) длина, то есть при  $s_1 = 0$ ,

$$s_j = s_{j-1} + \sqrt{(x_j - x_{j-1})^2 + (y_j - y_{j-1})^2}, j = \overline{2, M}$$

Далее под  $f(s)$  понимается одна из функций  $x(s)$  и  $y(s)$ .

После предобработки происходит размещение в пространстве исходных сечений, представления их в параметризованном виде и симметризация.

### 2.3 Размещение в пространстве

Сечение  $h_i$  располагается в плоскости  $x'_i O_i y'_i$  параллельной плоскости  $xOy$  глобальной системы координат. Положение сечения определяется полюсом  $O_i(x_i, y_i, z_i)$  и углом  $\theta_i$  между проекцией оси абсцисс собственной системы координат  $x'_i O_i y'_i$  на плоскость  $xOy$ , и осью абсцисс системы координат  $xOy$  (см. рисунок 1). За полюс принимается проекция шейной

точки (основание шеи сзади) на плоскость сечения  $h_i, z_i$  равен высоте соответствующего сечения. Исходные сечения подвергаются преобразованиям поворота и переноса так, что  $\theta_i = x_i = y_i = z_i = 0$ .

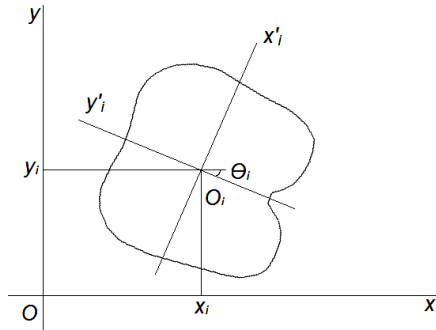


Рисунок 1: Положение сечения на плоскости

Таким образом, параметрами, характеризующими форму объекта, являются значения координат  $z_i$  и  $x_i$  для каждого сечения.

### 2.4 Симметризация

Одним из базовых свойств манекена является симметричность относительно плоскости  $xOz$ . Оно позволяет рассматривать полусечения, что сокращает расчеты вдвое, однако исходные сечения несимметричны, поэтому они подвергаются симметризации – определению средних значений между левыми и правыми половинами. Для этого правые части зеркально отображаются относительно  $Ox_i$ . Значения симметричного сечения рассчитываются как средние кусочно-линейной функции  $f(s)$  левой и правой стороны, вычисленной по всей длине с заданным интервалом  $\Delta s$  (см. рисунок 2).

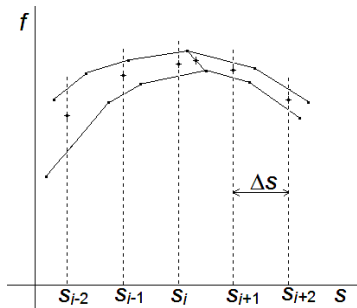


Рисунок 2: Определение симметричного сечения

Вершины, в которых  $f_{i-1} < f_i$  и  $f_{i+1} < f_i$ , называются характерными, они важны при параметризации. Средние значения характерных вершин определяются как середина отрезка, соединяющего соответствующие характерные вершины левой и правой стороны (см. рисунок 2). Предполагается, что левая и правая сторона сечения достаточно близки по форме, поэтому количество характерных вершин в них совпадает.

### 3. ПАРАМЕТРИЗАЦИЯ

После симметризации исходные сечения подвергаются параметризации. В качестве метрических характеристик формы сечения, рассматриваются продольные и поперечные диаметры, на рисунке 3 это отрезки  $a, b, c, d$ .

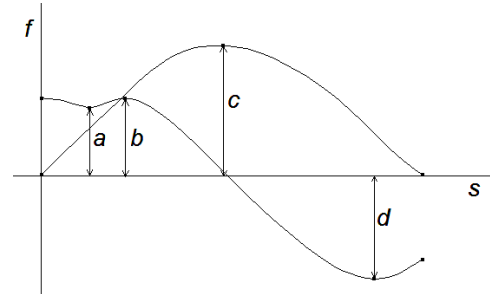


Рисунок 3: Пример метрических характеристик половины сечения

Сегменты сечений между характерными вершинами представляются в виде полинома третьей степени [1]:

$$f(s) = C_1 + C_2(s - s_1) + C_3(s - s_1)^2 + C_4(s - s_1)^3,$$

$$C_1 = f_1, C_2 = a_1, C_3 = (\Delta f - a_1)/\Delta s - C_4\Delta s,$$

$$C_4 = (a_1 + a_2 - 2\Delta f)/\Delta s^2, \Delta f = (f_2 - f_1)/\Delta s, \Delta s = s_2 - s_1,$$

где  $a_1, a_2$  – угловые коэффициенты в начальной и конечной точках сегмента соответственно,  $(f_1, s_1), (f_2, s_2)$  – координаты начальной и конечной точек сегмента соответственно.

Значения  $a_1$  и  $a_2$  рассчитываются для каждого сегмента отдельно по формуле:

$$a = (f_{j-1} - f_j)/(s_{j-1} - s_{j+1}),$$

если характерная вершина крайняя справа (слева), то по формуле:

$$a = (f_j - f_{j-1})/(s_j - s_{j-1}),$$

где  $(f_i, y_i)$  – координаты соответствующей характерной вершины,  $(f_{i-1}, y_{i-1}), (f_{i+1}, y_{i+1})$  – координаты вершин, соседствующих с характерной.

Для уменьшения погрешности восстановленных по параметрам сечений для каждого сегмента определяются оптимальные значения  $a_1$  и  $a_2$ . Эта задача решается методом градиентного спуска, в качестве условия остановки используется среднеквадратичное отклонение  $E$  расстояний между исходным и восстановленным сечением.

Алгоритм поиска оптимальных угловых коэффициентов выглядит следующим образом:

- 1) задается начальное приближение  $a$ , шаг  $step$  и ошибка  $\zeta$ ;
- 2) задается  $a_{j+1} = a_j + step$ ;
- 2) определяется  $E_{j+1}, E_j$ ;
- 3) если  $(E_{j+1} - E_j) < \zeta$ , то остановка, иначе переход к шагу 4;
- 4) определяется  $grad = (E_{j+1} - E_j)/(a_{j+1} - a_j)$ . Если  $E_{j+1} > E_j$ , то  $\lambda = -\lambda/2$ ;
- 5) определяется шаг  $step = \lambda \cdot grad$ . Переход к шагу 2,  $j = j + 1$ .

На рисунке 4 представлено сечение на уровне наиболее выступающей точки живота до (сплошная линия) и после (пунктирная линия) оптимизации, параметры оптимизации:  $\zeta = 0.0001$ ,  $step = 0.1$ ,  $\lambda = 0.01$ .

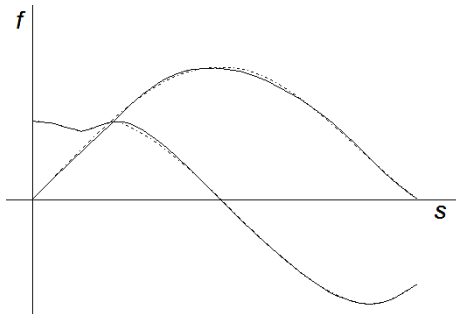


Рисунок 4: Половина сечения до оптимизации и после

Оптимизации угловых коэффициентов – заключительный этап параметризации базовых сечений. Описанная параметризация позволяет деформировать сечения, изменяя продольные или поперечные диаметры. Это приводит к изменению координат характерных вершин, на основе которых происходит перерасчет сегментов, т.е. деформация всего сечения. На рисунке 5 представлен пример деформации полусечения путем изменения одного продольного и трех поперечных диаметров.

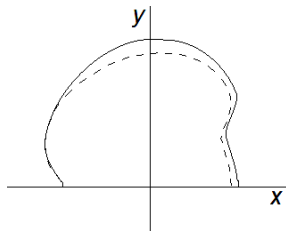


Рисунок 5: Пример деформации полусечения

#### 4. СРАВНЕНИЕ ОБЪЕКТОВ

Набор параметров характеризующих  $i$ -ое полусечение включает:

- координаторы  $z_i$  и  $x_i$  полюса  $O_i$ ;
- наибольший продольный  $L_i$  и поперечный  $T_i$  диаметры полусечения.

В связи с этим в качестве критериев оценки близости двух моделей  $\Phi'$  и  $\Phi$  используются среднеквадратичные смещения по осям  $Oz$  и  $Ox$ :

$$S_z = \frac{1}{N} \sum_i \sqrt{(z'_i - z_i)^2}, S_x = \frac{1}{N} \sum_i \sqrt{(x'_i - x_i)^2}, i = \overline{1, N},$$

и средняя разность отношений продольного и поперечного диаметров.

$$R = \frac{1}{N} \sum_i \left( \frac{L'_i}{T'_i} - \frac{L_i}{T_i} \right), i = \overline{1, N},$$

Для оценки схожести форм полусечений вводятся дополнительные ограничения:

- 1)  $x$ -координата первой и последней вершин равна 0;
- 2) содержит не менее трех точек.

В качестве критерия оценки используется коэффициент  $k$ , характеризующий отличие формы от полуокружности, описанной вокруг полусечения (см. рисунок 6).

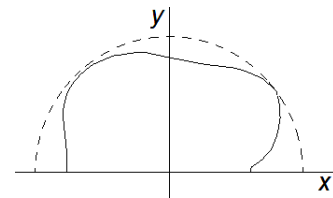


Рисунок 6: Пример описанной полуокружности

Полуокружности соответствуют всем описанным выше ограничениям полусечений, количество вершин в них совпадает с количеством вершин в полусечениях.

Коэффициент  $k$  определяется следующим образом:

$$k = \frac{1}{m} \sum_i \left( \frac{|A_i \cos \varphi_i|}{|B_i|} \right), i = \overline{1, m},$$

где  $m$  – количество вершин в полусечении,  $A_i$  – радиус-вектор с концом в  $i$ -ой вершине полусечения,  $B_i$  – радиус-вектор с концом в  $i$ -ой вершине полуокружности,  $\varphi_i$  – угол между  $A_i$  и  $B_i$ . Значение коэффициента  $k$  изменяется в интервале от 0 до 1. Для оценки близости двух моделей  $\Phi'$  и  $\Phi$  рассчитывается значение  $|k' - k|$ , также изменяющееся от 0 до 1, чем оно ближе к 0, тем более модели схожи между собой.

#### 5. ЗАКЛЮЧЕНИЕ

В статье описан алгоритм получения базовой модели заданной структуры по трехмерным сканам исходного объекта. Рассмотрена оригинальная параметризация модели трехмерного объекта со сложной структурой на основе диаметров поперечных сечений, сохраняющая геометрическую информацию об объекте. Такой подход позволяет, изменяя диаметры сечений, деформировать базовую модель. Авторами предложены критерии сравнения схожести каркасных моделей данного представления как по параметрам, так и по форме сечений.

Перспективным развитием рассмотренных решений является определение границ изменения параметров, характеризующих тело человека, и их взаимосвязь как между собой, так и с параметрами, определяющими форму сечений. Оптимизация критериев оценки схожести сечений. Определение параметров сечений по фотографиям. Построение патч-модели на основе базовой модели с использованием продольно-вертикальных и поперечно-вертикальных сечений, с заполнением четырехугольной сетки поверхностями определенного типа [6].

#### 6. ЛИТЕРАТУРА

- [1] Голованов Н. Н., Ильютко Д. П., Носовский Г. В. Компьютерная геометрия. М. Издательский центр «Академия». Сер. Прикладная математика и информатика. 2006. -512 с.



[2] Коблякова Е. Б. Т. Н. Дунаевская, Г. С. Ивлева, Р. В. Ивлева Размерная типология населения с основами анатомии и морфологии. М. Мастерство. Издательский центр «Академия». 2001. – 288 с.

[3] Стоян Ю. Г., Яковлев С. В. Математические модели и оптимизационные методы геометрического проектирования. Киев. Наук. Думка. 1986.- 268 с.

[4] Фроловский В. Д. Избранные задачи геометрического проектирования. Параметризация сложных поверхностей. Новосибирск. Изд-во «Прогресс-сервис». 2005. -165 с.

[5] Фроловский В. Д. Компьютерное проектирование манекенов и одежды. Трехмерные модели и математические методы. Издательский Дом: LAP LAMBERT Academic Publishing. Saarbrucken, Germany. 2012. 269 с.

[6] Wang C C. L. Parameterization and parametric design of mannequins, Computer-Aided Design 37, pp. 83-98, 2005.

### **Об авторах**

Грудинин Сергей Николаевич, аспирант кафедры АСУ НГТУ.  
email: sn\_grudinin@mail.ru

Фроловский Владимир Дмитриевич, д.т.н., профессор  
кафедры АСУ НГТУ. email: frolovsky@asu.cs.nstu.ru

# Выделение контуров зданий и распознавание служебных символов для трехмерной реконструкции объектов городской обстановки по топографическому плану

Игорь Соловьёв

Школа естественных наук, Дальневосточный федеральный университет, Владивосток, Россия

ИАПУ ДВО РАН, Владивосток, Россия

[igorek.solovyev@mail.ru](mailto:igorek.solovyev@mail.ru)

## Аннотация

В данной работе рассматривается метод реконструкции объектов городской обстановки по топографическому плану. Предлагается новый алгоритм распознавания служебных символов на топлане, содержащих информацию о типе здания и количестве этажей. Также, предлагается модификация волнового алгоритма, используемого для нахождения границ здания.

**Ключевые слова:** Распознавание, Реконструкция, Топоплан.

## 1. ВВЕДЕНИЕ

Создание трёхмерных моделей реальных объектов остаётся актуальной проблемой компьютерной графики и машинного зрения. Одной из важных практических задач является реконструкция объектов городской обстановки.

На сегодняшний день уже разработан ряд методов, направленных на решение поставленной задачи [3]. Одним из таких методов является реконструкция городского пространства по топографическому плану. При таком подходе необходимо определить геометрическую форму основания здания, а также распознать надпись содержащую информацию о типе здания и количестве этажей. Автоматическое выделение контура здания производится с помощью метода, описанного в [2]. Получение информации об этажности производится путём сравнения выделенной надписи с базой эталонов.

## 2. НАЧАЛЬНЫЕ ДАННЫЕ

Топоплан представляет собой чёрно-белое изображение, содержащее контуры зданий, улиц, информацию о количестве этажей зданий, данные о ландшафте, различные городские объекты и т.д.

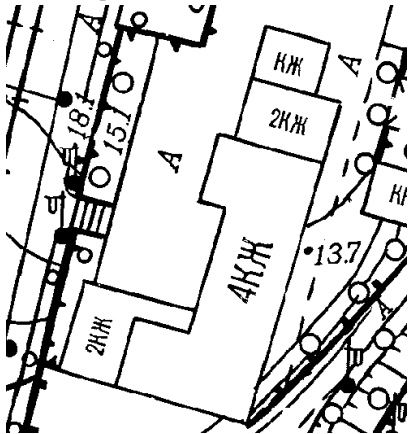


Рис. 1: Топографический план.

Преобразуем данное изображение в матрицу  $M$ , заполняя её нулями и единицами. Ноль соответствует белому цвету на топлане, а единица – чёрному.

## 3. ВОЛНОВОЙ АЛГОРИТМ

В рассматриваемом методе используется волновой алгоритм [1], однако скорость его выполнения не так высока, поэтому скорость работы распознавания также будет небольшой, особенно при анализе топлана больших размеров. Поэтому для интерактивной работы приложения существует необходимость в ускорении волнового алгоритма.

Использование данного алгоритма имеет одно преимущество. После его работы получаются данные о контуре и служебной информации. Причём эти данные едины, т.е. нет необходимости в том, чтобы устанавливать соответствия между символами и контуром, к которому они относятся.

### 3.1 ОПИСАНИЕ ВОЛНОВОГО АЛГОРИТМА

Волновой алгоритм довольно прост в реализации. Пусть на некоторой прямоугольной сетке все элементы равны либо 0 (свободные клетки), либо 1 («стены»). Выбирается любая свободная ячейка и ей присваивается значение  $t_0 > 1$ . Далее, сканируется вся сетка и если находится клетка со значением  $t_0$ , то проверяются её соседи. Каждой пустой соседней ячейке присваивается значение  $t_0 + 1$ . Если какая-нибудь соседняя ячейка является «стеной», то ей присваивается значение  $-1$ . Далее, вся сетка сканируется вновь, только на этот раз ищутся ячейки со значением  $t_0 + 1$  и т.д. Алгоритм повторяется до тех пор, пока в замкнутой области, окруженной «стенами» не останется пустых ячеек.

Скорость работы такого алгоритма напрямую зависит от числа ячеек. Если количество элементов сетки равно  $N$ , то за  $t$  шагов волновому алгоритму придётся проверить  $N \cdot t$  клеток. В худшем случае, при большом количестве пустых ячеек и одинаковых размерах сетки, число элементов будет равно  $N = (2t + 1)^2$ . Тогда алгоритму придётся обработать  $S_1$  ячеек:

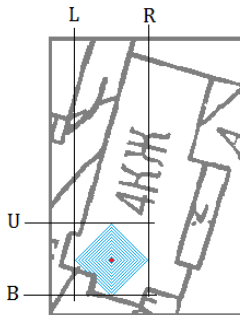
$$S_1 = (2t + 1)^2 t = 4t^3 + 4t^2 + t \quad (1)$$

Как правило, волновой алгоритм применяется для задач поиска кратчайшего пути в лабиринте. В нашем случае, путь не играет никакого значения, нам важно только воздействие алгоритма на рабочую область.

### 3.2 МОДИФИКАЦИЯ АЛГОРИТМА

Модификация алгоритма использует тот факт, что при поиске очередной ячейки нет необходимости в сканировании всей сетки, достаточно только проверить

соседние элементы уже рассмотренных ячеек. Таким образом, ускорить время работы алгоритма можно при помощи четырёх дополнительных переменных **L**, **T**, **R**, **B**, которые являются соответственно левой, верхней, правой и нижней границами некоторого окна поиска. Каждая переменная имеет значение минимального или максимального номера клетки по своему направлению (номера **Li** и **Ri** имеют значения минимального и максимального номеров по горизонтали, а номера **Ti** и **Bi** - по вертикали). Таким образом, требуется сканирование не всей сетки, а только лишь окна размером  $(R-L) \cdot (B-T)$ . Если на каком-то шаге волны обнаруживается пустая ячейка, то она проверяется на то, выходит ли она за рамки окна, если так, то соответственная граница сдвигается.

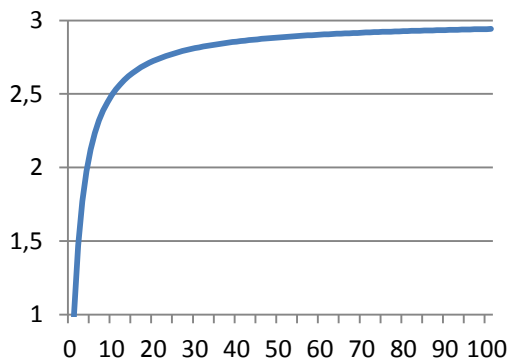


**Рис2:** Процесс распространения волны в окне.

Худшим вариантом для алгоритма будет ситуация, при которой начальной точку  $t_0$  будет окружать множество пустых ячеек. В этом случае придётся сканировать окно размером  $3 \times 3$ , на следующем шаге  $5 \times 5$  и т.д. На шаге  $t$  окно сканирования будет иметь размеры  $(2t+1) \times (2t+1)$ . Таким образом максимальное число ячеек  $S_2$ , которое необходимо просканировать будет определяться по формуле

$$S_2 = \sum_{k=1}^t (2k+1)^2 = \frac{4t^3 + 12t^2 + 11t}{3} \quad (2)$$

Ускорение, которое будет давать модифицированный алгоритм, определится как  $a = S_1/S_2$ .



**Рис. 3:**График ускорения работы модифицированного алгоритма. Слева расположено ускорение, снизу – высота и ширина топоплана.

Таким образом, применение модификаций к волновому алгоритму позволяет ускорить алгоритм в 3 раза. Это особенно полезно при работе с сеткой больших размеров.

#### 4. ВЫДЕЛЕНИЕ СЛУЖЕБНОЙ ИНФОРМАЦИИ

На топоплане информация о количестве этажей здания находится внутри его контура. Прежде, чем приступить к распознаванию этой информации, её необходимо выделить из всего набора тех данных, что находится на изображении.

Выберем любую нулевую точку матрицы **M**, причем так, чтобы она находилась внутри контура, образованного единичными элементами. Из этой точки запустим волновой алгоритм. После его работы матрица **M** изменится следующим образом:

- элементы, совпадающие с границей контура, будут равны -1;
- элементы, находящиеся на границе служебной информации, также будут равны -1;
- все точки, лежащие внутри контура и не принадлежащие служебной информации, будут равны числам больше 1.

Имеется два набора данных, отмеченных числом -1, поэтому необходимо их разделить. Применим к матрице **M** несколько фильтров, чтобы оставить только точки, принадлежащие служебной информации.

Первый фильтр удаляет точки границы контура. Для этого построчно перебираются все элементы матрицы **M**. Если найден элемент со значением -1, то он будет являться источником для расширенного волнового алгоритма по контуру. Действие такого алгоритма будет отличаться от рассмотренного выше следующими положениями:

- волна распространяется не по четырем направлениям, а по восьми;
- «пустыми» клетками будут клетки со значением -1;
- «стенами» будут все ячейки, значения которых больше нуля.

Результатом применения алгоритма будет отсутствие в матрице **M** элементов границы контура. Таким образом, все элементы матрицы равные -1 будут именно точками служебной информации.

Далее, перебираются все элементы матрицы. Если значение ячейки не равно -1, то из неё будет запущен волновой алгоритм, работающий по следующему принципу:

- волна распространяется по четырем направлениям;
- «пустыми» клетками будут те, значение которых не равно -1;
- «стенами» будут все ячейки, значения которых равно -1.

После применения такого алгоритма в матрице **M** образуются элементы со значением больше 1. Такие элементы никак не относятся к служебной информации. Проведённое действие позволило отделить нулевые элементы внутри служебной информации от тех, которые лежат вне её. Таким образом, элементы матрицы **M** равные 0 и -1 будут тем набором данных, который необходимо распознать как текст. Для простоты, изменим значения ячеек следующим образом:

- Если значение элемента равно 0 или -1, то ей присваивается значение 1;
- Если значение элемента больше 1, то ей присваивается значение 0.

Требуемый набор данных сформирован.

## 5. РАЗБИЕНИЕ НА СИМВОЛЫ

Перед распознаванием, полученный набор данных требуется разбить на отдельные символы. Для этого поочерёдно перебираются все элементы матрицы  $M$ . Если элемент равен 1, то из него запускается волновой алгоритм. «Пустыми» клетками для него будут клетки со значением 1, а «стенами» - клетки со значением 0. После этого, все элементы, значение которых больше 1, становятся частью другой матрицы  $J_0$ . Размеры такой матрицы соответствуют размеру области действия алгоритма. Элементы матрицы  $M$  со значением больше 1 меняют значение на 0. Повторяется процесс поиска единичного элемента. После его отыскания снова запускается волновой алгоритм и формируется матрица  $J_1$ . Таким образом создаются матрицы символов  $J_k$ , где  $k$  принимает значения от 0 до  $w-1$ .

## 6. ФОРМИРОВАНИЕ БАЗЫ ЭТАЛОНОВ

Каждым эталоном является один из символов служебной информации, который находится под одним из возможных угловповорота. Пусть топоплан содержит  $n$  служебных символов. В данном случае служебными символами будут именно те, которые находятся внутри контуров зданий. Ориентация топоплана может быть различной, следовательно ориентация символов тоже. В связи с этим количество вариантов каждого символа будет определяться формулой:

$$z = \text{round}\left(\frac{360^\circ}{\alpha}\right), \quad (3)$$

где  $\alpha$  – минимальный угол поворота одного символа, выраженный в градусах. Каждый  $i$ -й вариант эталона будет повернут на угол  $\alpha \cdot i$ , где  $i$  принимает значения от 0 до  $z-1$ .

База эталонов будет представлять из себя матрицу  $V$  с размерами  $n \cdot z$ . Каждый элемент такой матрицы содержит в себе данные об эталоне.  $V_{ij}$  также является матрицей, где  $i$  принимает значения от 0 до  $n-1$ , а  $j$  – от 0 до  $z-1$ . Такая матрица заполнена 0 и 1, 0 соответствует чёрному цвету эталонного изображения, а 1 – чёрному.

0123456789ЖКМНС/

Рис. 4: Эталон служебных символов.

## 7. РАСПОЗНАВАНИЕ

Распознавание матрицы  $J_k$  будет производиться путём сравнения её с каждым эталоном из  $V$ . Пусть матрица  $J_k$  имеет размеры  $W \times H$ , а размеры матрицы эталона  $V_{ij}$  равны  $F \times G$ . Тогда между координатами элемента матрицы  $J_k$  и координатами матрицы эталона существует соотношение в виде:

$$\begin{aligned} x &= \frac{W}{F} \bar{x} \\ y &= \frac{H}{G} \bar{y} \end{aligned} \quad (4)$$

где  $x, y$  – координаты некоторого элемента матрицы  $J_k$ , а  $\bar{x}, \bar{y}$  – соответствующие им координаты матрицы эталона  $V_{ij}$ . Если элементы матриц с такими координатами равны, то процент их схожести увеличивается. Таким образом, перебираются все эталоны и выбирается тот, чей процент схожести выше. Процент схожести определяется путём поэлементного сравнения матриц:

$$P = \frac{d}{W \cdot H} * 100\%, \quad (5)$$

где  $P$  – процент схожести,  $W, H$  – соответственно ширина и высота символа  $J_k$ ,  $d$  – количество элементов, для которых справедливо равенство:

$$J_{xy} = V_{\bar{x}\bar{y}} \quad (6)$$

При таком подходе, число сравнений равно произведению  $n \cdot z \cdot w$ , где  $n$  – число эталонов,  $z$  – число вариантов одного эталона,  $w$  – количество символов, которое требуется распознать.

## 8. РЕЗУЛЬТАТЫ

Представленный алгоритм был применён на топоплане рис. 5. Размер изображения 1550x1222 пикселей. Топоплан содержит 137 символов той служебной информации, которую необходимо распознать. Служебная информация состоит из 16 различных символов. Средние размеры всех эталонов 25x36 пикселей.

Результаты работы алгоритма представлены в таблице 1.

Таблица 1.

$\alpha$ , градусы	Время загрузки, секунды	Время распознавания, секунды	Правильно распознано	%
0,1	543	381,8	135	98,5
1	59,8	44,2	132	96,3
2	29,4	23,5	129	94,1
3	24,2	16,0	126	91,9
4	14,7	13,2	121	88,3
5	11,4	9,1	115	83,9
6	11,2	8,9	113	82,4
10	7,1	7,4	104	75,9
30	5,2	5,5	98	71,5

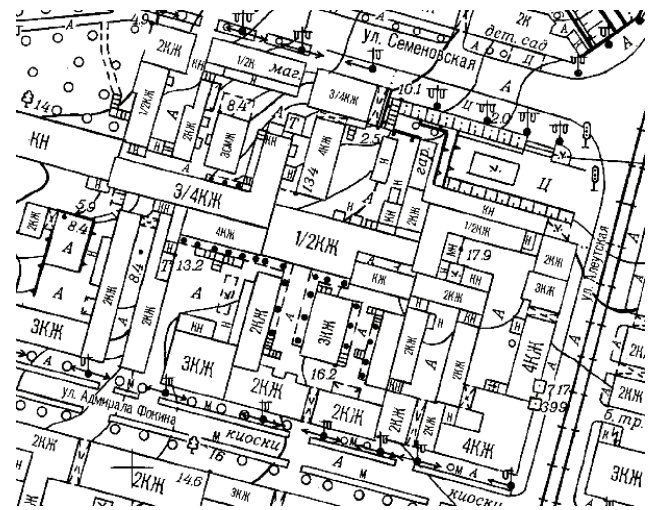
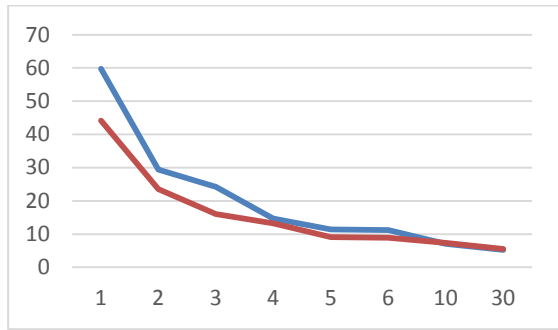


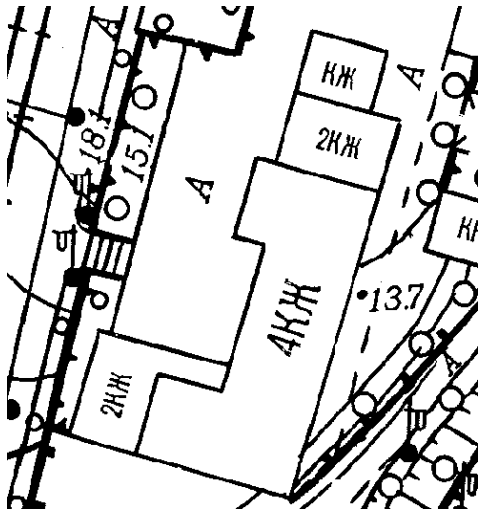
Рис. 5: Топографический план №1.

Полученные результаты позволяют говорить об эффективности метода. Даже при малом количестве эталонов в базе, количество верно распознанной информации больше 70%. Трудностью для алгоритма является определение таких символов, как «6» и «9», «К» и «Н». Однако, правило записи

информации на топоплане позволяет легко уточнить полученный результат.

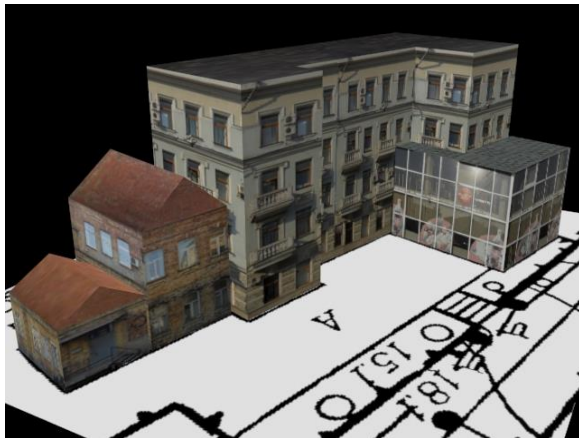


**Рис. 6:**Время загрузки базы эталонов (синяя линия) и время распознавания символов (красная линия). Подпись снизу – угол поворота (в градусах), подпись справа – время (в секундах)

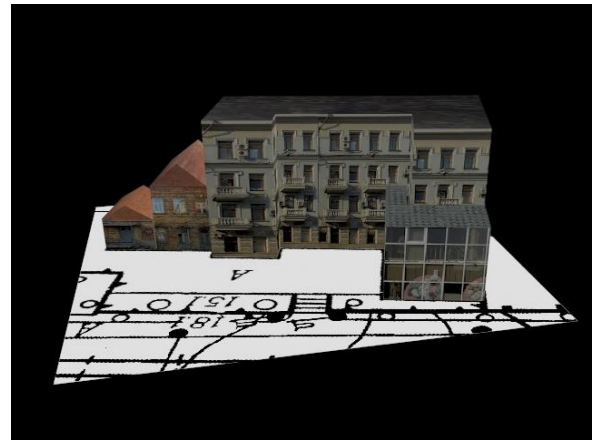


**Рис. 7:** Топографический план №2.

Предложенный алгоритм был применён для реконструкции реальной сцены по топографическому плану, изображенного на рисунке 1. Время загрузки базы эталонов составило 26,2 секунды при  $\alpha=2^\circ$ . Время выделения контуров заняло менее секунды.



**Рис. 8.а:** Реконструкция реальной сцены. Вид №1.



**Рис. 8.б:** Реконструкция реальной сцены. Вид №2.

Приведённые результаты были получены на машине со следующими характеристиками: IntelCorei3-2320VCPU 2200 GHz, 8 ГБ ОЗУ.

## 9. ЗАКЛЮЧЕНИЕ

Предложенные алгоритмы по выделению контуров показали свою эффективность, поэтому могут быть использованы для решения прикладных задач, связанных не только с реконструкцией, но и с другими областями компьютерного зрения.

Рассмотренные алгоритмы являются частью системы, задача которой состоит в реконструкции объектов городской обстановки по топографическому плану.

## 10. ССЫЛКИ

- [1] Lee, C.Y., «An Algorithm for Path Connections and Its Applications», IRE Transactions on Electronic Computers, vol. EC-10, number 2, pp. 364—365.
- [2] Бобков В.А., Соловьёв И.В., Алгоритм выделения контуров зданий на топографических планах //Материалы Международного научного форума студентов, аспирантов и молодых ученых стран Азиатско-Тихоокеанского региона - 2012, Владивосток, 14-17 мая 2012 г. /под общей редакцией Н.В. Воеводиной. - Владивосток: Издательский дом Дальневост. федерал. ун-та, 2012. - С.- 62..
- [3] Кудряшов А.П. Реконструкция трехмерных сцен городской обстановки. Информационные технологии, №7, 2009. с. 63-68.

## Обавторе

Соловьёв Игорь Владимирович – магистр по направлению прикладная математика и информатика. Окончил Дальневосточный федеральный университет в 2013 году. Его адрес [igorek.solovyev@mail.ru](mailto:igorek.solovyev@mail.ru).

# Отслеживание направления взгляда в реальном времени без использования специальной видеоаппаратуры

Иван Малин

Факультет прикладной математики и физики

Московский Авиационный Институт, Москва, Россия

ivan.malin@gmail.ru

## Аннотация

В данной статье рассматривается задача отслеживания направления взгляда в реальном времени с использованием видеокамер бытового уровня в видимом диапазоне. Приводится анализ и сравнение метода Даугмана и преобразования Хаффа для обнаружения радужки. Предлагается способ определения точки взгляда по положению центров радужек с использованием однослойного перцептрона. Приводятся оценки точности полученной системы.

**Ключевые слова:** eye-tracking, обнаружение радужки, метод Даугмана, метод Хаффа, перцептрон

## 1. ВВЕДЕНИЕ

В настоящее время системы отслеживания направления взгляда используются для решения широкого круга задач. К сферам применения относятся маркетинговые, медицинские и психологические исследования, а также построение бесконтактных интерфейсов для человеко-машинного взаимодействия. В работе (Hansen & Pese, 2005) приводится исчерпывающий обзор современных подходов к отслеживанию направления взгляда. Большинство систем, используемых в настоящее время в исследовательских и коммерческих целях, требуют дополнительного оборудования, например, активной подсветки, ИК- и стереокамер. Таким образом, задача отслеживания направления взгляда в реальном времени с использованием одной видеокамеры, работающей в видимом диапазоне, остается актуальной. Целью данной работы является исследование возможности построения системы, решающей задачу определения области на экране, в которую смотрит пользователь, и отвечающей следующим требованиям:

- Отсутствует необходимость использования специализированной аппаратуры
- Функционирование осуществляется в реальном времени
- Процедура калибровки проста, отсутствует необходимость ввода данных о камере или сцене, таких как фокусное расстояние камеры, взаимного расположение камеры, пользователя и экрана и т. д.

Общая идея всех методов определения точки взгляда по видеопоследовательности глаз заключается в выделении на изображении некоторого вектора признаков и построения отображения пространства признаков на координаты рассматриваемого объекта, чаще всего - плоского экрана. В зависимости от конкретной реализации, используемыми признаками могут быть координаты центров радужек и зрачков, отражений на роговице от активной подсветки и пр. (Ji & Yang, 2001), (Ebisawa & Satoh, 1993), (Morimoto, Koons,

Amir, & Flickner, 2000). Многомерным вектором признаков может являться само изображение глаза, отмасштабированное до некоторого стандартного размера (Xu & Machin, 1998). В данной работе в качестве признаков используются координаты центров радужек глаз, так как они не требуют активной подсветки и могут быть эффективно выделены при обычных условиях съемки. В целях быстродействия радужки ищутся как окружности, несмотря на то, что в общем случае их изображение представляет эллипс. Раздел 2 посвящен сравнению и исследованию применимости двух наиболее популярных методов обнаружения окружностей, в разделе 3 описывается построенная система и результаты измерений её точности.

## 2. МЕТОДЫ ОПРЕДЕЛЕНИЯ ПОЛОЖЕНИЯ РАДУЖКИ

### 2.1 Оператор Даугмана и преобразование Хаффа

Рассмотрим и сравним два метода поиска окружностей на полутоновом изображении. Первый из них – метод Даугмана (Daugman, 2004), ставший стандартом для определения границ радужки в задачах биометрии. Идея метода заключается в поиске наиболее выраженного кругового контура. Для этого ищутся параметры окружности  $(x_0, y_0, r)$  – координаты центра и радиус, в которых достигается максимум оператора Даугмана:

$$\max_{x_0, y_0, r} \left| \frac{1}{2\pi r} G_\sigma(r) * \frac{\partial}{\partial r} \oint_{(x_0, y_0, r)} I(x, y) ds \right| \quad (1)$$

Символ \* означает свертку, а  $G_\sigma(r)$  является сглаживающей функцией, в данном случае гауссианом со среднеквадратическим отклонением  $\sigma$ . Т.к.

$$\frac{\partial}{\partial r} \oint_{(x_0, y_0, r)} I(x, y) ds = \oint_{(x_0, y_0, r)} \left( \frac{\partial I(x, y)}{\partial x} \frac{\partial x}{\partial r} + \frac{\partial I(x, y)}{\partial y} \frac{\partial y}{\partial r} \right) ds$$

(1) можно переписать в ином виде:

$$\max_{x_0, y_0, r} \left| \frac{1}{2\pi r} G_\sigma(r) * \oint_{(x_0, y_0, r)} \langle \nabla I(x, y), c \rangle ds \right| \quad (2)$$

Здесь  $\langle \nabla I(x, y), c \rangle$  обозначает скалярное произведение вектора градиента яркости изображения  $\nabla I(x, y)$  и единичного вектора  $c$ , направленного от центра окружности  $x_0, y_0$  к текущей точке интегрирования (положению элемента  $ds$ ).

Таким образом, для дискретного цифрового изображения процесс поиска окружности методом Даугмана сводится к вычислению градиента изображения и его свертке с маской

радиально ориентированных единичных векторов. Точка, в которой достигается максимальное значение свертки, является центром окружности. Применительно к задаче поиска радужки следует учитывать, что она имеет меньшую яркость, чем белки глаз, поэтому в операторе (2) нужно убрать скобки модуля. Сложность такого алгоритма составляет  $O(M \cdot N \cdot R^2)$ , где  $R = r_{max} - r_{min}$  – величина диапазона возможных радиусов искомой окружности.

Другим общепризнанным методом для обнаружения окружностей является преобразование Хаффа и его различные модификации, в частности, градиентное преобразование. Эта группа методов успешно используется в том числе для обнаружения радужки (Тоенниес, 2002). Ниже описана модификация, использованная в данной работе при проведении сравнения с методом Даугмана.

Для определения положения и радиуса окружности используется трехмерный аккумулятор  $A(x_0, y_0, r)$ . Каждая точка изображения  $I(x, y)$  добавляет вес  $\frac{1}{r} |\nabla I(x, y)|$  к значению аккумулятора  $A$  в точке  $(x - \frac{I_x(x, y)}{|\nabla I(x, y)|} r, y - \frac{I_y(x, y)}{|\nabla I(x, y)|} r, r)$  для всех значений  $r$  в устанавливаемых границах  $[r_{min}, r_{max}]$ . Иными словами, каждая точка изображения голосует с весом  $\frac{1}{r} |\nabla I(x, y)|$  за расположение центра окружности на расстоянии  $r$  от нее в направлении антиградиента. Алгоритмическая сложность процедуры голосования составляет  $O(M \cdot N \cdot R)$ .

Следует отметить, что в большинстве работ, использующих круговое преобразование Хаффа, отсутствует коэффициент  $\frac{1}{r}$  при вычислении веса, с которым голосует точка изображения. Однако в этом случае большим окружностям будут соответствовать большие значения в аккумуляторе, т.к. количество голосующих пикселей на границе окружности линейно зависит от ее радиуса. Введение коэффициента  $\frac{1}{r}$  осуществляет необходимую нормировку (как и в операторе Даугмана).

Также необходимым этапом является постобработка аккумулятора – сглаживание значений путем свертки с трехмерным гауссианом. Это связано с тем, что малая погрешность в оценке направления градиента в точке может дать значительное смещение координат центра, за который эта точка голосует, если радиус достаточно велик. Учитывая сепарабельность гауссовского фильтра, свертку можно осуществить за  $O(M \cdot N \cdot R)$  операций. Таким образом, алгоритмическая сложность данного метода составляет также  $O(M \cdot N \cdot R)$ .

## 2.2 Сравнение методов

Описанная реализация метода Даугмана проигрывает методу Хаффа в скорости работы, однако является более устойчивой к ошибкам определения направлений градиента – в силу интегрального характера оператора (2).

Для тестирования использовались синтезированные изображения окружностей. В каждой точке изображения к вычисленному с помощью оператора Собеля градиенту добавлялся случайно ориентированный вектор случайной длины  $r \sim |N(0; \sigma)|$ . График на рисунке 1 показывает зависимость среднеквадратического отклонения координат центра окружности в зависимости от параметра  $\sigma/G_{max}$ , где  $G_{max}$  – максимальное значение модуля градиента исходного синтезированного изображения. На рисунке 2 показано сравнение времени работы методов в зависимости от размера изображения. Для тестирования использовались

синтезированные изображения с отношением сторон 1:1 с изображениями окружностей, радиус которых составлял от 0.2 до 0.3 стороны изображения.

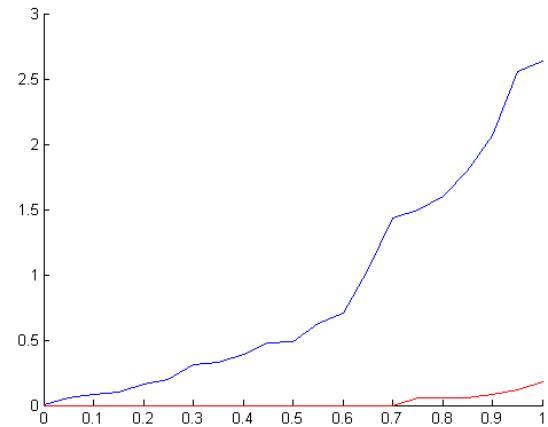


Рисунок 1. Зависимость среднеквадратической ошибки определения положения центра окружности (в пикселях) от  $\sigma/G_{max}$ . Красным обозначен метод Даугмана, синим – Хаффа.

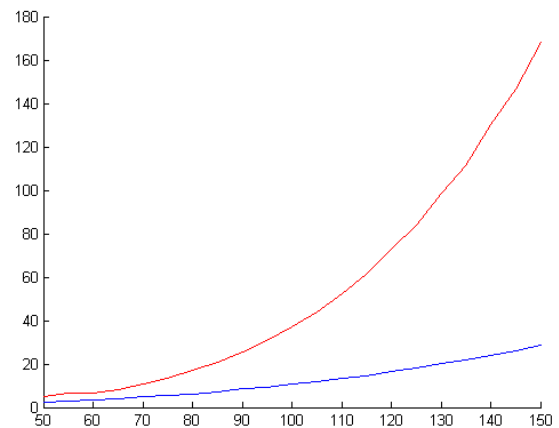


Рисунок 2. Зависимость времени работы методов в миллисекундах от линейного размера изображения в пикселях. Красным обозначен метод Даугмана, синим – Хаффа.

## 3. ПРЕДЛАГАЕМАЯ СИСТЕМА

### 3.1 Общая структура

Как было указано во введении, одной из целей данной работы являлась разработка системы реального времени. В связи с этим за основу алгоритма нахождения центра радужки был взят градиентный метод Хаффа. Следует учитывать, что поиск окружности на всем кадре затратен по времени, а также ведет к большей вероятности ошибок детектирования. Поэтому зона поиска радужек предварительно сужается путем нахождения региона глаз с помощью метода Виолы-Джонса (Viola & Jones, 2001). Также для исследования потенциальной возможности обеспечения пользователю свободы движения головой, осуществлялся поиск налобного маркера круглой формы (см. рисунок 3).

Следующим этапом является построение отображения  $M(x) = y$ , где  $x$  – вектор выделенных на кадре признаков, а  $y \in R_2$  – координаты предполагаемой точки взгляда на экране. В данном случае  $x \in R_6$  и представляет собой набор координат центров радужек обоих глаз и налобного маркера на кадре. В качестве модели отображения  $M$  был выбран однослойный перцептрон. Обучение перцептрона осуществляется методом обратного распространения ошибки (Хайкин, 2006). Обучающей выборкой являются пары  $(x_i, y_i)$ , полученные в результате процесса калибровки, когда пользователю предлагается последовательно посмотреть на несколько меток, расположенных в узлах регулярной сетки на экране.

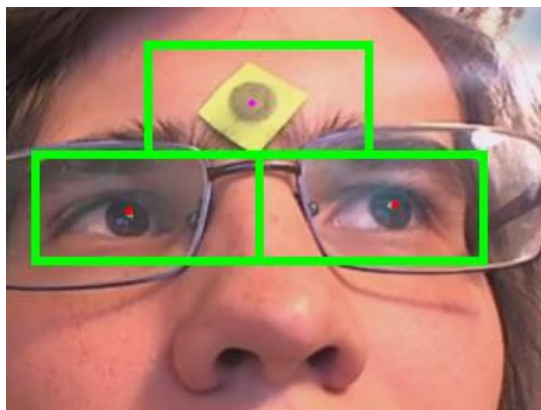


Рисунок 3. Найденные области поиска окрестностей и их полученные центры.

### 3.2 Результаты тестирования

Процедура тестирования была проведена для четырех пользователей, среди которых были лица как с темной, так и со светлой радужкой. Эксперимент проводился в двух вариантах – в условиях естественного дневного света и освещения лампой накаливания. Рассматривалось два случая движений головы – без фиксации (однако с указанием пользователю по возможности держать ее неподвижно) и с фиксацией подбородка на подставке. Расстояние от глаз до экрана составляло около 0,45 м, сам экран имел размеры около 0,41 м по горизонтали и 0,26 м по вертикали. В процессе тестирования пользователь последовательно смотрел на 25 меток на экране, расположенных в узлах регулярной прямоугольной сетки размером 5 на 5, целиком покрывающей экран. В дальнейшем 13 точек с четными номерами были использованы в качестве обучающей выборки для построения функции отображения координат, а оставшиеся 12 – в качестве тестирующей выборки для оценки точности системы. Съемка осуществлялась веб-камерой Logitech HD Pro Webcam C920.

Среднеквадратическая ошибка определения координат точки взгляда приведена в таблице 1. Визуализация вычисленных точек взгляда представлена на рисунках 4 и 5. Таким образом, среднеквадратическая ошибка на тестирующей выборке в наилучшем случае составила около 30 пикселей по вертикали и 50 по горизонтали на мониторе с разрешением 1440 на 900.

Таблица 1. Среднеквадратическая ошибка определения координат точки взгляда (в долях от линейных размеров экрана).

	Обучающая выборка	Тестирующая выборка
Без фиксации, без налобного маркера	0.13	0.20
Без фиксации головы, с налобным маркером	0.021	0.036
С фиксацией головы	0.017	0.033

	$\sigma_x$	$\sigma_y$	$\sigma_x$	$\sigma_y$
Без фиксации, без налобного маркера	0.13	0.07	0.20	0.24
Без фиксации головы, с налобным маркером	0.021	0.027	0.036	0.039
С фиксацией головы	0.017	0.025	0.033	0.035

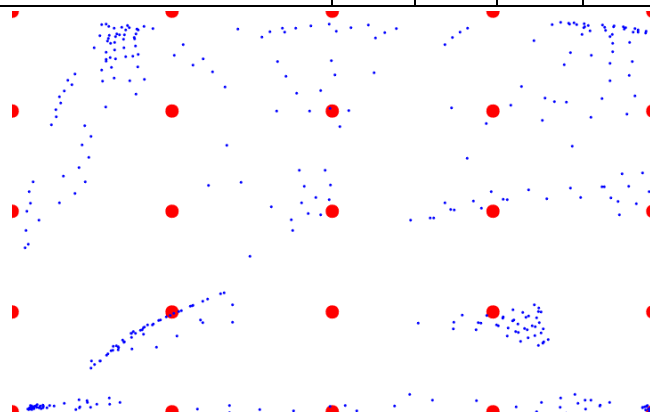


Рисунок 4. Пример некорректного отображения координат без использования налобного маркера и фиксации головы.

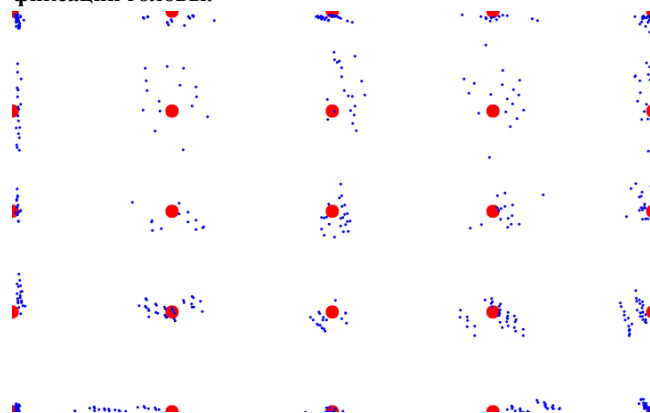


Рисунок 5. Пример отображения координат при использовании налобного маркера без фиксации головы.

## 4. ЗАКЛЮЧЕНИЕ

Предлагаемый метод позволяет создать систему отслеживания направления взгляда, работающую в режиме реального времени и использующую только широкодоступную видеоаппаратуру бытового уровня. Использование одного налобного маркера как источника дополнительной информации о положении головы позволяет обеспечить некоторую малую свободу движений пользователя. Однако для обеспечения свободы амплитудных движений требуется дополнительная информация. Вопрос об оценке требуемого количества маркеров и калибровочных точек для обеспечения свободы движений головы, а также о структуре и размерах слоев перцептрона, также остается открытым.



## 5. ССЫЛКИ

- Daugman, J. (January 2004 г.). How Iris Recognition Works. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, 14(1).
- Ebisawa, Y., & Satoh, S. (1993). Effectiveness of pupil area detection technique using two light sources and image difference method. *5th Annual Int. Conf. of the IEEE Eng. in Medicine and Biology Society*, (стр. 1268–1269).
- Ji, Q., & Yang, X. (2001). Real time visual cues extraction for monitoring driver vigilance. *Comput Sci 2095*, 107.
- Morimoto, C., Koons, D., Amir, A., & Flickner, M. (2000). Pupil detection and tracking using multiple light sources. *IVC 18 (4)*, (стр. 331–335).
- Toennies, K. (2002). Feasibility of Hough-transform-based iris localisation for real-time-application. *Pattern Recognition, 2002. Proceedings on 16th International Conference, 2*, стр. 1053 - 1056.
- Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *ACCEPTED CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION 2001*.
- Xu, L., & Machin, D. (1998). A novel approach to real-time non-intrusive gaze finding. *British Machine Vision Conference*.
- Хайкин, С. (2006). *Нейронные сети*. Москва: Вильямс.

# Создание панорамных карт подводного дна на основе больших массивов изображений

Камаев Александр  
Вычислительный центр ДВО РАН, Хабаровск, Россия  
kamaev\_an@mail.ru

## Аннотация

Подводное дно является интересным для изучения, но труднодоступным объектом. Одним из способов получения сведений о поверхности дна, о флоре и фауне, а также о состоянии инженерных коммуникаций, проложенных на дне, является составление его панорамных карт. Массивы изображений для составления подобных карт могут быть получены с автономных необитаемых подводных аппаратов АНПА. При этом возникает ряд трудностей, связанных с плохими условиями освещённости, характерными особенностями дна, отсутствием точной gps навигации под водой. В статье рассматриваются все факторы, затрудняющие составление панорамных карт дна и предлагаются пути решения возникающих проблем.

**Ключевые слова:** Панорама, АНПА, Камера, Особые Точки, Метод Наименьших Квадратов, Bundle Adjustment.

## 1. ВВЕДЕНИЕ

Рассмотрим АНПА, оснащённый камерой с известной матрицей калибровки  $\mathbf{K}=\text{diag}(f_x, f_y, 1)$  [5], жёстко связанной с аппаратом и направленной вниз. Элементы  $f_x$  и  $f_y$  – фокусные расстояния в пикселях (если пиксели имеют одинаковую высоту и ширину, то  $f_x=f_y$ ). АНПА совершает проходы над дном, непрерывно его фотографируя. На момент окончания миссии получается  $m$  фотографий. При этом координаты аппарата  $\mathbf{p}_j=(t_{jx}, t_{jy}, t_{jz})$ ,  $j=1, 2, \dots, m$  в момент получения каждой фотографии неизвестны. Известными является курс аппарата  $\gamma_j$ , его глубина  $d_j$  и высота над дном  $h_j$ . Как правило, эти величины доступны из навигационной системы любого АНПА с известной точностью. Координаты  $\mathbf{p}_j$  также могут быть представлены системой навигации АНПА, однако их точность обычно неудовлетворительна для использования в задаче построения панорамы.

В процессе фотографирования дна крен  $\alpha_j$  и дифферент  $\beta_j$  колеблются около нулевых значений, часто эти колебания слишком малы для точной регистрации, но могут оказаться существенными при построении панорамы, поэтому данные параметры будем также считать неизвестными.

Задача заключается в определении точных значений векторов  $\mathbf{p}_j=(p_{jx}, p_{jy}, p_{jz})$  и  $\mathbf{r}_j=(\alpha_j, \beta_j, \gamma_j)$  на основе визуальных особенностей, присутствующих на фотографиях и известных данных  $\gamma_j$ ,  $d_j$  и  $h_j$ . Вектора  $\mathbf{p}_j$  и  $\mathbf{r}_j$  могут быть использованы для точного позиционирования каждой фотографии в панорамной карте дна.

## 2. СПЕЦИФИКА ЗАДАЧИ

У рассматриваемой задачи есть ряд специфических особенностей, которые затрудняют её решение. Эти особенности, так или иначе, вытекают из плохих условий съёмки (вода может быть мутной, свет может не проникать на большую глубину), из специфики подводных объектов и из

отсутствия возможности точного определения координат  $\mathbf{p}_j$  с помощью GPS, как это возможно делать при аэрофотосъёмке.

Из-за отсутствия света под водой приходится использовать искусственное освещение, что приводит к неравномерной освещённости каждого снимка. Это в свою очередь затрудняет процесс сопоставления. Искусственное освещение и мутная вода способствуют получению слабоконтрастных, нечётких снимков. Также задачу осложняет тот факт, что на дне часто встречается большое количество совершенно одинаковых объектов: ракушки, морские ежи, водоросли и т.д. Всё это приводит к низкой различимости разных визуальных особенностей на снимках.

Отсутствие естественного света делает невозможным съёмку с большой высоты  $h_j$ , высота съёмки ограничивается мощностью прожекторов АНПА. Следовательно, зона покрытия каждого снимка оказывается небольшой и для покрытия больших территорий требуются тысячи и даже десятки тысяч снимков. При этом точное положение АНПА  $\mathbf{p}_j$  в момент получения каждого снимка остаётся неизвестным. В результате получается задача с огромным количеством изображений и очень слабо выраженными и плохо различимыми визуальными особенностями.

Кроме негативных факторов, усложняющих задачу, присутствует ряд особенностей, делающих решение поставленной задачи возможным. К таким факторам относятся известные величины  $\gamma_j$ ,  $d_j$  и  $h_j$ , а также близкая к плоской траектория движения АНПА. Используя данные особенности, в следующих разделах поэтапно рассмотрим процесс решения задачи построения панорамы.

## 3. СОПОСТАВЛЕНИЕ ПАР ИЗОБРАЖЕНИЙ

В основе задачи построения панорамы лежит задача определения связи двух изображений. Рассмотрим эту задачу для пары изображений  $i, j$ ,  $j=1, 2, \dots, m$ . Для того, чтобы определить как связаны  $i$ -е и  $j$ -е изображения необходимо на этих изображениях выделить визуальные особенности. Устойчивыми особенностями являются особые точки SIFT [4]. Дескрипторы SIFT [4] устойчивы к плоскому повороту, масштабу, перемещению изображения и к линейному изменению освещённости. Однако эти дескрипторы довольно чувствительны к нелинейным изменениям освещённости, которые имеют место быть при подводной съёмке – рис. 1а.

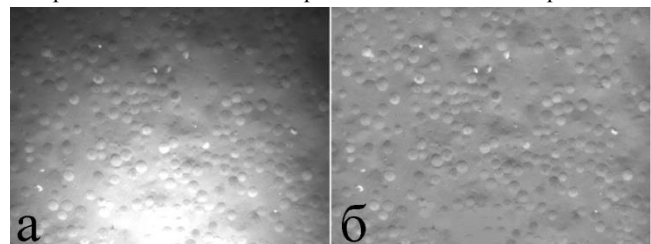


Рис 1: Пример компенсации нелинейной освещённости.

По этой причине до составления дескрипторов, необходимо выровнять освещение на всех снимках, воспользовавшись алгоритмом Single Scale Retinex [6]. Пример применения алгоритма к изображению на рис. 1а показан на рисунке 1б.

Далее в результате сравнения евклидовых расстояний между всеми возможными парами дескрипторов особых точек  $i$ -го и  $j$ -го изображений получим список пар точек, расстояние между которыми меньше определённого порога. В качестве порога в случае нормализованных дескрипторов можно выбрать расстояние, до 0.5. Среди сопоставленных точек будет очень высокий процент ложных соответствий. Из-за низкой контрастности и чёткости снимков, а также повторяющихся однотипных объектов на дне, различимость дескрипторов становится низкой, и количество ложных соответствий может достигать 90% (рис. 2а).

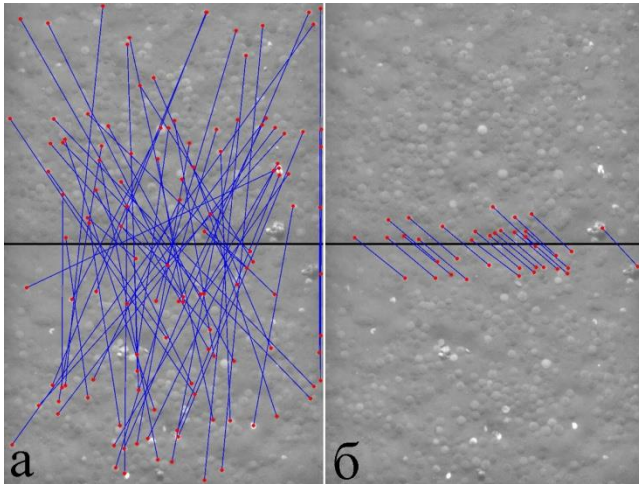


Рис 2: Фильтрация ложных соответствий.

Для фильтрации ложных соответствий обычно используется процедура RANSAC [10], однако, она становится неэффективной, когда количество ложных соответствий превышает 60%. Чтобы отфильтровать ложные соответствия в рассматриваемом случае введём такую модель связи, которая содержит минимум параметров.

Так как перемещение АНПА над дном в первом приближении может считаться плоским, для приблизительной оценки связи изображений ограничимся следующей моделью:

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \frac{h_j}{h_i} \cdot \begin{pmatrix} \cos(\gamma_j - \gamma_i) & \sin(\gamma_j - \gamma_i) \\ -\sin(\gamma_j - \gamma_i) & \cos(\gamma_j - \gamma_i) \end{pmatrix} \cdot \begin{pmatrix} x_j \\ y_j \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}.$$

Здесь  $(x_i, y_i)^T$  и  $(x_j, y_j)^T$  – координаты соответствующих точек на  $i$ -ом и  $j$ -ом изображениях, а  $\mathbf{t}_{ij} = (t_x, t_y)$  – вектор смещения центра  $j$ -го изображения относительно центра  $i$ -го изображения. Неизвестный вектор  $\mathbf{t}_{ij}$  имеет масштаб  $i$ -го изображения и может быть найден на основе одного точечного соответствия. Если перебрать все точечные соответствия и для каждого построить модель связи, а затем из всех моделей выбрать согласующуюся с наибольшим количеством точек, то выбранная модель и будет истинной, в случае если изображения  $i$  и  $j$  действительно связаны. Определить, существует ли связь изображений, можно воспользовавшись следующим условием:

$$n_i > \alpha + \beta \cdot n_f$$

Здесь  $n_i$  – количество соответствий, удовлетворяющих выбранной модели, а  $n_f$  – количество соответствий, находящихся в зоне пересечения изображений, но не

удовлетворяющих модели связи. Коэффициент  $\alpha=2$ , а  $\beta=0.3$  [11]. Невыполнение условия говорит об отсутствии связи.

Если связь изображений подтвердилась, то процедура поиска соответствующих особенностей повторяется заново, при этом учитывается найденная модель связи. В результате повторного поиска, как правило, находятся дополнительные соответствия, найти которые без учёта модели связи было невозможно, а ложные соответствия отфильтровываются – рис. 2б.

#### 4. ПОИСК СВЯЗАННЫХ ПАР ИЗОБРАЖЕНИЙ

Поиск связанных пар изображений – самый затратный по времени этап. Очевидно, что поиск путём полного перебора всех возможных пар и применения к ним алгоритма из раздела 3, подходит лишь для небольшого количества изображений. В условиях, когда большое количество изображений является одной из характеристик решаемой задачи, такой поиск «грубой силой» не применим.

Ключом к организации быстрого поиска связанных пар изображений является процедура поиска похожих дескрипторов в пространстве с большой размерностью. Предположим, что имеется некоторый дескриптор особенности  $\mathbf{q}$ , необходимо из множества дескрипторов всех особенностей выбирать дескрипторы, наиболее близкие к дескриптору  $\mathbf{q}$ . Это задача называется поиском  $k$  ближайших соседей и различные методы её решения, как точные, так и приближенные хорошо изучены. Наиболее известные методы для решения такой задачи в пространствах с высокой размерностью – это «best bin first» [2], локально-чувствительное хеширование (LSH) [7], метрические деревья [3],  $sp$ -деревья и гибридные  $sp$ -деревья [8]. Гибридные  $sp$ -деревья, показавшие наилучшую производительность, были использованы в данной работе, совместно с приоритетной очередью из алгоритма «best bin first».

Из-за того, что дескрипторы особенностей в рассматриваемой задаче имеют низкую различимость (слабо отличаются друг от друга), группы похожих дескрипторов, найденных с помощью гибридных  $sp$ -деревьев нельзя напрямую использовать для вычисления связей между изображениями. Но информация о похожих дескрипторах может быть использована для определения групп изображений, которые могут пересекаться. А связи в этих группах могут быть рассчитаны с помощью метода из раздела 3.

Обозначим дескриптор  $j$ -ой особенности  $i$ -го изображения за  $\mathbf{v}_{ij}$ ,  $i=1, 2, \dots, m$ ,  $j=1, 2, \dots, \lambda(i)$ , где  $\lambda(i)$  – количество особенностей на  $i$ -ом изображении. С помощью гибридного  $sp$ -дерева найдём вектор  $\mathbf{l}_{ij}$ , состоящий из номеров изображений, содержащих первые  $k$  особенностей с дескрипторами, максимально близкими к дескриптору  $\mathbf{v}_{ij}$ :  $\mathbf{l}_{ij} = (l_1^{ij}, l_2^{ij}, \dots, l_k^{ij})$ . Затем составим списки предполагаемых кандидатов на сопоставление для каждого изображения. Для этого введём матрицу  $\mathbf{W} = (w_{pb})$ ,  $p, b=1, 2, \dots, m$ , где  $w_{pb}$  – количество похожих дескрипторов между парой изображений  $p$  и  $b$ . Изначально, все  $w_{pb}$  приравниваются к нулю. Далее перебираются все элементы  $l_n^{ij}$ ,  $i=1, 2, \dots, m$ ,  $j=1, 2, \dots, \lambda(i)$ ,  $n=1, 2, \dots, k$  и компоненты матрицы  $\mathbf{W}$ :  $w_{i, l_n^{ij}}$  и  $w_{l_n^{ij}, i}$  увеличиваются на единицу. В список предполагаемых кандидатов на сопоставление каждого изображения  $\mathbf{S}_p = (s_{p1}, s_{p2}, \dots, s_{p\eta})$ ,  $p=1, 2, \dots, m$  помещаются  $\eta$  номеров столбцов наибольших элементов из  $p$ -ой строки матрицы  $\mathbf{W}$ .

После вычисления списков  $S_p$ , процедуру сопоставления 3 нужно выполнять не  $m$  раз для каждого  $p$ -го изображения, а всего лишь  $\eta$  раз. Скорость и точность алгоритма можно варьировать, изменяя параметр  $\eta$ . Чем  $\eta$  выше, тем больше связанных пар может быть найдено, но тем медленнее будет происходить поиск. Также на скорость и точность алгоритма влияет параметр  $E$  – максимальное количество узлов гибридного sp-дерева, которое просматривается при поиске похожих дескрипторов. Влияния параметров  $E$  и  $\eta$  на точность и скорость поиска связанных пар изображений отображают графики на рис. 3а и 3б.

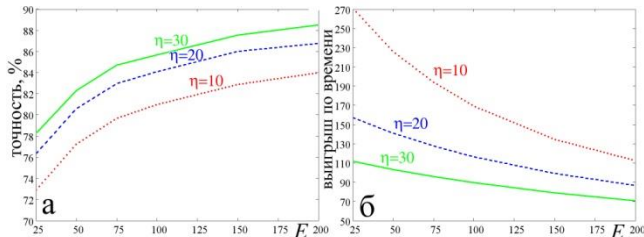


Рис 3: Влияние параметров  $\eta$  и  $E$  на процедуру поиска.

В качестве скорости работы алгоритма взят коэффициент выигрыша по времени, который даёт алгоритм в сравнении с поиском «грубой силой», а в качестве точности взят процент связанных пар изображений, которые удалось обнаружить от количества пар, обнаруженных поиском «грубой силой». Графики на рис 3 получены для задачи с  $m=3591$  изображениями и с 2362523 дескрипторами.

Ещё одним параметром, влияющим на точность алгоритма, является  $k$  – количество похожих дескрипторов, которые нужно находить для каждого дескриптора – запроса. Функция зависимости точности сопоставления от  $k$  достигает своего максимума при  $k=2$  независимо от значений других параметров. Поэтому целесообразно использовать для  $k$  именно это значение.

### 5. ПРИБЛИЗИТЕЛЬНОЕ ПОЗИЦИОНИРОВАНИЕ

Задача поиска точных значений векторов  $\mathbf{p}_j$  и  $\mathbf{r}_j$ ,  $j=1, 2, \dots, m$  по набору известных соответствующих проекций точек пространства является нелинейной, так как связь точки пространства и её проекции является нелинейной. Это означает, что необходимо использовать итерационные методы решения задачи, для которых требуется хорошее начальное приближение. Воспользовавшись тем, что движение АНПА преимущественно плоское, а колебания крена и дифферента аппарата не значительны, для поиска начального приближения будем решать двухмерную задачу, приняв крен и дифферент нулевыми.

После обнаружения всех связанных пар (раздел 4) и вычисления параметров связи  $\mathbf{t}_{ij}$  (раздел 3) задача приблизительного позиционирования может быть решена с помощью метода наименьших квадратов (МНК).

Будем считать, что было найдено  $n$  связанных пар изображений, а пары изображений с номерами  $i(l)$   $j(l)$ , где  $l=1, 2, \dots, n$  – связаны, т.е. известны векторы  $\mathbf{t}_{i(l)j(l)}$ . Предполагается, что все  $m$  изображений оказались связаны (граф, построенный из вершин – изображений и рёбер – известных связей  $i(l)$   $j(l)$  связан). Если это не так, то следует из множества изображений выделить подмножества связанных изображений и решать задачу приблизительного позиционирования для каждого подмножества отдельно.

Найдём положения центра каждого изображения  $\mathbf{c}_k=(c_{kx}, c_{ky})$ ,  $k=1, 2, \dots, m$  в системе координат некоторого изображения  $q \in \{1, 2, \dots, m\}$ . Переведём все векторы  $\mathbf{t}_{i(l)j(l)}$  в систему координат  $q$ -го изображения, домножив их на масштаб между  $q$ -ым и  $i(l)$ -ым изображением и повернув на угол  $\gamma_{i(l)} - \gamma_q$ :

$$\mathbf{d}_l = \frac{h_{i(l)}}{h_q} \cdot \begin{pmatrix} \cos(\gamma_{i(l)} - \gamma_q) & \sin(\gamma_{i(l)} - \gamma_q) \\ -\sin(\gamma_{i(l)} - \gamma_q) & \cos(\gamma_{i(l)} - \gamma_q) \end{pmatrix} \cdot \mathbf{t}_l$$

Здесь  $\mathbf{d}_l$  – вектор смещения центра изображения  $j(l)$  относительно центра изображения  $i(l)$ , заданный в системе координат  $q$ -го изображения. Каждый вектор  $\mathbf{d}_l$  даёт два уравнения относительно неизвестных координат центров изображений  $\mathbf{c}_k$ :  $\mathbf{d}_l = \mathbf{c}_{j(l)} - \mathbf{c}_{i(l)}$ . Совокупность уравнений для  $l=1, 2, \dots, n$  составляет систему линейных алгебраических уравнений. Приняв вектор  $\mathbf{c}_q=(0, 0)$ , разрешив данную систему, воспользовавшись линейным МНК [5].

Отметим, что матрица нормального уравнения МНК  $\mathbf{Q}=\mathbf{A}^T\mathbf{A}$ , где  $\mathbf{A}$  – матрица исходной системы, будет разреженной и к ней применимы методы эффективного решения из [12].

Зная координаты центров изображений  $\mathbf{c}_k$ , заданные в системе координат  $q$ -го изображения, легко получить планарные координаты АНПА  $p_{kx}, p_{ky}$ . Для этого достаточно выполнить следующие умножения:

$$p_{kx} = c_{kx} \cdot \frac{h_k}{f_x}, \quad p_{ky} = c_{ky} \cdot \frac{h_k}{f_y}$$

Поскольку никаких данных о привязке к глобальной системе координат нет, координаты АНПА определяются с точностью до некоторого вектора перемещения.

### 6. ТОЧНОЕ ПОЗИЦИОНИРОВАНИЕ

Заключительным этапом вычисления векторов  $\mathbf{p}_j$  и  $\mathbf{r}_j$  является процедура их уточнения. На начало процедуры уточнения приблизительные значения данных векторов известны. Обозначим эти значения через  $\mathbf{p}_j^0$  и  $\mathbf{r}_j^0$ . Значения  $p_{jx}^0$  и  $p_{jy}^0$ , составляющие вектор  $\mathbf{p}_j^0$  были найдены в разделе 5, курс АНПА  $\gamma_j^0$  известен из навигационной системы, а начальные крен и дифферент примем нулевыми:  $\alpha_j^0 = 0, \beta_j^0 = 0$ . Значение  $p_{jz}^0$ , как и начальный курс может быть извлечено из навигационной системы:  $p_{jz}^0 = d_j$ .

Начальные значения  $\mathbf{p}_j^0$  и  $\mathbf{r}_j^0$  могут содержать ошибки, недопустимые при построении панорам, и требуют уточнения. Стандартным способом уточнения является использование метода bundle adjustment [5], работающего, обычно, совместно с алгоритмом Левенберга–Марквардта [9].

Для использования bundle adjustment, дополнительно к векторам параметров камеры  $\mathbf{a}_j=(\mathbf{p}_j, \mathbf{r}_j)$  введём вектор точек пространства  $\mathbf{b}_i=(x_i, y_i, z_i)$ ,  $j=1, 2, \dots, m, i=1, 2, \dots, n$ , где через  $n$  обозначим количество точек дна, видимых одновременно на двух и более изображениях. Евклидовы координаты проекции точек  $\mathbf{b}_i$  на  $j$ -ое изображение обозначим через  $\mathbf{x}_i^j = (u_i^j, v_i^j)$ , а однородные координаты проекций через  $\tilde{\mathbf{x}}_i^j = (\tilde{u}_i^j, \tilde{v}_i^j, \tilde{w}_i^j)$ , где  $\tilde{w}_i^j = 1$ . Перейти от однородных координат к евклидовым можно посредством деления первых двух компонент на третью:  $u_i^j = \tilde{u}_i^j / \tilde{w}_i^j, v_i^j = \tilde{v}_i^j / \tilde{w}_i^j$ .

Каждая точка  $\mathbf{b}_i$  связана с координатами своей проекции на  $j$ -ое изображение посредством параметров  $\mathbf{a}_j$ :

$$\tilde{\mathbf{x}}_i^j = \mathbf{K} \cdot (\mathbf{R}^T(\mathbf{r}_j) - \mathbf{R}^T(\mathbf{r}_j) \cdot \mathbf{p}_j) \cdot (\mathbf{b}_i \ 1)^T$$

Здесь  $R(\mathbf{r}_j)$  – матрица поворота АНПА, являющаяся комбинацией трёх матриц поворота вокруг координатных осей:  $R(\mathbf{r}_j) = R_z(\gamma_j) \cdot R_y(\beta_j) \cdot R_x(\alpha_j)$ . Таким образом, можно говорить о некоторой функции проекции  $f_i^j$ :  $\mathbf{x}_i^j = f_i^j(\mathbf{a}_j, \mathbf{b}_i)$ , которая отображает параметры, подлежащие уточнению на координаты проекции  $i$ -ой точки на  $j$ -ом изображении.

С другой стороны, известны проекции каждой  $i$ -ой точки пространства (раздел 3), обозначим их через  $\hat{\mathbf{x}}_i^j = (\hat{x}_i^j, \hat{y}_i^j)$ .

Метод bundle adjustment заключается в итеративном уточнении параметров  $\mathbf{a}_j$ ,  $\mathbf{b}_i$ , приводящем к уменьшению ошибки репроекции  $E$ :

$$E = \sum_{i=1}^n \sum_{j=1}^m |\hat{\mathbf{x}}_i^j - \mathbf{x}_i^j|^2$$

Сам метод и его эффективная реализация рассмотрена в [5]. Использование методик решения разреженных систем линейных уравнений [12] в совокупности с принципом разделения параметров на координаты точек  $\mathbf{b}_i$  и параметры камер  $\mathbf{a}_j$  описанного в [5] позволяет решать задачу уточнения для десятков тысяч изображений и миллионов точек.

В качестве начальных значений для параметров камеры следует взять  $\mathbf{p}_j^0$  и  $\mathbf{r}_j^0$ , а начальные значения для координат точек  $\mathbf{b}_i^0$ , рассчитываются с использованием DLT метода [5].

## 7. СШИВКА ИЗОБРАЖЕНИЙ

Вопрос сшивки изображений подводного дна в единую панораму не имеет какой-либо специфики и может быть выполнен любым доступным способом.

Самый простой способ заключается в аппроксимации участка дна, представляемого каждым изображением плоскостью (выбирается плоскость с наименьшей суммой расстояний от этой плоскости до всех точек, видимых на изображении) и проекции этой плоскости на экран, параллельный горизонтальной плоскости дна. Пример карты куска дна, полученного таким образом, представлен на рисунке 4.

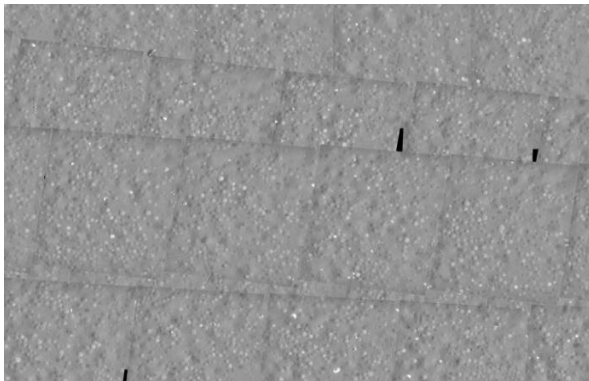


Рис 4: Участок панорамы подводного дна.

Более сложный метод заключается в триангуляции всех точек подводного дна и проецировании каждого треугольника на плоскость карты.

Для устранения швов на границах изображений, если это необходимо, можно использовать любой стандартный метода, например multi-band blending [1].

## 8. ЗАКЛЮЧЕНИЕ

Метод, представленный в данной статье, позволяет собирать панорамные карты подводного дна из десятков тысяч

изображений, сделанных в условиях плохой освещённости. При этом используются только визуальные особенности на самих изображениях и та информация навигационной системы, которая может быть получена с достаточной точностью большинством АНПА. Для работы метода не требуется знать координаты АНПА, получение которых обычно вызывает затруднения ввиду отсутствия GPS сигнала под водой.

Недостатком описанной техники является её неработоспособность в случае получения изображений без каких-либо деталей. Даже на гладком песчаном дне есть детали, однако, если большое число фотографий было выполнено с большого удаления, когда освещение не достаёт до дна, то в процессе вычисления положения снимков могут возникнуть проблемы. Задача может быть разбита на несколько несвязанных подзадач или вовсе не решена.

## 9. ССЫЛКИ

- [1] Allene C., Pons J.-P., Keriven R. Seamless image-based texture atlases using multi-band blending. *In Proc. of the ICPR 2008*, pp. 1-4.
- [2] Beis J., Lowe D.G. Shape indexing using approximate nearest-neighbor search in high-dimensional spaces. *In Proc. of CVPR. – Puerto Rico, 1997. – pp. 1000-1006.*
- [3] Ciaccia P., Patella M., Zezula P. M-tree: An efficient access method for similarity search in metric spaces. *In Proc. of the 23-rd VLDB International Conference, 1997. – pp. 426-435.*
- [4] David G. Lowe, Distinctive image features from scale-invariant keypoint. *International Journal of Computer Vision*, 60, 2 (2004), pp. 91-110.
- [5] Hartley R., Zisserman A. Multiple View Geometry in Computer Vision. – Cambridge University Press, 2004.
- [6] Hines G. D., Rahman Z., Jobson D. J., Woodell G. A. Single-Scale Retinex Using Digital Signal Processors. *Global Signal Processing Expo (GSPx)*, (2004).
- [7] Indyk. P., Motwani R. Approximate nearest neighbours: towards removing the curse of dimensionality. *In Proc. of the 30-th ACM Symposium on the STOC, 1998. – pp. 604-613.*
- [8] Liu T, Moore A., Gray A., Yang K. An investigation of practical approximate nearest neighbor algorithms. *In Proc. of the conference on NIPS, 2004. – P. 825-832.*
- [9] Marquard. D. An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.*, 1963, Vol. 11, pp. 431-441.
- [10] Martin A. Fischler, Robert C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Computations Of the ACM* 24: 381–395, 1981.
- [11] Matthew Brown, David G. Lowe, Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74, 1 (2007), pp. 59-73.
- [12] Джордж А., Лю Дж. Численное решение больших разреженных систем уравнений: Пер. с англ. – М.: Мир, 1984.

## Об авторе

Камаев Александр – аспирант ВЦ ДВО РАН.

Адрес email: [kamaev\\_an@mail.ru](mailto:kamaev_an@mail.ru).

# Плагин метода световых сеток для 3ds Max

Наталья Бологова

Факультет информационных технологий Новосибирского государственного университета,

Новосибирск, Россия

tasha.light2492@gmail.com

## Аннотация

Метод световых сеток (МСС) является модификацией алгоритма обратной рекурсивной трассировки лучей (ОРЛТ) Виттеда и позволяет генерировать мягкие тени для сцен с точечными источниками. Для очень сложных сцен с большим числом источников расчет изображений по МСС оказывается даже быстрее, чем расчет тех же сцен по ОРЛТ. При реализации МСС строится дополнительный объект *световая сетка*, видимость точек которой из источников света необходимо рассчитывать.

В работе рассмотрены две задачи. Первая заключается в разработке плагина для 3ds max и панели GUI для него, который позволял бы применять МСС для рендеринга сцен наряду с ОРЛТ. Также МСС может служить альтернативой методам AreaShadow и ShadowMap для генерации мягких теней.

Вторая задача – это разработка и реализация гибридного алгоритма МСС+АТК, который для расчета видимости точек световой сетки применяет алгоритм теневых карт (АТК). Описываются численные эксперименты, анализ достигнутых ускорений по сравнению с чистым МСС и сравнивается качество изображений, полученных по МСС и по МСС+АТК.

**Ключевые слова:** фотореалистический рендеринг, алгоритм Виттеда, метод световых сеток, теневые карты, мягкие тени, плагин 3ds max.

## 1. ВВЕДЕНИЕ

На современном этапе производство видеороликов на основе применения программных графических средств стало обычным делом: японские аниме, реклама, учебные фильмы и т.д. вплоть до киноиндустрии. И на каждом уровне применяются свои критерии соотношения цены (т.е. времени расчета) и качества. Возьмем, к примеру, создание учебного ролика. Разработчик ролика применяет 3ds Max для создания геометрии виртуальной динамической сцены, затем рассчитывает фильм. Правильное изображение теней имеет огромное значение для восприятия сцены. В реальности тени не четкие – мягкие, когда имеется зона полутени. На практике для рендеринга сцен очень часто применяется алгоритм Виттеда [5]. В модели Виттеда теоретически возможны только четкие тени. Что видим в 3ds Max для имитации мягких теней? Area Shadow (очень медленный), Shadow map (быстрый, но с артефактами), другие. Актуально добавлять новые методы для обогащения палитры инструментов.

Стоит обратить внимание на метод световых сеток (МСС), подробнее см. [1, 2, 4], из-за следующих его характеристик:

1. МСС является модификацией обратной рекурсивной трассировки лучей Виттеда ОРЛТ.
2. Простые формулы для расчета освещенности в точке. Видимость объектных точек из источников рассчитывать не надо, только видимость точек световой сетки.
3. Первое и второе позволяют просто включить МСС в любую реализацию ОРЛТ.
4. При увеличении сложности задачи рендеринга (большое число примитивов, много источников, большой размер изображения) время расчета изображения по МСС становится меньше времени расчета по ОРЛТ. Это качество позволит использовать МСС вместо ОРЛТ в ряде приложений.
5. Неотъемлемой чертой МСС является генерация мягких теней, МСС не позволяет получать четкие тени.

В работе представлена разработка плагина для 3ds Max, реализующего метод световых сеток, с диалогом пользовательского интерфейса в стиле 3ds Max для управления режимами рендеринга теней. При этом мы будем иметь в виду, что МСС дает мягкие тени не бесплатно, но дополнительные затраты на расчет мягких теней попробуем уменьшить.

## 2. ОПИСАНИЕ МЕТОДА

### 2.1 Обратная рекурсивная лучевая трассировка.

Пространственная сцена – это кусочно-непрерывная поверхность, представленная набором 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  $LP_i$ ,  $i = 1, \dots, nL$ . 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 объектной точке  $P$ . Базовую формулу расчета значения интенсивности в точке  $P$  по ОРЛТ можно кратко записать

$$I_{RT}(P) = A + \sum_{i=1}^{nL} V(LP_i, P) \Omega(L_i, P) + k_s I_r + k_t I_t \quad (1)$$

$$= A + U_{RT}(P) + k_s I_r + k_t I_t,$$

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  $P$ .

## 2.2 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  $DB_i[N, M]$ , рассчитанный для камеры, находящейся в позиции источника освещения  $LP_i$ . 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 [6, 2].

## 2.3 Алгоритм теневых объемов

АТО для определения видимости изображаемой точки использует дополнительные геометрические построения, называемые теневыми объемами. Его также можно было бы применить наряду с АТК, см. [3, 2], но в данной работе он не рассматривается.

## 2.4 Метод световых сеток

Следуя [2, 4], 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

$$U_{LM}(P) = \sum_{i=1}^{nL} \left[ \left( \frac{1}{m} \sum_{k=1}^m V(LP_i, x_k) \chi(P, L_i) \right) \Omega(L_i, P) \right], \quad (3)$$

$$I_{LM}(P) = A + U_{LM}(P) + k_s I_r + k_t I_t$$

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. по ОРЛТ.

В формуле (3) используется функция значения видимости источника световой точкой  $V(LP_i, x_k)$ , которая требует значительных вычислительных затрат. Для уменьшения этих затрат рассматривается алгоритм, являющийся гибридом МСС и АТК. Основная идея: заимствовать для МСС положительное качество АТК – быстрый тест видимости источника – здесь: видимости точек сетки.

## 2.5 Комплексование МСС с АТК

Рассмотрим этот алгоритм МСС (см. [4]). Для определения значения интенсивности в объектной точке по МСС применяется формула (3), в которой используется функция значения видимости источника световой точкой  $V(LP_i, x_k)$ .

1. Для определения видимости выполняется длинный тест, т.е. отыскивается пересечение отрезка  $[x_k, LP_i]$  с объектами сцены – операция, которая требует значительных вычислительных затрат.

Вспомним (2) как отбираются световые точки в интерполяционное множество  $D_{LM}(P, r)$ : строится полусфера радиуса  $r$  с центром в точке  $P$ , и для каждой световой точки  $x_j$ , попавшей в полусферу, проверяется видимость из объектной точки  $V(P, x_j)$ .

2. Для определения  $V(P, x_j)$  выполняется короткий тест, т.е. отыскивается пересечение отрезка  $[x_j, P]$  с точками объектов сцены – второй основной источник вычислительных затрат МСС. Короткий тест обеспечивает защиту от протечек света и тени.

Мы выделили два источника вычислительных затрат. Первый из них мы ускоряем за счет применения АТК, т.е. видимость световых точек осуществляется за счет использования теневой карты источника.

Чтобы «ослабить» второй источник затрат времени счета мы проведем исследование: нельзя ли отказаться от короткого теста вообще, хотя бы для получения черновых изображений.

## 3. РАЗРАБОТКА ПЛАГИНА AUTODESK 3DS MAX

Для внедрения МСС в практику, необходимо встроить его в какой-либо популярный редактор трехмерных сцен. Autodesk 3ds Max был выбран для реализации МСС в виде подключаемого модуля (плагины).

### 3.1 Рендеринг в 3ds Max

Рендеринг – процесс визуализации сконструированной в редакторе пространственной сцены. Типы подключаемых модулей, которые ближе всего относятся к рендерингу в

нашем контексте: Shadow Generator или Render. Для наших целей достаточно разработать модуль типа Shadow Generator, т.е. необходимо реализовать интерфейс, основным методом которого является `ShadowGenerator::Sample(ShadeContext&, Point3&, Color&)`. Он возвращает значение равно степени затененности точки пересечения луча со сценой (0 – точка в тени, 1 – точка освещена; промежуточные значения соответствуют полутени).

Использование модуля типа Render означает создание своего уникального построителя изображений (в данном случае еще и трассировщика лучей). Наш Shadow Generator использует стандартный трассировщик лучей.

### 3.2 Пользовательский интерфейс

Для каждого точечного источника освещения 3ds Max позволяет определить, будет ли он отбрасывать тень, и если да, то какой алгоритм будет применен для генерации теней от него, и задать параметры работы этого алгоритма для данного источника. Набор параметров зависит от конкретного алгоритма и задается его программной реализацией.

При выборе источника освещения, на вкладке General Parameters панели настройки Modify можно указать алгоритм генерации теней для этого источника. При выборе определенного алгоритма, на панели настройки появляется вкладка с опциями этого алгоритма.

Подключение к 3ds Max модуля FastLMM добавляет в список алгоритмов опцию «Fast Light Mesh Method». При выборе этой опции на панели настройки появляется вкладка Parameters, где можно редактировать необходимые параметры.

Интерфейс, созданный в рамках данной работы, выполнен в том же стиле, что и пользовательский интерфейс стандартных алгоритмов генерации теней, таких как «Area Shadows», «Shadow Map», «Ray Traced Shadows».

Для модуля FastLMM доступны следующие параметры:

- Алгоритм визуализации – MCC, MCC+ATK.
- Express – визуализация алгоритмом с возможными протечками (Light/Shadow Leaks). Данный параметр позволяет генерировать тени значительно быстрее, за счет того, что при построении интерполяционного множества не учитывается короткий тест – проверка локальной видимости  $V(P, x_j)$ . Такая визуализация возможна для случаев, когда необходимо отрисовать сцену быстро, например, для черного варианта.
- Quality – Качество изображаемой тени. При увеличении данного параметра, увеличивается количество световых точек в световой сетке. Шаг световой сетки вычисляется по формуле  $h = r / q$ , где  $r$  – радиус,  $q$  – качество.
- Radius – радиус размытия тени. Чем больше радиус, тем более размыты границы теней как вовне, так и вовнутрь границы четкой тени.
- Size – размер буфера глубины или теневой карты (только для MCC+ATK). Чем больше размер буфера, тем точнее теневая карта отражает видимость световых точек, и, следовательно, получаются более качественные изображения.

### 3.3 Подключаемый модуль FastLMM

Для реализации комплексированного алгоритма, было решено перенести расчет теневой карты на графический ускоритель. Для этого был использован DirectX 9. Сцена, с позиции источника освещения, отрисовывается в текстуру, Далее буфер считывается в память центрального процессора и используется для расчета видимости световых точек в алгоритме MCC+ATK.

### 4. ЧИСЛЕННЫЕ ЭКСПЕРИМЕНТЫ

Проведенные эксперименты можно разделить на две группы: по качеству и по скорости. Были проведены расчеты для сцен различной сложности. Для двух сцен данные экспериментов см. в таблицах 1 и 2: «Сферы» – 2028 треугольников; «Парк» – 145970 треугольников.

Приведены изображения: расчет по ОРЛТ (рис. 1), расчет по MCC (рис. 2), расчет по экспресс MCC (рис. 3). На рис. 4 приведена разница между изображениями на рис. 2 и рис. 3. Это показывает, что для ряда сцен возможные протечки света или тени почти не проявляются визуально.



Рис 1: Сцена «Парк», рассчитанная по ОРЛТ

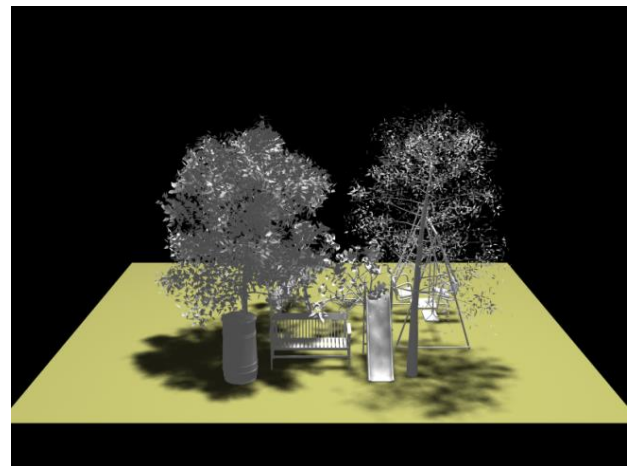


Рис 2: Сцена «Парк», рассчитанная MCC



Таблица 1. Времена расчета сцены «Сферы» (с)

	q = 6, r = 4.5, ТК: 1024	q = 6, r = 4.5, ТК: 512	q = 6, r = 5.5, ТК: 1024	q = 6, r = 5.5, ТК: 512
MCC	42.6	42.6	77.6	77.6
MCC+ATK	41.5	42.2	76.8	76.3
Express MSS	6.52	6.2	12.8	12.8
Express MCC+ATK	6.4	6.33	9.53	9.52

Таблица 2. Времена расчета сцены «Парк» (с)

	q = 4, r = 3, ТК: 1024x1024	q = 4, r = 3, ТК: 512x512	q = 4, r = 4, ТК: 1024x1024	q = 4, r = 4, ТК: 512x512
MCC	65.1	65.1	146.1	146.1
MCC+ATK	62.7	59.5	142.8	141.1
Express MCC	12.65	12.65	13.3	13.3
Express MCC+ATK	4.0	3.98	5.2	5.1

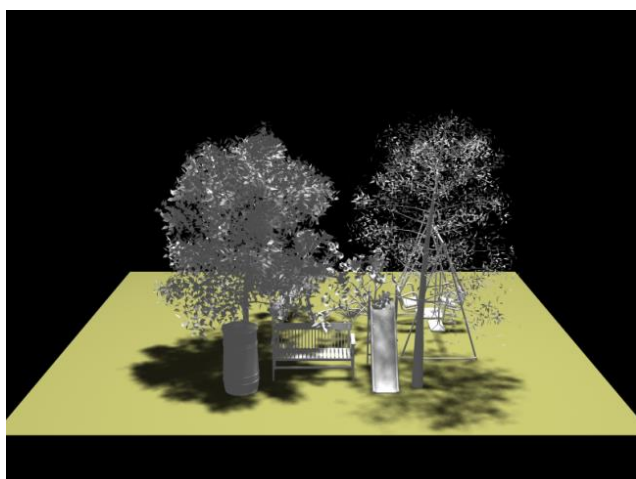


Рис 3: Сцена «Парк», экспресс режим MCC

Погрешность, вносимая алгоритмом MCC+ATK и экспресс режимом в расчет по MCC, составляет в среднем несколько процентов, что достаточно мало в контексте задачи генерации мягких теней. Таким образом, предложенные алгоритмы позволяют ускорить расчет изображений по MCC без ощутимых потерь качества.

## 5. ЗАКЛЮЧЕНИЕ

В результате выполнения данной работы был реализован метод световых сеток в форме подключаемого модуля для популярного графического редактора Autodesk 3ds Max. Реализовано ускорение расчета на основе применения теневых карт.

В качестве дальнейшего развития предполагаются два основных варианта:

1. Перенос всех вычислений на графический ускоритель. При этом использовать более

современное средство работы с видеокартой – архитектуру CUDA.

2. Реализовать комплексирование MCC с АТО в виде плагина 3ds Max. Несмотря на то, что АТО медленнее ATK, он позволяет получать более качественные изображения, содержащие меньше артефактов.

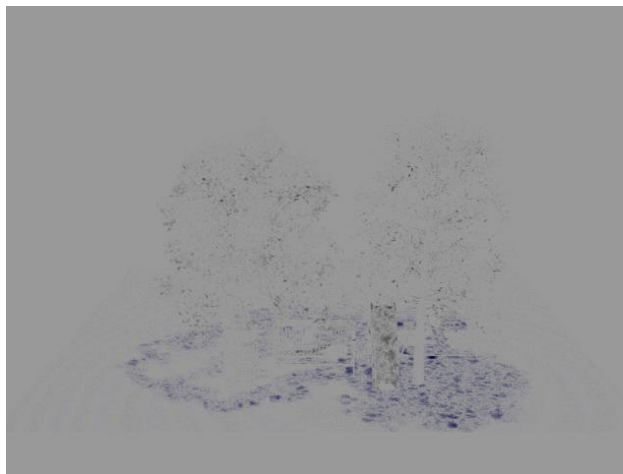


Рис 4: Разность изображений 2 и 3, усиленная по яркости и контрасту, затем негатив

## 6. БЛАГОДАРНОСТИ

Работа была выполнена при финансовой поддержке РФФИ в рамках научного проекта № 12-07-00386 а. Автор также выражает благодарность В.А. Дебелову и Л.Ф. Васильевой за постановку задачи и постоянное внимание к работе.

## 7. ССЫЛКИ

- [1] 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 Графikon'2001. – Нижний Новгород, 2001. – С. 18–24. 21.
- [2] 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.
- [3] Crow F. *Shadow Algorithms for Computer Graphics* // *Computer Graphics*. – 1977. – Vol. 11, № 2. – P. 242-247.
- [4] Debelov V.A., Novikov I., Vasilyeva L.. *Ray Tracing Based on Interpolation of Visibility of Point Light Sources* // *Intelligent Computer Graphics 2012 / Plemenos, Dimitri; Miaoulis, Georgios (Eds.)*. – Springer Series: Studies in Computational Intelligence, 2013, Volume 441/2013, P. 169-188. DOI: 10.1007/978-3-642-31745-3\_9.
- [5] Whitted T. *An Improved Illumination Model for Shaded Display / Commun. ACM*. – 1980. – Vol. 23, № 6. – P. 343-349.
- [6] Williams L., *Casting curved shadows on curved surfaces* // *Computer Graphics*. – 1978. – Vol. 10, № 2. – P. 270-274.

## Об авторе

Наталья Бологова получила квалификацию бакалавра в 2013г. на факультете информационных технологий НГУ.

# АЛГОРИТМ ФИЛЬТРАЦИИ ПАПИЛЛЯРНЫХ ЛИНИЙ, НАЛОЖЕННЫХ НА РЕГУЛЯРНУЮ СТРУКТУРУ

Суслов Александр Алексеевич, Южно-Уральский государственный университет, Челябинск, Россия

[suslov\\_aa@sonda.ru](mailto:suslov_aa@sonda.ru)

Гудков Владимир Юльевич, Челябинский государственный университет, Челябинск, Россия

[diana@sonda.ru](mailto:diana@sonda.ru)

## Аннотация

В статье рассматривается метод фильтрации папиллярных линий, наложенных на регулярную структуру, имеющую частотные характеристики, близкие к частотным характеристикам самих папиллярных линий. В таких условиях трудно отделить полезную информацию от фоновой, используя частотные методы фильтрации. Предлагается один из возможных алгоритмов такого разделения, основанный на использовании градиентных методов обработки изображений. Итогом работы является законченная интерактивная вычислительная процедура.

**Ключевые слова:** оператор Собеля, оператор Лапласа, преобразование Фурье.

## 1. ВВЕДЕНИЕ

Эффективность работы дактилоскопической системы зависит от ее способности обрабатывать изображения следов или отпечатков очень низкого качества, когда на изображении присутствуют различного рода графические помехи, а именно: наложенный текст разного размера, другой след (отпечаток), линии одинарные и множественные различной толщины. Существуют базы данных таких изображений с характерными помехами, в частности, база изображений NIST [1]. В настоящей статье рассматривается случай, когда помехой являются множественные параллельные линии, толщина которых примерно совпадает с толщиной папиллярных линий самого изображения отпечатка.

## 2. ПОСТАНОВКА ЗАДАЧИ

В настоящей работе рассматриваются изображения, дискретизированные с разрешением 500 точек на дюйм и хранящиеся в виде массива пикселей с градацией 256 оттенков серого цвета [2].

Примером такого изображения может служить изображение на рис. 1, взятое из базы изображений NIST. Оно имеет две отличительные особенности.

Во-первых, на нем в качестве фона присутствует структура параллельных линий с толщиной и периодичностью такой же, как у отпечатка, расположенного в центре. Поэтому использовать частотную фильтрацию, а именно, преобразование Фурье для подавления фона по отношению к изображению отпечатка, не представляется возможным. На рис. 2 приведен частотный спектр, полученный в результате применения Фурье-преобразования к исходному изображению. Для визуализации спектра коэффициенты Фурье-преобразования были пересчитаны по формуле:

$$a_{ij} = 128 - 30 * \log_{10} re_{ij},$$

где  $a_{ij}$  - массив пикселей, предназначенный для визуализации,

$re_{ij}$  - массив вещественной части коэффициентов Фурье-преобразования [3]. Из рисунка видно, что наиболее значимые коэффициенты сосредоточены компактной группой в центре изображения, что не позволяет идентифицировать наличие помехи в виде регулярных линий той же ширины, что и линии папиллярного узора.

Во-вторых, изображение обладает высокой контрастностью. На нем имеются как очень светлые, так и очень темные участки, прилегающие друг к другу. А, следовательно, четко выраженные границы между ними. Поэтому применение градиентных методов обработки на основе дискретных операторов Собеля [4] непосредственно к исходному изображению не приведет к желаемому результату, так как наибольшие градиенты находятся не в области перекрытия отпечатка и регулярных наклонных линий.



Рис. 1: Исходное изображение

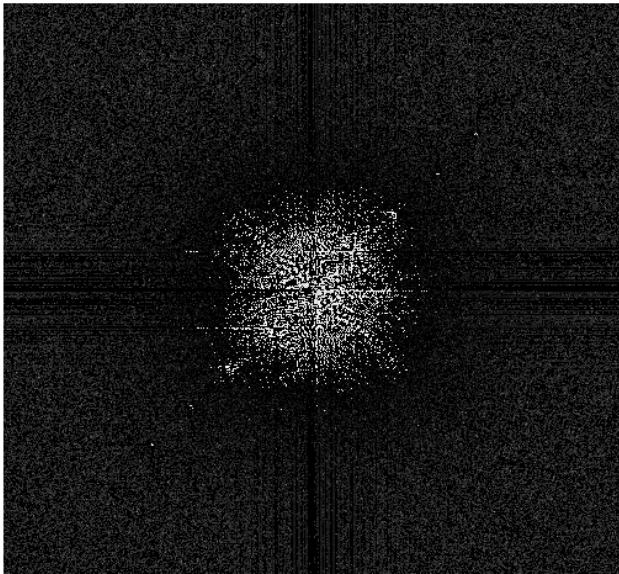


Рис. 2: Фурье-спектр исходного изображения

### 3. МЕТОД РЕШЕНИЯ

С учетом перечисленных выше характерных особенностей помех необходимо выполнить такую обработку изображения, после применения которого оно стало бы более однородным. Выполним следующую процедуру: для каждого пикселя вычислим среднее значение яркости для восьми прилегающих к нему соседних пикселей, вычтем его из яркости самого пикселя.

$p_6$	$p_7$	$p_8$
$p_5$	$p_0$	$p_1$
$p_4$	$p_3$	$p_2$

Рис. 2: Нумерация соседних пикселей

Получится новый массив значений, содержащий как положительные, так и отрицательные значения. Или, используя обозначения пикселей, приведенные на рис. 2, указанную процедуру можно выразить в виде формулы:

$$n_{ij} = p_0 - \frac{1}{8} \sum_{k=1}^8 p_k.$$

Такая процедура может быть названа вычислением высокочастотного шума  $n_{ij}$  [5]. Чтобы можно было визуализировать этот массив  $n_{ij}$ , умножим все его значения на коэффициент усиления  $k (\approx 4)$  и прибавим ко всем его значениям константу 128, так чтобы все значения лежали в диапазоне 0 – 255. Исходное изображение, будучи подвергнуто указанной процедуре, будет выглядеть так, как на рис. 3.

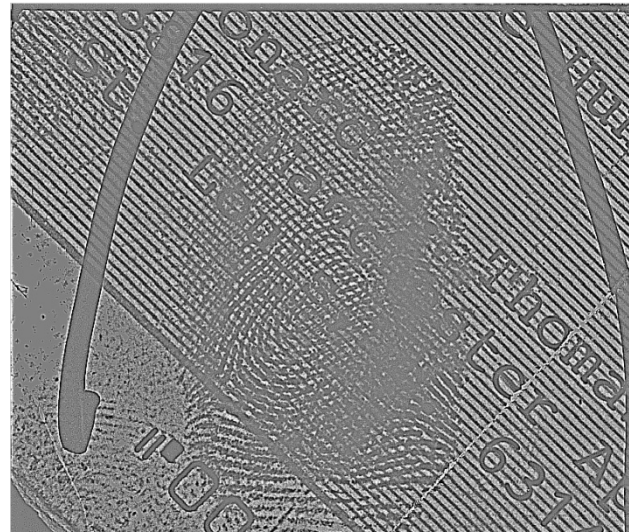


Рис. 3: Высокочастотный шум

Очевидно, изображение стало более однородным и напоминает исходное. Более светлым участкам исходного изображения соответствуют большие значения шума. Наоборот, более темным участкам исходного изображения соответствуют меньшие значения шума. В данном случае чтобы отфильтровать помехи, нужно погасить регулярную структуру линий, идущих слева сверху направо вниз как вне, так и внутри отпечатка. Для этого выполним следующую процедуру.

Вычислим для массива  $n_{ij}$  градиенты яркости, используя разностные операторы Собеля по четырем направлениям: горизонтальному ( $G_1$ ), вертикальному ( $G_3$ ) и двум диагоналям ( $G_2$  и  $G_4$ ):

$$G_1 = \text{abs}(p_8 - p_6 + 2 * (p_1 - p_5) + p_2 - p_4),$$

$$G_2 = \text{abs}(p_1 - p_7 + 2 * (p_2 - p_6) + p_3 - p_5),$$

$$G_3 = \text{abs}(p_4 - p_6 + 2 * (p_3 - p_7) + p_2 - p_8),$$

$$G_4 = \text{abs}(p_7 - p_5 + 2 * (p_8 - p_4) + p_1 - p_3).$$

В этих формулах используются принятые ранее обозначения для нумерации соседних пикселей.

Для рассматриваемого нами примера наиболее яркими и многочисленными будут градиенты в направлении  $G_4$ , а наименее яркими и малочисленными будут градиенты в направлении  $G_2$ . Выделим в каждом из этих массивов наиболее информативные элементы, а именно, те, которые имеют самые большие по модулю значения. Для этой цели вычислим гистограммы для каждого из этих массивов [1]. Находим минимальное  $\min$  и максимальное  $\max$  значения. Количество интервалов в гистограмме принимаем равным  $n = \max - \min + 1$ . Наконец элементы массива гистограмм  $h_k$  вычисляем по формуле:

$$k = n_{ij} - \min,$$

$$h_k = h_k + 1, \quad i = 1 \dots \text{rows}, \quad j = 1 \dots \text{cols},$$

где  $\text{rows}$  - число строк,

$\text{cols}$  - число столбцов массива  $n_{ij}$ .

Используя гистограмму градиентов, выберем самые большие по модулю значения. Для этого введем в рассмотрение

константу  $percent \approx 0.15$ . Тогда для нахождения граничного значения самых больших градиентов будем суммировать элементы гистограммы  $h_k$  до тех пор, пока выполняется неравенство

$$\sum_{k=k_{max}}^n h_k < percent.$$

Суммирование выполняется, начиная с максимального значения  $k$  в сторону его уменьшения.

Для визуализации картины распределения больших значений градиентов изобразим их светлыми пикселями на темном фоне, см. рис. 4.

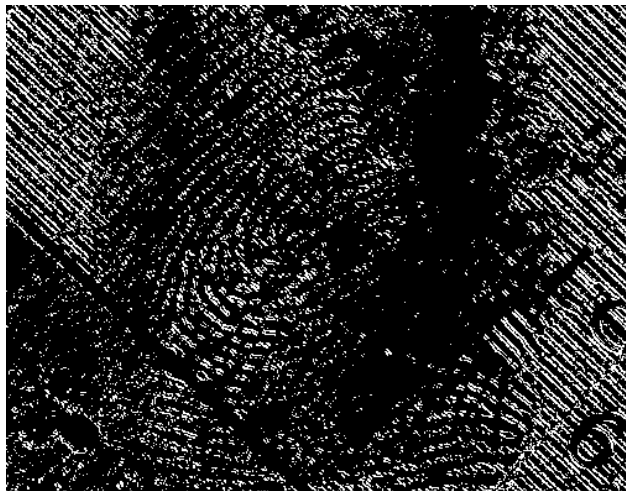


Рис.4: Визуализация расположения градиентов  $RtUpLtDn$

Из рисунка видно, что большие значения градиентов в направлении  $G_4$  располагаются по разные стороны от прямых линий. Легко понять, как располагаются большие значения градиентов в направлении  $G_2$  – по разные стороны от линий, идущих в направлении справа сверху налево вниз. Причем не исключен случай, когда для одного и того же пикселя градиенты принимают большие значения сразу в обоих направлениях, и  $G_4$ , и  $G_2$ . Поэтому для того, чтобы отфильтровать линии, проходящие в направлении слева сверху направо вниз, нужно выполнить следующую процедуру. В тех областях, где расположены самые большие градиенты  $G_4$ , а градиенты  $G_2$  не являются большими, нужно увеличить яркости пикселей на оригинальном изображении. Это приведет к тому, что толщина линий на оригинальном изображении уменьшится. Область больших градиентов сместится к центру каждой линии. В тех же областях, где расположены самые большие градиенты  $G_2$ , яркости пикселей на оригинальном изображении нужно уменьшить. Это предотвращает уменьшение толщин линий, идущих в направлении, перпендикулярном к направлению фильтрации.

Наиболее простым способом увеличения или уменьшения яркости пикселя является умножение его яркости на постоянный множитель, значение которого подбирается опытным путем из условия уменьшения градиентов.

Однократное применение указанной процедуры не приводит к полной фильтрации регулярной структуры линий. Однако возможно выполнение нескольких итераций, позволяющих достичь желаемого результата. На рис. 5 представлено отфильтрованное изображение для 15 итераций при

изменении яркостей пикселей на 10% от их первоначального значения.

Если количество итераций будет велико, то время, затрачиваемое на выполнение процедуры фильтрации, также увеличится. Одним из способов понижения временных затрат является использование операторов Лапласа [4], вычисляемых по тем же направлениям, что и при вычислении градиентов. Как известно, оператор Лапласа является разностным приближением второй производной изучаемой для функции одной переменной и может быть выражен формулой:

$$\Delta f = f_{i+1} + f_{i-1} - 2 * f_i,$$

где  $f_{i-1}, f_i, f_{i+1}$  – значения функции яркости в трех последовательных пикселях, расположенных вдоль какого-либо одного из направлений: горизонтального, вертикального или одного из двух диагональных.

Находя максимальные по модулю значения для функции изменения яркости в одном из четырех перечисленных выше направлений, получим пиксели, расположенные как раз посередине достаточно тонких линий, либо между ними. Соответственно, осветляя темные пиксели и затемняя светлые, можно сократить количество итераций в процедуре фильтрации. В данном случае достаточно тонкие линии – это такие линии, ширина которых сравнима с шириной линий папиллярного узора. Если ширина линии возрастает, то точки, лежащие на ее середине, не будут являться точками экстремальных значений функции яркости, следовательно, лапласиан не будет принимать большие по модулю значения. В связи с этим необходимо разумное сочетание способов фильтрации на основе операторов Собеля и операторов Лапласа.



Рис. 5: Отфильтрованное изображение

#### 4. ЗАКЛЮЧЕНИЕ

Разработан алгоритм фильтрации папиллярных линий, наложенных на регулярную структуру линий, имеющую частотные характеристики, близкие к частотным характеристикам самих папиллярных линий. Алгоритм основан на вычислении высокочастотного шума изображения, последующем применении дискретных операторов Собеля и Лапласа применительно к шуму по

четырем различным направлениям и изменении яркостей пикселей оригинального изображения в зависимости от значений градиентов шума.

На данный момент процедура носит интерактивный характер. Это связано с тем, что для того, чтобы ее запустить, необходимо выяснить характер присутствующих на изображении помех. В связи с тем, что использование частотных методов в данном случае проблематично, предполагается в дальнейшем изучить вопрос о применении когерентных свойств градиентов высокочастотного шума по различным направлениям.

## 5. ЛИТЕРАТУРА

[1] NIST Special Database 27, Fingerprint Database/ C.I. Watson and C.L. Wilson. – National Inst. of Standards and Technology, 1992.

[2] Компьютерное зрение. Современный подход / Д.А. Форсайт, Ж. Понс; пер. с англ. А.В. Назаренко, И.Ю. Дорошенко. – М.: Изд. Дом Вильямс, 2004. – 928 с.

[3] Обработка и анализ изображений в задачах машинного зрения / Ю.В. Визильтер, С.Ю.Желтов, А.В.Бондаренко и др. – М.: Физматкнига, 2010. – 672 с.

[4] Цифровая обработка изображений / Б. Яне; пер. с англ. А.М. Измайловой. - М.: Техносфера, 2007. – 584 с.

[5] Цифровая обработка изображений / Р. Гонсалес, Р. Вудс; пер. с англ.; под ред. П.А. Чочиа. – М.: Техносфера, 2006. – 1072 с.

# Гомеоморфное отображение циркулярных фигур

Никита Ломов

Факультет вычислительной математики и кибернетики

Московский государственный университет имени М.В. Ломоносова, Москва, Россия

nikita-lomov@mail.ru

## Аннотация

В статье предложен метод построения гомеоморфизма двух замкнутых односвязных областей, представленных в виде силуэтов циркулярных фигур. Цель работы — построить такое гомеоморфное отображение, в котором образ будет сохранять содержательный смысл прообраза, в частности, сохранится взаимное расположение значимых частей силуэта. Описан способ сегментации изображения на основе его циркулярного представления. Метод может быть использован для приложенных в компьютерной графике и для решения задач распознавания изображений.

**Ключевые слова:** гомеоморфизм, циркулярная фигура, сегментация изображений, нормализованные координаты.

## 1. ВВЕДЕНИЕ

Вследствие теоремы Римана об отображении, две односвязные замкнутые области на плоскости топологически эквивалентны, т.е. существует непрерывное взаимно-однозначное отображение одной области на другую, обратное к которому также непрерывно. Теоретически такое отображение может быть построено бесконечным числом способов. Однако для приложений в компьютерной графике и распознавании изображений интерес представляют далеко не все такие отображения, а лишь те из них, которые сохраняют «содержательный смысл» прообраза. Например, подобные отображения могут найти применение в традиционной анимации. При производстве рисованного фильма художники-прорисовщики создают только ключевые кадры, а задача отрисовки промежуточных фаз ложится на плечи художников-фазовщиков. Этот трудоемкий процесс можно автоматизировать, разработав методы построения силуэтов персонажей и предметов в промежуточных кадрах и раскраски этих силуэтов по эталонам, заданным ключевыми кадрами. Способы автоматического создания промежуточных фаз анимации описаны в работах [1] и [2].

Широкими возможностями для анализа и модификации формы изображения обладает его циркулярное представление [3, стр. 189]. Идея подхода заключается в том, чтобы для исходного дискретного представления фигуры получить адекватное непрерывное представление в виде семейства кругов с центрами в точках некоторого графа, который называют осевым. Мы будем рассматривать циркулярную фигуру как совокупность отдельных штрихов — жирных линий [4], каждая из которых соответствует одному из ребер осевого графа. Предложенный в данной статье способ сегментации изображений использует принцип, похожий на описанный в работе [5]: мы выделяем в изображении ключевые линии и относим каждую точку изображения к какой-либо из этих линий. Гомеоморфизм будем строить так, чтобы для каждой точки сохранить при отображении значение некоторой функции, характеризующей близость точки к осевому графу циркулярной фигуры.

## 2. СЕГМЕНТАЦИЯ ЦИРКУЛЯРНЫХ ФИГУР

Под *сегментацией* (декомпозицией) фигуры будем понимать ее разбиение на конечное множество областей (сегментов).

Аналогично [6] будем разбивать фигуру на собственные области ребер графа (только не скелетного, а осевого), в результате получим *осевую сегментацию*. Для описания положения точки в жирной будем использовать *циркулярные координаты* [3, стр. 199]. *Концевыми округлениями*  $R_a$  и  $R_b$  назовем множества точек жирной кривой, для которых  $\tau = 0$  и  $\tau = 1$  соответственно.

Пусть циркулярная фигура состоит из  $n$  жирных линий, для каждой из которых  $t \in [a, b]$ . Введем следующие обозначения:

- $F_k$ ,  $k = 1, \dots, n$  — жирные линии;
- $P_k$ ,  $k = 1, \dots, n$  — их оси, т.е. ребра осевого графа;
- $\lambda_k(M)$ ,  $\tau_k(M)$ ,  $\theta_k(M)$ ,  $\sigma_k(M)$  — циркулярные координаты точки  $M$  в  $k$ -ой жирной линии;
- $I_M$  — множество номеров тех жирных линий, которым принадлежит точка  $M$ ;
- $\rho_k(M)$  — минимальное расстояние от точки  $M$  до края дуги конечного округления слоя глубины  $\lambda(M)$ .

Объединение всех  $\lambda$ -слоев жирных линий, образующих циркулярный граф  $C$ , будем называть  $\lambda$ -силуэтом этого графа. Очевидно, что каждая точка циркулярной фигуры является граничной для некоторого  $\lambda$ -силуэта. Назовем величину  $\lambda$  *уровнем* соответствующей точки. Значение  $\lambda$  характеризует близость точки к осевому графу циркулярной фигуры и равно минимальной глубине точки в тех жирных линиях, составляющих циркулярную фигуру, которым эта точка принадлежит:  $\lambda = \min_{k \in I_M} \lambda_k$ . Мы дадим такое определение собственной области, которое позволит сохранить уровень точки в циркулярной фигуре при гомеоморфном отображении.

*Собственной областью*  $S_k$  ребра  $P_k$  (жирной линии  $F_k$ ) осевого графа  $T$  будем называть объединение множеств  $A_k$  и  $B_k$ :  $S_k = (A_k \cup B_k)$ , где

- $A_k = \left\{ M \in F_k \mid \tau_k(M) \in (0, 1); \lambda_k(M) = \min_{i \in I_M} \lambda_i \right\}$ ;
- $B_k = \left\{ M \in F_k \mid \tau_k(M) \in \{0, 1\}; \rho_k(M) = \min_{i \in I_M, \lambda_i(M) = \lambda_k(M)} \rho_i(M) \right\}$ .

При таком определении собственной области на границе между двумя смежными областями  $S_k$  и  $S_m$  выполнено условие  $\lambda_k = \lambda_m$ . Из этого следует, что для последовательностей  $\{K^i\}$  и  $\{M^i\}$ , составленных из точек собственных областей  $S_k$  и  $S_m$  соответственно и сходящихся к одной и той же точке границы, соответствующие им последовательности глубин  $\{\lambda_k^i\}$  и  $\{\lambda_m^i\}$  имеют равные пределы.

Пример сегментации циркулярной фигуры приведен на рис. 1. Два скелета считают *изоморфными* [7], если они изоморфны как графы и при обходе вершин сохраняется последовательность висячих вершин, соответствующих друг другу в изоморфизме графов. Применяв это определение не к скелетам, а к осевым графам, введем понятие изоморфизма осевых графов. Заметим, что ребра осевых графов можно сделать ориентированным таким образом, чтобы графы остались изоморфными. *Графом смежности сегментации* будем называть плоский граф, вершины которого соответствуют сегментам, две верши-

ны соединены ребром, когда соответствующие им сегменты являются смежными в разбиении (то есть имеют общую границу). Сегментации называют изоморфными, если изоморфны их графы смежности. Чтобы гарантированно получить изоморфные сегментации из циркулярных фигур с изоморфными осевыми графами, нужно предъявить к циркулярам дополнительные требования. Если выполняются следующие условия:

- жирные линии пересекаются только в том случае, если их осевые линии — смежные ребра;
- для любой пары различных жирных кривых  $F_k$  и  $F_m$   $F_k \setminus F_m$  — связное множество,  $F_k \cap F_m$  — связное или пустое множество;
- любая жирная кривая имеет участок границы, который не пересекается ни с какой другой жирной кривой;

собственные области жирных кривых будут иметь общую границу только тогда, когда оси этих жирных линий имеют общую вершину и при обходе ребер, инцидентных этой вершине, по кругу осевые линии следуют друг за другом. Если рассматривать концевые округления висячих вершин в качестве отдельных сегментов, необходимо также потребовать, чтобы они не пересекались с другими жирными линиями. Циркуляр, который отвечает вышеуказанным требованиям, будем называть циркуляром *общего положения*.

### 3. ГРАНИЦЫ МЕЖДУ СОБСТВЕННЫМИ ОБЛАСТЯМИ

Будем считать, что ребра осевого графа  $T$  циркулярной фигуры  $S$  произвольным образом ориентированы. Ось  $P$  разделяет собственную область  $S$  на две части. Будем называть *правой* собственной областью ту часть, граница которой содержит часть правой огибающей жирной линии  $F$  (возможно, всю огибающую), *левой* собственной областью — область, границе которой принадлежит часть левой огибающей. Важно, что после такого подразделения изоморфные осевые сегментации остаются изоморфными. В дальнейшем под собственной областью ребра будем понимать его правую или левую собственную область, а не их объединение.

*Разрезом* будем называть общую часть границы двух смежных собственных областей. Пусть  $C$  — циркуляр общего положения. Рассмотрим жирные линии  $F_1 = (u_1(t), v_1(t), r_1(t))$  и  $F_2 = (u_2(t), v_2(t), r_2(t))$ , имеющие общий концевой круг. Не ограничивая общности, будем считать, что он является начальным в обеих линиях. Пусть при обходе ребер, инцидентных общей вершине осей жирных линий, по часовой стрелке оси  $P_1$  и  $P_2$  следуют друг за другом. Это означает, что правая собственная область  $F_1$  имеет общую границу с левой собственной областью  $F_2$ . Рассмотрим взаимное расположение правой огибающей  $\lambda$ -слоя  $F_1$  и левой огибающей  $\lambda$ -слоя  $F_2$ .

Уравнения правой огибающей [3, стр. 184]:

$$\begin{cases} x_1(t_1, \lambda) = u_1 - \frac{\lambda^2 u_1' r_1' r_1}{u_1'^2 + v_1'^2} + \frac{\lambda v_1' r_1 \sqrt{u_1'^2 + v_1'^2 - \lambda^2 r_1'^2}}{u_1'^2 + v_1'^2}, \\ y_1(t_1, \lambda) = v_1 - \frac{\lambda^2 v_1' r_1' r_1}{u_1'^2 + v_1'^2} - \frac{\lambda u_1' r_1 \sqrt{u_1'^2 + v_1'^2 - \lambda^2 r_1'^2}}{u_1'^2 + v_1'^2}. \end{cases} \quad (1)$$

Уравнения левой огибающей:

$$\begin{cases} x_2(t_2, \lambda) = u_2 - \frac{\lambda^2 u_2' r_2' r_2}{u_2'^2 + v_2'^2} - \frac{\lambda v_2' r_2 \sqrt{u_2'^2 + v_2'^2 - \lambda^2 r_2'^2}}{u_2'^2 + v_2'^2}, \\ y_2(t_2, \lambda) = v_2 - \frac{\lambda^2 v_2' r_2' r_2}{u_2'^2 + v_2'^2} + \frac{\lambda u_2' r_2 \sqrt{u_2'^2 + v_2'^2 - \lambda^2 r_2'^2}}{u_2'^2 + v_2'^2}. \end{cases} \quad (2)$$

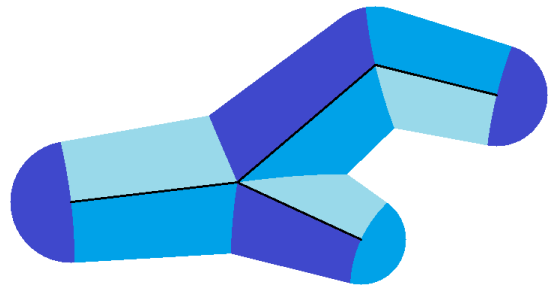


Рис. 1: Разбиение фигуры на собственные области

Для того чтобы определить точку пересечения огибающих, необходимо решить систему уравнений относительно  $(t_1, t_2)$ :

$$\begin{cases} x_1(t_1, \lambda) = x_2(t_2, \lambda), \\ y_1(t_1, \lambda) = y_2(t_2, \lambda). \end{cases} \quad (3)$$

Если решение системы  $(t_1, t_2), t_1, t_2 \in [0, 1]$  существует и единственно при всех  $\lambda \in [0, 1]$ , то функции  $t_1 = t_1(\lambda), t_2 = t_2(\lambda)$  непрерывно дифференцируемы, поскольку непрерывно дифференцируемы функции  $u_1', v_1', r_1', u_2', v_2', r_2'$ , и разрез задается уравнениями

$$\begin{cases} x(\lambda) = x_1(t_1(\lambda), \lambda), \\ y(\lambda) = y_1(t_1(\lambda), \lambda). \end{cases} \quad (4)$$

Такой разрез будем называть разрезом *1-го типа*.

Если система имеет решение только при  $\lambda = 0$ , то границы  $\lambda$ -слоев жирных линий имеют общую часть концевой дуги. Согласно определению собственной области, разрезу принадлежит точка дуги, равноудаленная от ее концов, т.е. разделяющая дугу пополам. Угловая величина  $\alpha(\lambda)$  общей дуги является непрерывно дифференцируемой функцией на интервале  $[0, 1]$ , следовательно, функции  $x(\lambda)$  и  $y(\lambda)$ , задающие разрез, также непрерывно дифференцируемы. Разрезы, состоящие из центральных точек общих концевых дуг, будем называть разрезами *2-го типа*.

Для терминального ребра, ориентированного к висячей вершине (от висячей вершины), разрез между собственной областью и концевым округлением задается уравнением  $t = 1$  ( $t = 0$ ) и также имеет одну и только одну точку уровня  $\lambda, \lambda \in [0, 1]$ .

## 4. РАСЧЕТ ГОМЕОМОРФИЗМА

### 4.1 Нормализованные координаты

*Верхним разрезом* собственной области будем называть разрез, которому принадлежит конечная точка оси, соответственно, *нижним разрезом* — разрез, содержащий начальную точку оси. *Линией  $\lambda$ -уровня* собственной области будем называть лежащую в собственной области кривую, заданную уравнением  $\lambda = const$ .

Если собственная область содержит часть концевого округления, поставим в соответствие каждой точке концевого округления пару чисел  $(t, \lambda)$ , где

- $\lambda$  — соответствующая циркулярная координата;
- для точек верхнего концевого округления  $t \geq 1$ , нижнего концевого округления  $t \leq 0$ ;
- значения  $(x'(t, \lambda))^2 + (y'(t, \lambda))^2$  на дугах вида  $\lambda = \lambda_0, t \geq 1$  ( $t \leq 0$ ) постоянны;
- значения частных производных  $\frac{\partial x(t, \lambda)}{\partial t}, \frac{\partial y(t, \lambda)}{\partial t}$  на концах дуг совпадают с соответствующими значениями на концах огибающих 1–2.

Теперь мы можем задать верхний и нижний разрезы (как 1-го, так и 2-го типа) функциями  $t = g(\lambda), t = h(\lambda) \in C^1[0, 1]$  и получить взаимно однозначное соответствие между точками собственной области и множеством  $\{\lambda \in [0, 1], t \in [g(\lambda), h(\lambda)]\}$ . Отметим, что функции  $x(t, \lambda)$  и  $y(t, \lambda)$  непрерывно дифференцируемы всюду в собственной области.

Наконец, приведем все координаты к единому диапазону. Определим относительное положение точки с координатами  $(t_0, \lambda_0)$  на линии  $\lambda$ -уровня:

$$\mu(t_0, \lambda_0) = \frac{\int_{g(\lambda_0)}^{t_0} \sqrt{(x'(\tau, \lambda_0))^2 + (y'(\tau, \lambda_0))^2} d\tau}{\int_{g(\lambda_0)}^{h(\lambda_0)} \sqrt{(x'(\tau, \lambda_0))^2 + (y'(\tau, \lambda_0))^2} d\tau}. \quad (5)$$

Пару чисел  $(\mu, \lambda), \mu, \lambda \in [0, 1]$ , будем называть *нормализованными координатами* точки в собственной области. Таким образом, мы получили взаимно-однозначное соответствие между точками собственной области и множеством  $[0, 1] \times [0, 1]$ . Например, точка  $M$  на рис. 2 имеет нормализованные координаты  $(0,7; 0,6)$ .

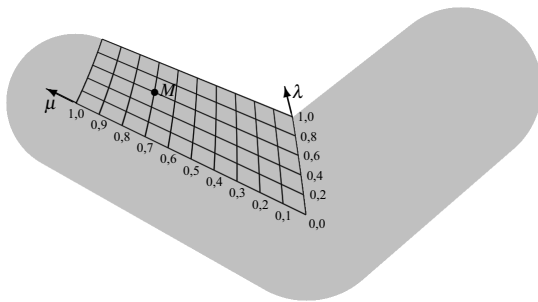


Рис. 2: Координатная сетка собственной области

Нормализованными координатами точки концевой округления будем называть пару чисел  $(\theta, \lambda), \theta, \lambda \in [0, 1]$ , имеющих те же значения, что и циркулярные координаты этой точки в жирной линии.

### 4.2 Доказательство гомеоморфности отображения

**Утверждение 1.** Отображение  $f : S \rightarrow [0, 1] \times [0, 1]$ , которое ставит в соответствие точке собственной области ее нормализованные координаты, является гладким гомеоморфизмом. **Доказательство.** Взаимная однозначность отображения очевидна, покажем его гладкость. Отображения  $(t, \lambda) \rightarrow (\mu, \lambda)$  и  $(\mu, \lambda) \rightarrow (t, \lambda)$  — гладкие. Отображение  $(t, \lambda) \rightarrow (x, y)$  также гладкое, и векторы касательных к кривым вида  $t = t_0$  и  $\lambda = \lambda_0$  нигде в собственной области не коллинеарны. По теореме о разрешимости системы функциональных уравнений [8, стр. 609] всюду в собственной области определено гладкое отображение  $(x, y) \rightarrow (t, \lambda)$ . Тогда отображения  $(\mu, \lambda) \rightarrow (x, y)$  и  $(x, y) \rightarrow (\mu, \lambda)$  будут гладкими как суперпозиции гладких отображений.

**Утверждение 2.** Отображение  $f : R_a \rightarrow R_b$  одного концевой округления на другое с сохранением нормализованных координат является гладким гомеоморфизмом. **Доказательство.** Это следует из того, что угловая величина дуги концевой округления  $\alpha(\lambda) \in C^1[0, 1]$  (в точке  $\lambda = 0$  доопределим функцию по непрерывности), и одно концевое округление получается из другого равномерным растяжением его дуг.

**Утверждение 3.** Пусть  $C_1$  и  $C_2$  — циркуляры общего положения

с изоморфными осевыми графами и силуэтами  $D_1$  и  $D_2$ . Отображение  $Q : D_1 \rightarrow D_2$ , которое ставит в соответствие точке прообраза точку соответствующей в изоморфизме сегментаций собственной области с теми же нормализованными координатами, является гомеоморфизмом. **Доказательство.** Отображение между двумя собственными областями и двумя концевыми округлениями является гомеоморфным. Осталось рассмотреть последовательности точек  $\{M_n\}$ , сходящиеся к точке разреза. Если существует не менее трех собственных областей, содержащих бесконечное число точек последовательности, то последовательность сходится к вершине  $v$  осевого графа  $T_1$ ,  $\deg(v) \geq 3$ , и все подпоследовательности образов сходятся к соответствующей ей в изоморфизме вершине  $v'$  осевого графа  $T_2$ . Теперь рассмотрим случай, когда таких областей ровно две. Очевидно, что все предельные точки последовательности  $\{Q(M_n)\}$  принадлежат одному и тому же разрезу. Так как последовательность уровней точек сходится, и отображение сохраняет значение уровня, то все предельные точки  $\{Q(M_n)\}$  имеют уровень, равный предельному, а для каждого разреза такая точка существует и единственна. Таким образом, последовательность  $\{Q(M_n)\}$  имеет только одну предельную точку, и отображение является непрерывным. Для доказательства непрерывности обратного отображения, можно заметить, что если  $Q'$  — отображение  $D_2$  на  $D_1$ , построенное по тому же принципу, то отображение  $Q' \circ Q$  — тождественное.

### 4.3 Алгоритм расчета гомеоморфизма

Для того чтобы определить координаты  $(x, y)$  образа точки  $M$ , необходимо выполнить следующую последовательность шагов:

1. Определить множество жирных линий, содержащих точку  $M$ .
2. Определить  $\lambda$ -координаты точки в этих жирных линиях.
3. Выбрать жирную линию с наименьшим значением  $\lambda$ . Если таких линий несколько,
  - для точки, принадлежащей концевым округлениям, следует выбрать линию с минимальным расстоянием от точки до края огибающей  $\lambda$ -слоя;
  - для точки, не принадлежащей концевым округлениям, можно выбрать любую из жирных линий, поскольку точка принадлежит разрезу.
4. Определить сторону собственной области:
  - для точки из множества  $A$  сторона определяется циркулярной координатой  $\sigma$ ;
  - для точки из множества  $B$  — стороной огибающей, расстояние до края которой минимально.
5. Найти концы  $A$  и  $B$  линии  $\lambda$ -уровня — точки  $(t_1(\lambda), \lambda)$  и  $(t_2(\lambda), \lambda)$ , принадлежащие разрезам.
6. Вычислить нормализованную координату  $\mu = \frac{l(AM)}{l(AB)}$ .
7. Найти аналогичные концы линии  $\lambda$ -уровня в собственной области образа — точки  $A'$  и  $B'$ .
8. Определить координату  $t$  точки с тем же значением  $\mu$ .
9. Если  $t > 1$  или  $t < 0$ , вычислить циркулярную координату  $\theta$ .
10. Используя циркулярные координаты, определить положение точки на плоскости.

### 5. ПРИМЕР ПРОСТОГО ШАРНИРА

*Простым шарниром* назовем циркулярную фигуру, состоящую из двух жирных кривых Безье [4] первого порядка. Так как огибающие частей шарнира являются отрезками прямых, уравнения в системе 3 — линейные, и выражения для разреза



1-го типа можно получить в явном виде. Пусть

$$F_1 = (u_1(t), v_1(t), r_1(t)), F_2 = (u_2(t), v_2(t), r_2(t)),$$

$$u_i = u_0 + a_i t, v_i = v_0 + b_i t, r_i = r_0 + c_i t, i = 1, 2.$$

Обозначим  $l_1^2 = a_1^2 + b_1^2, l_2^2 = a_2^2 + b_2^2$ . Тогда система, задающая точки разреза, примет вид:

$$\begin{cases} a_1 t_1 - \frac{\lambda^2 a_1 c_1 (r_0 + c_1 t_1)}{l_1^2} \pm \frac{\lambda b_1 (r_0 + c_1 t_1) \sqrt{l_1^2 - \lambda^2 c_1^2}}{l_1^2} = \\ = a_2 t_2 - \frac{\lambda^2 a_2 c_2 (r_0 + c_2 t_2)}{l_2^2} \mp \frac{\lambda b_2 (r_0 + c_2 t_2) \sqrt{l_2^2 - \lambda^2 c_2^2}}{l_2^2}, \\ b_1 t_1 - \frac{\lambda^2 b_1 c_1 (r_0 + c_1 t_1)}{l_1^2} \mp \frac{\lambda a_1 (r_0 + c_1 t_1) \sqrt{l_1^2 - \lambda^2 c_1^2}}{l_1^2} = \\ = b_2 t_2 - \frac{\lambda^2 b_2 c_2 (r_0 + c_2 t_2)}{l_2^2} \pm \frac{\lambda a_2 (r_0 + c_2 t_2) \sqrt{l_2^2 - \lambda^2 c_2^2}}{l_2^2}. \end{cases} \quad (6)$$

Эталонная раскраска простых шарниров с различными комбинациями типов разрезов показана на рис. 3–4.

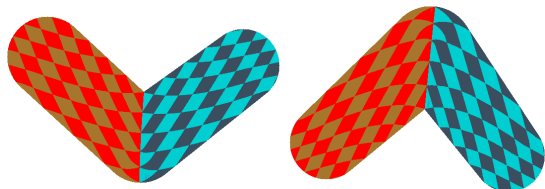


Рис. 3: Простые шарниры с разрезами разных типов

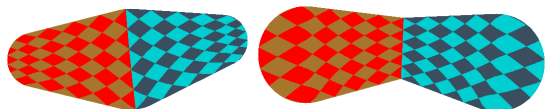


Рис. 4: Простые шарниры с разрезами одного типа

Простой шарнир можно использовать в качестве элементарного «строительного блока» при приближении формы фигуры простыми многоугольниками. При этом можно снять ограничение на гладкость осей и аппроксимировать их ломаными. Тогда линиями  $\lambda$ -уровня собственной области будут не границы  $\lambda$ -слоев, а границы  $\lambda$ -силуэтов. Другой возможный подход — использовать составные жирные кривые Безье с кривыми 1-й степени на концах. Для таких кривых поиск точки разреза сводится к решению системы линейных уравнений, как и в случае простого шарнира.

Наконец, можно развить предложенную в [9] идею ворпирования и строить контейнер в виде такой композиции простых жирных линий, которая бы полностью включала в себя преобразуемый объект. Пример преобразования растрового образа через аппроксимацию простым шарниром представлен на рис. 5–6.

## 6. ЗАКЛЮЧЕНИЕ

Предложенный в статье метод решает проблему неоднозначности, возникающую при преобразовании сложной композиции, составленной из нескольких пересекающихся жирных линий и существенно расширяет класс топологических преобразований над растровыми изображениями. Однако корректная работа метода возможна лишь в том случае, если рассматриваемые системы нелинейных уравнений имеют одно и только одно решение. Кроме того, поиск решения таких систем и

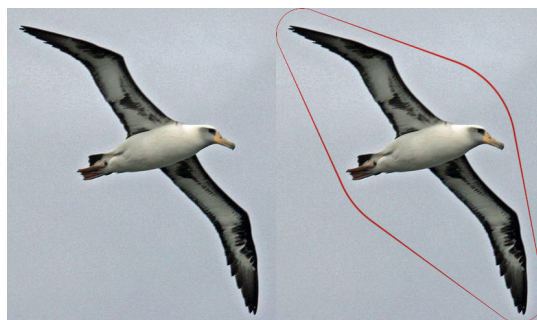


Рис. 5: Аппроксимация образа простым шарниром

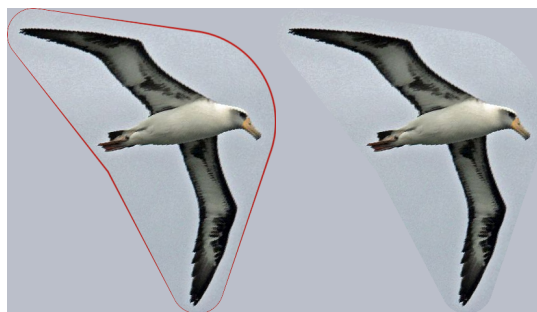


Рис. 6: Контейнерное преобразование растрового образа

необходимые преобразования координат требуют значительных вычислительных затрат. Эффективная программная реализация может открыть методу широкие перспективы для применения в компьютерной графике и решения задач распознавания.

## 7. СПИСОК ЛИТЕРАТУРЫ

- [1] W. Van Haevre, F. Di Fiore, F. Van Reeth. *Uniting Cartoon Textures with Computer Assisted Animation*. Proceedings of the 3rd International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia, 2005, pp. 245–253.
- [2] A. Kort. *Computer Aided Inbetweening*. Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering, 2002, pp. 125–132.
- [3] Л.М. Местецкий. *Непрерывная морфология бинарных изображений* — М.: ФИЗМАТЛИТ, 2009.
- [4] Л.М. Местецкий. *Компьютерная графика на основе жирных линий*. Тр. межд. конф. «Графикон-2000». — М.: МГУ, 2000. — С 165–173.
- [5] K. Melikhov, F. Tian, H.S. Seah, Q. Chen, J. Qiu. *Frame Skeleton Based Auto-Inbetweening in Computer Assisted Cel Animation*. Proceedings of the 2004 International Conference on Cyberworlds, 2004, pp. 216–223.
- [6] Л.Г. Домахина. *Об одном методе сегментации объектов для задач преобразования формы*. Труды 13-й Всероссийской конф. «Математические методы распознавания образов» (ММРО-13). С. 311–314.
- [7] Л.Г. Домахина, А.Д. Охлопков. *Изоморфные скелеты растровых изображений*. Тр. межд. конф. «Графикон-2008». — М.: МГУ, 2008.
- [8] В.А. Ильин, В.А. Садовничий, Бл.Х. Сендов. *Математический анализ (начальный курс), ч.1* — М.: МГУ, 1985.
- [9] Л.М. Местецкий, А.Б. Семенов. *Преобразование цветных изображений на основе жирных B-сплайновых кривых*. Тр. межд. конф. «Графикон-2003». — М.: МГУ, 2003. С. 235–242.

# Влияние Ложных Признаков На Качество Сравнения Дактилоскопических Изображений

Гудков Владимир Юльевич, Лепихова Дарья Николаевна  
 Челябинский государственный университет, Миасс, Россия  
[diana@sonda.ru](mailto:diana@sonda.ru), [daria.lepikhova@yandex.ru](mailto:daria.lepikhova@yandex.ru)

## Аннотация

В статье рассматривается представление дактилоскопических изображений (ДИ) при помощи графической структуры в виде графа; при этом идентификация изображений производится с помощью операций над графами. Для идентификации используются частные признаки ДИ. Ошибки детектирования частных признаков могут в значительной степени повлиять на точность идентификации.

**Ключевые слова:** дактилоскопические изображения, частный признак, граф ДИ

## 1. ВВЕДЕНИЕ

Представление ДИ в виде графа позволяет выполнить оценку подобия изображений при помощи операций над графами. В этом случае Графом ДИ  $G(V, E)$  называется совокупность непустого множества вершин  $V$  (частных признаков) и множества ребер  $E$  (связей) двухэлементных подмножеств множества  $V$ :

$$G(V, E) = \langle V; E \rangle, V \neq \emptyset, E \subset 2^V \text{ и } \forall e \in E (|e| = 2)$$

Вершина  $u \in V$  образует множество смежности

$$\Gamma^+(u) = \{v \in V | (u, v) \in E\}$$

На рис. 1 показаны три множества смежности

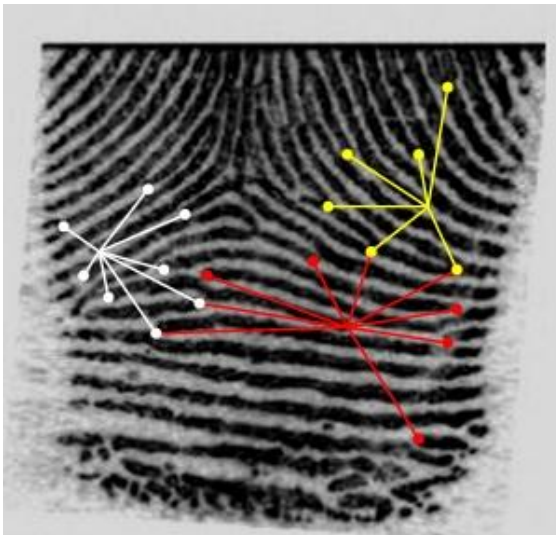


Рис. 1: Множества смежности частных признаков

На этапах обработки изображения и формирования шаблона возможно появление ложных или пропуск истинных частных признаков.

Maltoni упоминает, что, в соответствии с судебной системой США, для подтверждения идентичности отпечатков

необходимо совпадение не менее 12 частных признаков [3]. В других странах это число может меняться.

**Целью работы** является исследование зависимости качественных характеристик алгоритма сравнения ДИ от количества ложных частных признаков, используемых для сравнения.

## 2. ЧАСТНЫЕ ПРИЗНАКИ ИЗОБРАЖЕНИЯ

Список частных признаков узора  $L_m$  формируется в виде:

$$L_m = \{M_i = \{(x_i, y_i), \alpha_i, t_i, s_i, v_i, \theta_i, p_i, h_i\} | i \in 1..n_1\},$$

где  $M_i$  – частный признак,  $|L_m| = n_1$ ;

$i$  – индекс как номер частного признака;

$(x_i, y_i), \alpha_i, t_i, s_i, v_i, \theta_i, p_i, h_i$  – координаты, направление, тип, метка сомнительности, величина и направление кривизны, вероятность и плотность линий в окрестности частного признака.

Координаты  $(x_i, y_i)$  частного признака  $M_i$  определяются координатами вершины скелета. Направление  $\alpha_i$  как угол определяется связанными вершинами скелета. Тип  $t_i \in \{0; 1\}$ , где 0 – разветвление и 1 – окончание. Метка сомнительности  $s_i \in \{0; 1\}$  определяется положением частного признака, близким к неинформативным областям, причем 1 означает сомнительность.

Для сравнения используются только достоверные признаки, метка сомнительности для которых имеет значение 0.

## 3. СРАВНЕНИЕ ОТПЕЧАТКОВ КАК ГРАФОВ

Сравнение графа  $G_1(V_1, E_1)$  с графом  $G_2(V_2, E_2)$  основано на гомоморфизме  $f$ . Необходимо выбрать такие наборы операций  $\phi$  для графа  $G_1(V_1, E_1)$  и  $\psi$  для графа  $G_2(V_2, E_2)$ , что  $f(\phi(V_1)) = \psi(f(V_2))$ .

Для работы с шаблоном ДИ как с графом использовался следующий набор функций: удаление из шаблона частного признака, удаление связи между признаками, а также добавление в шаблон «ложных» признаков и связей. Перечисленные функции используют следующие операции над графами.

- Удаление признака из шаблона – удаление вершины графа.  

$$G_1(V_1, E_1) - v \in V_1 = G_2(V_2, E_2) | \{V_2 = V_1 - v \text{ и } E_2 = E_1 \setminus \{e = (v_1, v_2) | v_1 = v \vee v_2 = v\}\}$$
- Удаление связи между признаками – удаление ребра графа.  

$$G_1(V_1, E_1) - e \in E_1 = G_2(V_2, E_2) | \{V_2 = V_1 \text{ и } E_2 = E_1 - e\}$$
- Добавление признака в шаблон – добавление вершины.  

$$G_1(V_1, E_1) + v \notin V_1 = G_2(V_2, E_2) | \{V_2 = V_1 + v \text{ и } E_2 = E_1\}$$

- Добавление связи между признаками – добавление ребра (ложные ребра и вершины)

$$G_1(V_1, E_1) + e \in E_1 = G_2(V_2, E_2) \{V_2 = V_1 \& E_2 = E_1 + e\}$$

Операции вставки и удаления вершин, а также вставки и удаления ребер являются базовыми для работы с графом. На их основе можно реализовать ряд других операций, такие как стягивание подграфа, вставку и удаление множества смежности какой-либо вершины.

- Стягивание подграфа

$$\begin{aligned} G_1(V_1, E_1) / A | A \subset V_1 \\ = G_2(V_2, E_2) \{V_2 = (V_1 \setminus A) + v \& E_2 \\ = E_1 \setminus \{e = (u, w) | u \in A \vee w \in A\} \\ \cup \{e = (u, v) | u \in \Gamma(A) \setminus A\}\} \end{aligned}$$

- Вставка множества смежности вершины

$$\begin{aligned} G_1(V_1, E_1) + \Gamma^+(u) = G_2(V_2, E_2) \{V_2 = V_1 + u \& E_2 \\ = E_1 + (u, v) | v \in E_1\} \end{aligned}$$

- Удаление множества смежности вершины

$$\begin{aligned} G_1(V_1, E_1) - \Gamma^+(u) = G_2(V_2, E_2) \{V_2 = V_1 - u \& E_2 \\ = E_1 - (u, v) | v \in E_1\} \end{aligned}$$

#### 4. ЭКСПЕРИМЕНТЫ

Были проведены несколько серий вычислительных экспериментов. Для сравнения использовались изображения из баз Db1 и Db2 FVC2002 и FVC2003. Исследовались зависимости уровней ошибок первого и второго рода от количества ложных признаков, присутствующих в шаблоне. На рис. 2 приведены ROC-кривые для различных значений q, где q – количество ложных признаков в шаблоне.

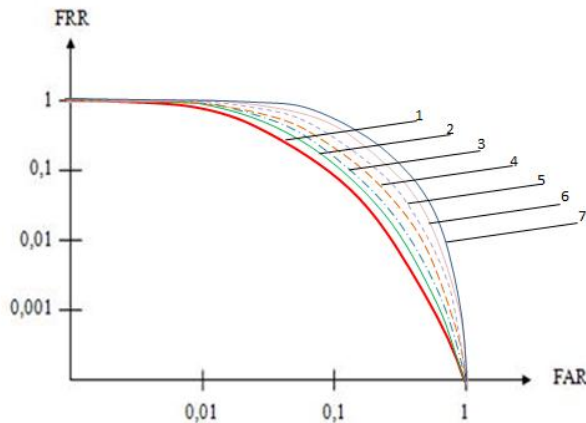


Рис 2: Кривые ошибок

Цифрами на рис. 2 обозначены:

- 1 – исходная ROC-кривая;
- 2 – ROC-кривая для q = 1;

- 3 – ROC-кривая для q = 3;
- 4 – ROC-кривая для q = 5;
- 5 – ROC-кривая для q = 7;
- 6 – ROC-кривая для q = 9;
- 7 – ROC-кривая для q = 11.

Очевидно, что при увеличении числа ложных признаков в шаблоне качество сравнения ухудшается.

#### 5. ЗАКЛЮЧЕНИЕ

Присутствие в шаблоне ложных признаков оказывает значительное влияние на точность идентификации. Чем больше таких признаков, тем выше вероятность ошибки. При этом отсутствие некоторых истинных частных признаков приводит к заметно более значительному увеличению вероятности ошибок идентификации, чем наличие ложных признаков в шаблоне.

#### 6. ССЫЛКИ

- [1] Ф.А. Новиков. Дискретная математика для программистов. 2-е издание – СПб.: Питер, 2013. – 432 с.
- [2] Пат. 2331108 Российская Федерация, МПК G06K 9/62. Способ сравнения отпечатков папиллярных узоров / Д.И. Аркабаев, В.Ю. Гудков. – № 2006140939/09; заявл. 20.11.2006; опубл. 10.08.2008; Бюл. № 22. – 17 с.
- [3] Maltoni D. Handbook of fingerprint recognition / D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar. – London: Springer-Verlag, 2009. – 496 p

#### Об авторах

Гудков Владимир Юльевич – доктор физико-математических наук, доцент Челябинского государственного университета. Его адрес: [diana@sonda.ru](mailto:diana@sonda.ru). Область научных интересов: обработка изображений, распознавание образов, идентификация отпечатков пальцев, комплексы программ.

Лепихова Дарья Николаевна – аспирант Челябинского государственного университета. Адрес [daria.lepikhova@yandex.ru](mailto:daria.lepikhova@yandex.ru). Область научных интересов: идентификация отпечатков пальцев, параллельные вычисления.

# Автоматическая настройка параметров алгоритма сегментации изображений Quick Shift

Максим Колосовский, Елена Крючкова  
Факультет информационных технологий

Алтайский государственный технический университет им. И.И. Ползунова, Барнаул, Россия  
maxim.astu@gmail.com kruchkova\_elena@mail.ru

## Аннотация

Качество работы алгоритмов пересегментации принципиально зависит от свойств изображения и требует настройки параметров под конкретное изображение. При использовании неподходящих параметров пересегментация не сохранит границы объектов изображения или построит слишком много мелких суперпикселей, что снизит компактность такого представления. В данной статье представлены исследования, направленные на выбор оптимальных параметров алгоритма сегментации Quick Shift. Для построения системы адаптации этого алгоритма, мы исследовали настройку отдельных параметров. Наш подход отличается от аналогичных тем, что основан на использовании метрик изображения, требующих принципиально меньше вычислительных затрат, чем многократное выполнение сегментаций, характерное для большинства других подходов. Полученные выводы позволяют перейти к построению единой системы настройки параметров Quick Shift.

**Ключевые слова:** пересегментация, семантическая сегментация, Quick Shift, настройка параметров.

## 1. ВВЕДЕНИЕ

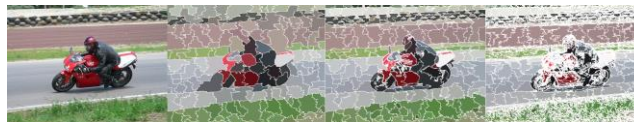
Одной из главных сложностей задачи семантической сегментации произвольного изображения является зависимость оптимальных значений параметров алгоритма от свойств изображений. Например, изображения, где объекты и фон хорошо различимы, и изображения, где объекты и фон похожи по цвету, требуют различных значений параметров, отвечающих за определение разницы цвета граничащих объектов. Другой пример – это размер объектов. Области одного и того же размера на одном изображении могут являться интересующими нас объектами, а на другом – деталями, которые не существенны для семантики изображения.

В данной работе мы на примере алгоритма Quick Shift [6, 7] представляем подход к настройке параметров алгоритма пересегментации. Для построения системы адаптации алгоритма мы исследуем настройку отдельных параметров при помощи вычисления сконструированных нами метрик, определяющих необходимые свойства изображения. Следующие три секции посвящены трем параметрам алгоритма Quick Shift: компактность цветов суперпикселей (секция 2), расстояние между пикселями в кластере (секция 3), ширина окна оценки плотности (секция 4).

В работе использовалась база изображений PASCAL VOC 2007 [1].

## 2. КОМПАКТНОСТЬ ЦВЕТОВ СУПЕРПИКСЕЛЕЙ

В рассматриваемой реализации алгоритма Quick Shift в качестве признаков пикселя используется не только значение цвета пикселя, но и его координаты на изображении. Благодаря этому кластеры в таком пространстве будут соответствовать группам пикселей, находящихся рядом на изображении. Без учета пространственного положения части одного кластера могут быть «разбросаны» по всему изображению, что порождает множество очень мелких суперпикселей и не соответствует цели пересегментации.



**Рис 1.** Исходное изображение и его пересегментации при различных значениях параметра цветовой компактности суперпикселей.

Данная реализация позволяет регулировать компактность цветов суперпикселя за счет параметра *ratio* в условии объединения пикселей в один кластер:

$$(da^2 + db^2 + dc^2) * ratio + dx^2 + dy^2 \leq maxdist^2 \quad (1)$$

где *da*, *db*, *dc* – расстояние между пикселями по компонентам цвета (их может быть и меньше трех), *ratio* – параметр компактности цветов суперпикселя, *dx*, *dy* – разность координат сравниваемых пикселей, *maxdist* – параметр, задающий максимальное расстояние между пикселями, при котором их можно объединить в один кластер. Чем больше значение параметра *ratio*, тем более похожие по цвету пиксели нужно объединять в один суперпиксель, тем, соответственно, более мелкие суперпиксели получатся и тем менее компактно будет представление изображения. С другой стороны, при слишком малом значении параметра *ratio* в один суперпиксель будут попадать не похожие по цвету пиксели, что, как и излишнее количество регионов, не отвечает целям сегментации изображения. Получается, что нужен компромисс в определении цветовой компактности суперпикселей (рисунок 1).

Основная проблема настройки этого параметра заключается в том, что компромиссное значение для разных изображений может быть очень различно (рисунок 2). Авторами были выявлены некоторые зависимости оптимального значения параметра *ratio* от свойств изображения [8]:

- чем более похожи цвета граничащих объектов, тем значение *ratio* должно быть больше;
- чем больше резко изменчивой текстуры (ветки деревьев, трава, провода, мелкие надписи и т.п.), тем больше при

увеличении параметра появляется очень мелких суперпикселей. Если все текстуры гладкие, то увеличение *ratio* слабо изменит сегментацию. Таким образом, отсутствие изменчивой текстуры позволяет задавать меньшее значение параметра;

- чем мельче размеры искомых объектов, тем *ratio* должно быть больше.



**Рис 2.** Разные изображения по разному сегментируются при увеличении параметра цветовой компактности

Эти критерии демонстрируют зависимости, но не позволяют автоматически вычислять оптимальное значение параметра для конкретного изображения. В этой работе мы попытались решить эту задачу. Мы нашли метрику изображения, которая взаимосвязана с оптимальным значением параметра *ratio*, после чего построили график этой зависимости метрики от значения параметра.

Было замечено, что оптимальное значение *ratio*, при котором не образуется очень много мелких суперпикселей, зависит от «изменчивости» текстуры. Например, если текстура представляет собой комбинацию пикселей очень разных цветов, то при большом значении *ratio* текстура «распадается» во множество мелких суперпикселей. Если же текстуры на изображении гладкие, то можно использовать гораздо большее значение *ratio* и получить более компактное представление изображения. Изменчивость текстуры *TextureContast* мы измеряем как среднюю разность по цвету между всеми парами соседних пикселей, имеющими общую сторону:

$$HorizContrast = \frac{1}{h * (w - 1)} \sum_{i=1}^h \sum_{j=1}^{w-1} dist(I_{i,j}, I_{i,j+1})$$

$$VertContrast = \frac{1}{(h - 1) * w} \sum_{j=1}^w \sum_{i=1}^{h-1} dist(I_{i,j}, I_{i+1,j})$$

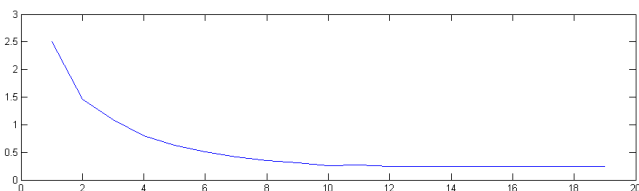
$$TextureContast = HorizContrast + VertContrast$$

где *h*, *w* – высота и ширина изображения в пикселях, *I<sub>ij</sub>* – цвет пикселя в позиции (*i*, *j*), *dist(a,b)* – расстояние между цветами *a* и *b* в пространстве Luv. На рисунке 2 представлены два изображения с наименьшей и наибольшими значениями метрик среди коллекции PASCAL VOC.

Эксперименты подтвердили наличие обратно пропорциональной зависимости оптимального значения параметра от значения предложенной метрики. Чтобы формализовать зависимость параметра от метрики, мы ввели определение «оптимальности» значения параметра, основанное на эмпирических наблюдениях. Оптимальным значением параметра *ratio* будем считать максимальное

значение, при котором суперпиксели размером менее 50 пикселей покрывает не более 2% площади изображения.

На изображениях коллекции PASCAL VOC 2007 выполнены пересегментации с различными значениями параметра *ratio* (от 0.1 до 2.5) и выбрано оптимальное из них. Для всех изображений также вычислена представленная метрика. Изображения сгруппированы по значению метрики. Для каждой группы изображений подсчитано среднее оптимального значения параметра. График, получившейся зависимости, представлен на рисунке 3.



**Рис 3.** Зависимость оптимального значения цветовой компактности от значения метрики

Таким образом, мы получили метод автоматической адаптации параметра цветовой компактности суперпикселя в алгоритме Quick Shift. Этот метод позволяет по представленному изображению оценить оптимальное значение параметра алгоритма. Заметим, что оценка производится без подгонки параметров или какого-либо предварительного обучения по прецедентам за исключением использования графика зависимости, представленного на рисунке 3.

### 3. РАССТОЯНИЕ МЕЖДУ ПИКСЕЛЯМИ В КЛАСТЕРЕ

В данном разделе продемонстрирован способ оценки оптимального значения параметра *maxdist* при помощи обучения на тестовых изображениях, имеющих эталонную семантическую сегментацию.

Алгоритм Quick Shift решает задачу сегментации как задачу кластеризации. В отличие от алгоритма кластеризации *k*-средних, требующего задания количества кластеров, Quick Shift требует максимального расстояния между элементами кластера. В этом разделе мы рассмотрим подход к оценке этого расстояния.

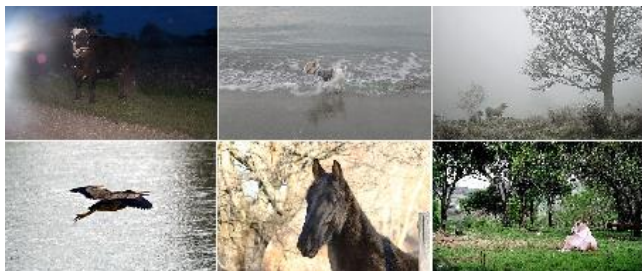
Quick Shift при кластеризации объединяет все пиксели изображения в одно дерево. В этом дереве родитель пикселя – это ближайший к нему в пространстве признаков (т.е. цвет + расположение) пиксель с большей плотностью. После этого удаляются все ребра больше некоторого порога – *maxdist*. Получившийся лес является результатом кластеризации пикселей, т.е. набором суперпикселей. Ключевое значение в кластеризации занимает выбор этого порога. Чем больше порог, тем более непохожие по цвету или дальше расположенные пиксели будут объединяться в один суперпиксель, тем, следовательно, выше вероятность объединения пикселей разных объектов. С другой стороны, если порог будет слишком мал, то суперпиксели будут чересчур мелкими. Цель подбора этого параметра – выбрать значение, при котором будет достаточно точно определяться граница между объектами при минимально возможном числе суперпикселей.

Как и в случаях с другими параметрами, оптимальное значение сильно зависит от содержания изображения. Если объект(ы) и фон имеют контрастирующие цвета на своих границах и граница не размыта, то порог можно увеличить. Если же цвета очень схожи, то придется пожертвовать компактностью пересегментации и уменьшить значение параметра *maxdist*. На рисунке 4 видно, что для одних изображений увеличение параметра *maxdist* незначительно ухудшает точность выделения границ, в то время как для других требуется очень мелкая сегментация из-за слабого контраста фона и объекта.



**Рис 4.** Влияние на различные изображения параметра максимального расстояния между пикселями

Оценка метрики происходит следующим образом. Метрика вычисляется по эталонной семантической сегментации. Напомним, что на эталоне пиксели отмечаются как принадлежащие к какому-либо объекту или фону. В коллекции PASCAL VOC также отмечаются граничные пиксели, семантическое значение которых не определено.



**Рис 5.** Несколько изображений с минимальным (верхний ряд) и максимальным (нижний ряд) значением метрики среди изображений коллекции PASCAL VOC 2007

Рассмотрим *i*-ую строку эталонной семантической сегментации. Будем сканировать эту строку слева направо. Нас интересуют переходы от одного объекта к другому. Пусть мы рассматриваем переход от пикселей объекта А к пикселям объекта В. Пусть крайний правый пиксель объекта А находится в столбце *j*, а крайний левый пиксель объекта В в столбце *k*.

$$Sem_{i,j} = Object A$$

$$Sem_{i,k} = Object B$$

$$Sem_{i,t} = BORDER, j < t < k$$

где  $Sem_{ij}$  – семантическая метка пикселя, расположенного на пересечении *i*-ой строки и *j*-го столбца, в частности, метка «BORDER» означает граничный пиксель.

Вычислим для этого перехода следующую величину:

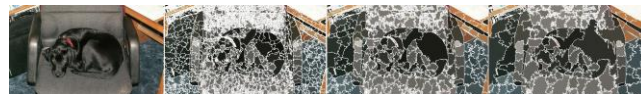
$$T = \sqrt{ratio * dist(I_{i,j}, I_{i,k})^2 + (k - j)^2},$$

где *ratio* – параметр цветовой компактности,  $I_{ij}$  – цвет пикселя,  $dist(C1, C2)$  – расстояние между цветами *C1* и *C2*.

Эта величина равняется расстоянию между пикселями (*i,j*) и (*i,k*) в используемом пространстве признаков (формула (1)).

Подсчитаем среднее значение этой величины по всем переходам во всех строках, а также подобные переходы при сканировании не только строк, но и столбцов. Отметим, что если группа граничных пикселей шириной более 10 пикселей, то такой переход не учитывается, т.е.  $j < k \leq j + 10$ .

На рисунке 5 представлены изображения, имеющие минимальное и максимальное значение метрики среди изображений коллекции PASCAL VOC 2007. На изображении с минимальным значением трудно рассмотреть объект, так как контраст объекта и фона очень небольшой. На изображении, где метрика максимальна, напротив, объект хорошо отличим от фона.



**Рис 6.** Пример пересегментации при значении параметра, равного половине значения рассматриваемой метрики (3-ий столбец), а также пересегментации при большем и меньшем значении параметра (2-ой и 4-ый столбцы)

Было отмечено, что оптимальное значение параметра достигается при значении половины значения метрики. Дело в том, что эта метрика вычисляет расстояние в пространстве признаков между пикселями, находящимися по разные стороны от границы, поэтому для разделения пикселей границы примерно посередине нужно брать порог, равный половине этой метрики. Примеры пересегментации со значением параметра, равным половине значения метрики, представлены на рисунке 6. Таким образом, мы получили инструмент для определения оптимального значения параметра *maxdist* по коллекции изображений, который позволяет оценить средний уровень контраста между объектами и фоном и использовать эту информацию при обработке новых изображений.

#### 4. ШИРИНА ОКНА ОЦЕНКИ ПЛОТНОСТИ

Кластеризация пикселей в алгоритме Quick Shift основывается на оценке плотности расположения пикселей в пространстве признаков: место где, плотность достигает локального максимума, считается центром кластера. Плотность точки можно оценить как количество точек, попавших в окрестность заданного радиуса (*окно Парзена*). Однако чтобы учитывать не только количество, но и расстояние до этих точек, вводят специальную функцию, называемую *ядром*, которая делает вклад более удаленных точек меньше. Обычно в качестве ядра используют Гауссово распределение.

Чем больше размер окна, тем функция плотности будет более гладкой, а значит, тем более крупные кластеры будут получаться. С другой стороны, слишком узкое окно построит функцию плотности со слишком большим количеством экстремумов. Напомним, что хорошая пересегментация – это компромисс между компактностью представления и точностью выделения контуров объектов.

На рисунке 7 представлено изображение и значения плотностей при радиусе окна, равного 1 и 12 пикселям. Плотности изображены в виде градаций серого, где белый цвет соответствует максимальному значению плотности, а

черный – минимальному. При узком окне значение плотности большинства пикселей ближе к минимальному, а при широком наоборот. Это связано с тем, что функция плотности представляет собой сумму функций ядер, а, как известно, ядро с узким окном – это резкий пик, а широкое ядро – очень широкий «колокол», где значение, близкое максимальному, распространяется далеко от центра. Таким образом, функция суммы таких «колоколов» имеет много значений близких к максимуму.



**Рис 7.** Исходное изображение и значение плотности при различных размерах окна, равных 1 (в центре) и 12 (справа)

В экспериментах радиус окна Парзена варьировался в пределах от 1 до 12 пикселей. Меньшее значение ширины окна бессмысленно, так как в окно будет попадать лишь сам пиксель, а окрестность не будет учитываться. Ограничение сверху объясняется существенным увеличением вычислительных затрат при увеличении ширины окна. Окно радиусом  $r$  пикселей способно строить отдельные суперпиксели для деталей изображения размером  $2r$  и более. Поэтому ширина окна, как правило, гораздо меньше размера объектов, а значит, оптимальная ширина окна зависит не от размера объектов, а от частей объекта, которые хотелось бы выделять как отдельные суперпиксели. Кроме того, на точность выделения границ влияет то, как расположены области внутри объекта. Например, если на границе объекта есть область небольшого размера, похожая по цвету на фон, то более широкое окно не выделит для этой области отдельного суперпикселя, а присоединит ее к фону. Таким образом, для определения оптимальной ширины окна будет недостаточно даже эталонной семантической сегментации, т.к. ошибки в выделении контуров объектов зависят от конфигурации отдельных частей объектов.

В общем случае увеличение ширины окна должно улучшать сегментацию, потому как учитывается более широкая окрестность пикселя, а пропуск мелких деталей является негативным последствием объединения цветового и пространственного расстояния между пикселями (формула (1)). Например, возможно объединение в один кластер далеких по цвету, но близко расположенных пикселей, вследствие чего в один суперпиксель могут попадать пиксели очень разных цветов. Другим часто наблюдаемым недостатком алгоритма Quick Shift являются ошибки на размытых границах объектов. Если даже цвета объектов существенно различаются, то размытый переход может объединить части этих объектов в один суперпиксель, так как расстояние между соседними пикселями в этом переходе не слишком большое.

Таким образом, этот параметр нуждается скорее не автоматической настройке, а в модификации алгоритма Quick Shift, исправляющей дефекты, связанные с неподходящим значением этого параметра.

## 5. ОБЗОР ПОДОБНЫХ РАБОТ

Автоматическая настройка параметров алгоритмов сегментации является активно исследуемой областью в настоящее время. В ней применяется крайне широкий спектр принципиально различных подходов, среди которых сравнение нескольких сегментаций одного изображения, построенных с разными параметрами [2], многократный прогон алгоритма сегментации с настройкой параметров по предыдущему прогону [5], сложная калибровка одних параметров при фиксировании других [3].

Несмотря на такое большое разнообразие применяемых подходов, наш алгоритм отличается тем, что мы настраиваем его при помощи вычисления метрик изображения, которые требуют принципиально меньших вычислительных затрат, чем построение нескольких сегментаций. Кроме того, мы ставим цель настраивать сразу все параметры алгоритма, а не только ширину окна, что характерно для работ по настройке алгоритмов на основе оценки плотности [4].

## 6. ЗАКЛЮЧЕНИЕ

В данной статье мы исследовали настройку отдельных параметров алгоритма сегментации изображений на примере алгоритма Quick Shift. Полученные выводы по настройке отдельных параметров позволяют перейти к построению единой системы настройки параметров Quick Shift.

## 7. ССЫЛКИ

- [1] Everingham M., van Gool L., Williams C., Winn J., Zisserman. A. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results // International Journal of Computer Vision (IJCV). June, 2010. Vol. 88, No. 2. P. 303-338.
- [2] Franek L., Jiang X. Adaptive Parameter Selection for Image Segmentation Based on Similarity Estimation of Multiple Segmenters. // 10th Asian Conference on Computer Vision, Queenstown, New Zealand, November 8-12, 2010. pp 697-708.
- [3] Franek L., Jiang X. Alternating Scheme for Supervised Parameter Learning with Application to Image Segmentation. // 14th International Conference, CAIP 2011, Seville, Spain, August 29-31, 2011, pp 118-125.
- [4] Gilani S., Rao N. Data Driven Bandwidth for Medoid Shift Algorithm // International Conference, Santander, Spain, June 20-23, 2011. pp 534-546.
- [5] Khan A., Mikut R., Schweitzer B., Weiss C., Reischl M. Automatic Tuning of Image Segmentation Parameters by Means of Fuzzy Feature Evaluation // Synergies of Soft Computing and Statistics for Intelligent Data Analysis. Vol 109. pp 459-467.
- [6] Vedaldi A., Soatto S. Quick Shift and Kernel Methods for Mode Seeking // European Conference on Computer Vision (ECCV). Marseille, France. 12-18 October, 2008. pp. 705-718.
- [7] Vedaldi A., Fulkerson B. VLFeat: An open and portable library of computer vision algorithms // 18th Annual ACM International Conference on Multimedia (ACM MM). Firenze, Italy. 25-29 October, 2010. P. 1469-1472.
- [8] Колосовский М., Крючкова Е. Настройка параметров алгоритма сегментации изображений QuickShift // Программная инженерия. №5, 2013, сс. 11-20.

## **Об авторах**

Максим Колосовский – аспирант ФИТ АлтГТУ им. И.И. Ползунова. Его адрес: [maxim.astu@gmail.com](mailto:maxim.astu@gmail.com)

Елена Крючкова – профессор ФИТ АлтГТУ им. И.И. Ползунова. Ее адрес: [kruchkova\\_elena@mail.ru](mailto:kruchkova_elena@mail.ru).



# Визуализация поляризации света в сценах с кристаллами

Д.С. Козлов

Новосибирский государственный университет, Новосибирск, Россия

kozlov@oapmg.sccc.ru

## Аннотация

В последнее время всё больше работ по фотореалистическому рендерингу обращают внимание на поляризацию света. Действительно, рендеринг поляризованным лучом позволяет получать изображения, которые могут значительно отличаться от изображений, полученных на основе рендеринга неполяризованным лучом. Рендеринг кристаллов вообще невозможен без учёта поляризации.

Свет бывает поляризованный, неполяризованный и частично поляризованный. При этом он бывает поляризован линейно, циркулярно или эллиптически. В научной визуализации часто представляют различные объекты реального мира и их параметры при помощи изображений. В данной работе мы задались целью графически изобразить степень поляризации лучей в сцене и её характеристики при помощи изображений, отличающихся от традиционной визуализации интенсивности лучей.

По нашему мнению, такое средство может быть востребовано само по себе, а также поможет отлаживать алгоритмы рендеринга поляризованным лучом, используя широко известные факты из оптики и простые сцены со сложными оптическими материалами.

**Ключевые слова:** Фотореалистический рендеринг, кристаллы, двойное лучепреломление, поляризация света, дисперсия света, визуализация поляризации.

## 1. ВВЕДЕНИЕ

В классических алгоритмах лучевой трассировки такое свойство лучей как поляризация вообще не рассматривается и не учитывается. Все вычисления обычно опираются на выкладки геометрической оптики, а волновые параметры луча (поляризация, фаза волны) остаются за кадром. Это объясняется тем, что в сценах с диффузными объектами распространяющийся свет не поляризован, а интерференция света вообще наблюдается только крайне в специализированных сценах, поэтому вычислять и учитывать волновые свойства луча дорого, сложно и нет особого смысла.

Волновые свойства света начинают проявлять себя, если в сцену добавить полупрозрачные объекты или металлы. В этих случаях свет изменяет своё состояние поляризации при отражении от них и прохождении через них. Изменение поляризации света, однако, в общем случае падения луча на поверхность таких объектов оказывается не настолько значительным, чтобы его нельзя было проигнорировать. Стоит отметить, что разница между изображениями, рассчитанными с учётом поляризации и без такового, всё же наблюдаема [6].

Если в сцену поместить анизотропный кристалл (оптические свойства которого зависят от направления), то ситуация кардинально меняется. Дело в том, что такие кристаллы являются естественными поляризаторами, поэтому свет, проходящий через них, оказывается линейно поляризованным. Отсутствие осевой симметрии электрического поля луча при такой поляризации приводит к тому, что взаимодействие поляризо-

ванного света с объектами очень сильно зависит от свойств этих объектов. В частности, при попадании на другой поляризатор свет может быть полностью погашен, или может отсутствовать отражённый луч при падении света на полупрозрачный изотропный объект. Таким образом, сцены, содержащие анизотропные объекты вообще не могут быть корректно рассчитаны без учёта поляризации света.

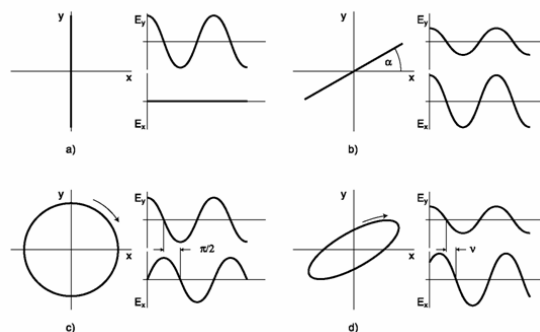
В целом причины появления поляризованного света в сцене следующие<sup>1</sup>: 1) излучение поляризованного света источником; 2) анизотропия среды распространения света; 3) преломление и отражение на границе двух сред.

В данной статье мы подробно остановимся на изучении и визуализации степени присутствия поляризованных лучей в сцене и их поляризационных характеристиках в сценах со всеми перечисленными «источниками» поляризации.

Отметим так же, что при преломлении на границе полупрозрачных сред может наблюдаться скачкообразный сдвиг фазы [7], что может влиять на интерференцию в сцене, но в данной статье это не рассматривается.

## 2. ПОЛЯРИЗАЦИИ СВЕТА

Поляризация луча света – это характеристика, описывающая поведение вектора электрического поля в плоскости перпендикулярной направлению распространения луча. Поляризация может быть линейной, круговой или в общем случае эллиптической.



**Рис. 1.** а) линейная поляризация вдоль оси ординат, б) общий случай линейной поляризации, в) круговая (циркулярная) поляризация, д) эллиптическая поляризация. Изображение взято из работы [2].

На рис. 1 показано траектории описываемые концом вектора электрического поля с началом на траектории распространения поляризованного луча в плоскости перпендикулярной лучу, в зависимости от типа поляризации луча.

Каждая электромагнитная волна в отдельности полностью поляризована. Квазимонохромный луч света, как совокупность волн, распространяющихся по одной прямой и имеющих одну длину волны, может быть полностью поляризован,

<sup>1</sup> [http://ru.wikipedia.org/wiki/Поляризация\\_волн](http://ru.wikipedia.org/wiki/Поляризация_волн)

частично поляризован или не поляризован вообще, в зависимости от степени корреляции колебаний отдельных волн в луче. Если колебания не коррелируют – луч не поляризован, если степень корреляции максимальная – луч полностью поляризован и имеет один из трёх перечисленных типов поляризации.

Для описания и расчёта состояния поляризации существует несколько математических конструкций (подходов): вектора Джонса и матрицы Джонса, вектора Стокса и матрицы Мюллера, матрицы когерентности и их модификаторы, см. подробнее [5]. Конструкция Джонса подходит лишь для описания полностью поляризованного луча, тогда как остальные два позволяют описать и частично поляризованный луч. Для достижения целей данной работы (оценка степени поляризации луча) удобно использовать подход основанный на матрицах когерентности (они содержат достаточный минимум информации для этого).

Рассмотрим этот подход детальнее. Электрическое поле полностью поляризованного луча может быть описано комплексным вектором Джонса:

$$\vec{E} = \begin{bmatrix} \tilde{E}_x \\ \tilde{E}_y \end{bmatrix} = \begin{bmatrix} E_x e^{i(\omega t + \delta_x)} \\ E_y e^{i(\omega t + \delta_y)} \end{bmatrix},$$

здесь  $\omega$  – частота колебаний,  $\delta_x$  и  $\delta_y$  – начальные фазы колебаний,  $E_x$  и  $E_y$  – амплитуды проекций колебаний электрического поля на два перпендикулярных наперёд выбранных вектора перпендикулярных волновому вектору. Реальным значением вектора электрического поля будет  $\text{Re } \vec{E}$ . Тогда матрицей когерентности называется следующая матрица [4, 5]:

$$J = \langle \vec{E} \times \vec{E}^\dagger \rangle = \begin{bmatrix} \langle E_x E_x^* \rangle & \langle E_x E_y^* \rangle \\ \langle E_y E_x^* \rangle & \langle E_y E_y^* \rangle \end{bmatrix} = \begin{bmatrix} J_{xx} & J_{xy} \\ J_{yx} & J_{yy} \end{bmatrix}. \quad (1)$$

Матрица и базис (два вектора перпендикулярные лучу), для которого она вычислена или измерена, определяют состояние поляризации луча. Элементы главной диагонали матрицы вещественные и представляют собой интенсивности  $x$ - и  $y$ -компонент электрического вектора, таким образом, интенсивность луча равна следу матрицы когерентности:

$$I = \text{Tr } J = J_{xx} + J_{yy} = \langle E_x E_x^* \rangle + \langle E_y E_y^* \rangle. \quad (2)$$

Внедиагональные элементы определяют степень корреляции компонент электрического поля. Справедливы следующие утверждения: 1) для естественного (полностью неполяризованного) света внедиагональные элементы матрицы когерентности равны 0, а  $J_{xx} = J_{yy}$ ; 2) для полностью поляризованного света детерминант матрицы когерентности равен 0; 3) матрица когерентности лучей распространяющихся по одной прямой равна сумме их матриц когерентности.

Отметим, что частично поляризованный луч можно рассматривать как сумму двух лучей: полностью поляризованного и полностью неполяризованного. Отношение интенсивности полностью поляризованной части к полной интенсивности луча называется степенью поляризации луча. Она определяется соотношением [5]:

$$P = \frac{I_{\text{поляр}}}{I_{\text{полн}}} = \sqrt{1 - \frac{4 J_{xx} J_{yy} - J_{xy} J_{yx}}{(J_{xx} + J_{yy})^2}}, \quad (3)$$

то есть  $P$  равно 1 для полностью поляризованного луча и – 0 для естественного света.

Имеет смысл также оценить форму поляризации полностью поляризованной компоненты. Введём понятие степени эллиптичности полностью поляризованной компоненты луча, как соотношение малой полуоси к большой полуоси эллипса, описываемого концом вектора электрического поля. В таком случае циркулярно поляризованный свет будет иметь максимальную степень эллиптичности равную 1, а линейно поляризованный минимальную – 0. Численно степень эллиптичности может быть вычислена как [8]:

$$E = \left| \tan\left(\frac{1}{2} \arcsin(\sin(2 \arctg(\sqrt{J_{yy}} / \sqrt{J_{xx}})) \sin \gamma)\right) \right|, \quad (4)$$

где  $\gamma$  – азимут эллипса, см. далее.

Другим важным параметром формы эллипса (в том числе, и вырожденного в линию для линейной поляризации) является азимут, то есть угол между большой полуосью эллипса и осью абсцисс:

$$\gamma = \arcsin(\text{Im } J_{yx} / \sqrt{J_{xx} J_{yy}}). \quad (5)$$

Величина азимута изменяется в интервале  $[-\pi / 2; \pi / 2]$ .

Нижнее и верхнее граничное значение достигаются, например, при правой и левой круговой поляризации, соответственно.

Последняя важная характеристика эллипса – это его размер, но он прямо пропорционален интенсивности луча, поэтому оценивать его отдельно не имеет смысла.

### 3. МЕТОД ВИЗУАЛИЗАЦИИ

Идея визуализации распределения поляризованного света в сцене состоит в том, чтобы модифицировать классическую камеру, используемую в алгоритмах лучевой трассировки, заставив её «фотографировать» требуемые параметры входящих лучей, вместо их интенсивности, как это делает стандартная камера.

В данном случае модификация состоит в следующем. Каждый входящий в камеру луч, содержит информацию о поляризации луча, т.е. матрицу когерентности и базис в котором она определена. Для того чтобы построить изображение, необходимо по данной информации вычислить значение визуализируемого параметра, а затем преобразовать его в значение интенсивности пикселя изображения, которое будет записано в спектральное изображение, собираемое камерой.

Алгоритм такого преобразования зависит от визуализируемого параметра, но общая идея состоит в том, чтобы получить такие значения интенсивностей представляющие собой доли от какого-либо стандартного спектра. Преимущество такого подхода состоит в том, что получаемое изображение может быть преобразовано в RGB-изображение путём стандартного преобразования CIE [3]. Полученное изображение будем называть *картой* того или иного параметра, например, *карта поляризации* света в сцене.

**Для степени поляризации света.** Параметр  $P$  изменяется от 0 до 1. Для получения интенсивности его необходимо просто умножить на спектр стандартного источника света CIE D65.

$$P(\lambda) = \text{CIED65}(\lambda) * P,$$

То есть, для полностью поляризованного света в результате мы получим спектр CIE D65, т.е. белый, а для полностью не поляризованного получим нулевой спектр, т.е. чёрный цвет.

Для степени эллиптичности. Параметр  $E$  изменяется от 0 до 1. Дополнительно будет считать, что эллиптичность неполяризованного света равна 0.

$$E(\lambda) = CIED65(\lambda) * E$$

Отметим, что имеет смысл рассчитывать только монохромные изображения, поскольку если рассчитать спектр для нескольких длин волн, то при пересчёте его в RGB-цвет стандартным преобразованием CIE полученная цветовая информация окажется бессмысленной, так как преобразование CIE рассчитано только на цветовую информацию в компонентах спектра.

#### 4. ЭКСПЕРИМЕНТЫ

Для проведения численных экспериментов использовалась виртуальная сцена, состоящая из одного монокристалла и расположенным под ним квадратным источником света. Спектр источника, определяется RGB изображением, преобразованным в спектральный вид по методу, предложенному в [3], в качестве белого спектра использовался CIE D65. Для расчётов использовалось представление луча и модель взаимодействия света с кристаллами из работы [1].

Проведены, три серии экспериментов, которые демонстрируют распределение поляризованного света в сценах с различными причинами его возникновения: 1) отражение и преломление на гранях изотропного прозрачного объекта; 2) отражение и преломление света на естественных поляризаторах – анизотропных кристаллах; 3) освещение сцены источником поляризованного света.

##### 4.1 Изотропный объект

Монокристалл, представляет собой кусок прозрачного стекла, ограниченного одной из стандартных огранок драгоценных камней, будет называть его стеклянным кристаллом (хотя стекло и не имеет кристаллической решётки). Одна и та же сцена рассчитана в четырёх различных режимах. На рис. 2 приведено полноцветное изображение, рассчитанное на 21 длине волны, распределённых равномерно на видимом диапазоне волн от 380 от 780 нм, на рис. 3 – монохромное изображение (580 нм).

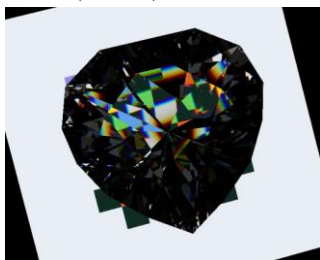


Рис. 2. Полноцветное изображение

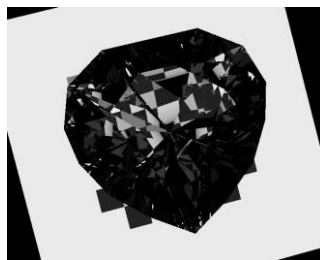


Рис. 3. Монохромное изображение (580 нм)

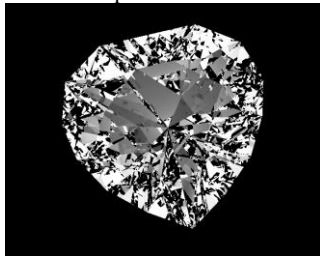


Рис. 4. Карта поляризации (580 нм)

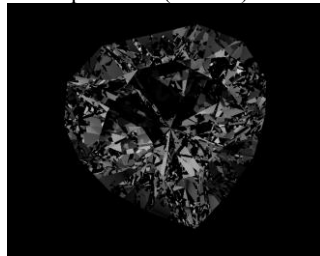


Рис. 5. Карта эллиптичности (580 нм)

Рис. 4 и 5 содержат карту поляризации и эллиптичности, соответственно. Глубина трассировки для всех приведённых изображений равна 13.

Как можно видеть на изображениях источник света излучает не поляризованный свет (источник света чёрный на карте поляризации), который при отражении частично линейно поляризуется (см. центр рис. 4), а при прохождении через кристалл поляризация может становиться эллиптической (рис. 5). Эллиптическая поляризация обычно возникает при полном внутреннем отражении или при слиянии частично линейно поляризованных лучей.

Спектр пикселей монохромных на изображениях на рис. 3, 7, 12 рассчитывается как отношение рассчитанного значения в пикселе к значению CIE D65 для той же длины волны, умноженный на спектр CIE D65, в результате изображения выглядят серыми, а не окрашенными цветом, соответствующим длине волны.

##### 4.2 Анизотропный объект

Ситуация с эллиптичностью и степенью поляризации света в сцене кардинально меняется если, заменить стеклянный кристалл на кристалл вырезанный из кальцита (анизотропный), оставляя источник света неполяризованным, результат показан на рис. 6–9. На рис. 9 видно, что практически весь прошедший через кристалл свет стал линейно поляризованным. А эллиптически поляризованный свет в сцене отсутствует вообще, поскольку прозрачные анизотропные кристаллы являются естественными поляризаторами (в них может распространяться только линейно поляризованный свет).

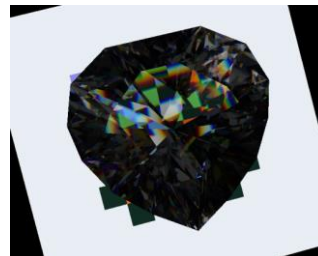


Рис. 6. Цветное изображение



Рис. 7. Монохромное изображение (580 нм)

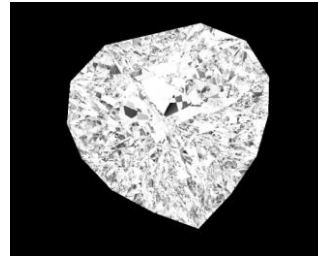


Рис. 8. Карта поляризации (580 нм)

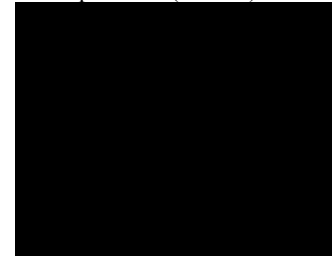


Рис. 9. Карта эллиптичности (580 нм)

##### 4.3 Поляризованный источник света

Рассмотрим, что произойдёт, если заменить источник света линейно поляризованным, то есть, как будто бы кристалл разместили на фоне ЖК-монитора. Для этого над источником размещена виртуальная пластинка кальцита, через которую пропускаются испущенные источником лучи. Внутри пластинки алгоритмически оставляется только один из двух образованных при преломлении на её границе лучей. Оптическая ось пластинки выбрана так, что в результате, плоскость поляризации выходящих из пластины лучей расположена по диагонали текстуры.

Результат для сцены со стеклянным кристаллом показан на рис. 10–13. Монохромное и цветное изображения практически не изменились, на карте поляризации источник стал белым, что говорит о его полной поляризации, а эллиптичность всё равно осталась (хотя и меньше, чем на рис. 5), хотя освещение изначально было линейно поляризованным.

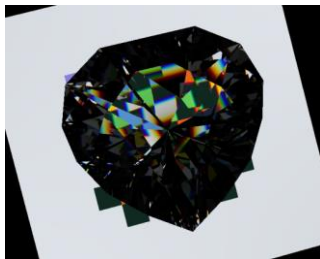


Рис. 10. Цветное изображение

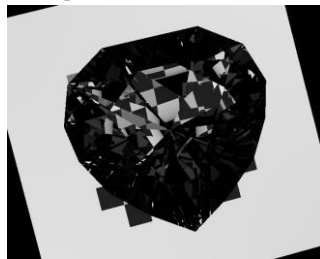


Рис. 11. Монохромное изображение (580 нм)



Рис. 12. Карта поляризации (580 нм)

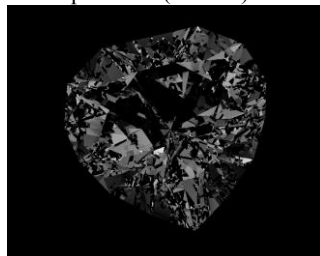


Рис. 13. Карта эллиптичности (580 нм)

Если в такой сцене заменить стеклянный кристалл, на анизотропный кристалл кальцита то результат будет следующий:

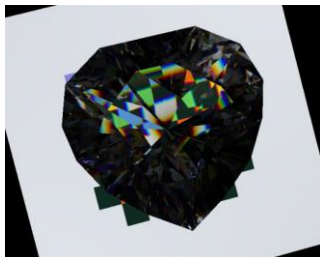


Рис. 14. Цветное изображение



Рис. 15. Монохромное изображение (580 нм)

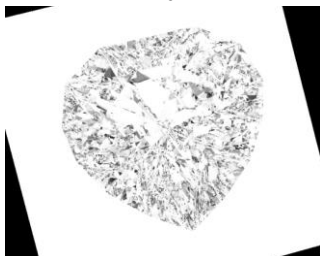


Рис. 16. Карта поляризации (580 нм)

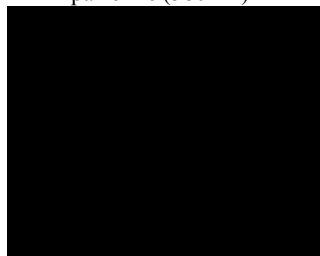


Рис. 17. Карта эллиптичности (580 нм)

Почти весь свет оказывается полностью поляризованным. Тёмные пятна на фоне кристалла могут быть объяснены тем, что часть трассирующих лучей не достигают источника из-за ограничения глубины трассировки.

## 5. ЗАКЛЮЧЕНИЕ

В работе предложен подход к визуализации распределения поляризованного света в сцене. Распределение рассматривается по таким параметрам, как степень поляризации и степень эллиптичности. Камера в алгоритме лучевой трассировки

модифицирована и «фотографирует» вместо традиционного параметра луча (интенсивности), значение выбранного параметра поляризации луча, а затем собирает изображение, показывающее распределение значений данного параметра в свете идущим из сцены.

Предложенный подход показал хорошую наглядность и позволяет «оценить» количество поляризованного света в сцене, его распределение и его характеристики.

## 6. БЛАГОДАРНОСТИ

Выражаю благодарность моему научному руководителю Дебелову Виктору Алексеевичу за постановку задачи и постоянное внимание к данной работе.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 12-07-00391 а.

## 7. БИБЛИОГРАФИЯ

- [1] Debelov V.A., Kozlov D.S. *A Local Model of Light Interaction with Transparent Crystalline Media*. // *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 8, pp. 1274-1287, Aug. 2013, doi:10.1109/TVCG.2012.304.
- [2] Devlin K., Chalmers A., Wilkie A., Purgathofer W. *STAR: Tone Reproduction and Physically Based Spectral Rendering*. *State of the Art Reports, Eurographics 2002*, pp. 101–123.
- [3] Glassner A.S. *Principles of digital image synthesis*. // *Morgan Kaufmann Publishers, Inc, San Francisco, California, 1994*.
- [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] Борн М., Вольф Э. *Основы Оптики*. Москва, “Наука”, 1973.
- [6] 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, Protvino-Altai*. - Изд-во ИФТИ, Протвино. - С. 15-22.
- [7] Фёдоров Ф.И., Филиппов В.В. *Отражение и преломление света прозрачными кристаллами*. Минск, “Наука и техника”, 1976.
- [8] Шерклифф У. *Поляризованный свет. Получение и использование*. Москва, “Мир”, 1965.

## Об авторах

Дмитрий Сергеевич Козлов – аспирант Новосибирского государственного университета.

E-mail: kozlov@oapmg.sccc.ru.

# Двухуровневый параллелизм в задачах обработки и визуализации синоптических данных

Черкашин Александр  
Лаборатория машинной графики

Федеральное государственное бюджетное учреждение науки

Институт автоматизации и процессов управления Дальневосточного отделения Российской академии наук, Владивосток,  
Россия

alexandr@cherkashin.ws

## Аннотация

В этой статье описывается методология применения гибридных архитектур массового параллелизма для обработки и визуализации синоптических данных.

Приводятся примеры конкретных реализаций алгоритмов обработки данных на различных вычислительных архитектурах, в том числе кластерных, в контексте программного комплекса для визуализации синоптических данных.

Оценивается эффективность и производительности каждой архитектуры на основе результатов обработки тестового набора данных.

**Ключевые слова:** CUDA, GPU, CPU, гибридные параллельные вычисления, кластер, синоптика, данные

## 1. ВВЕДЕНИЕ

С появлением технологий массового параллелизма удалось ощутимо увеличить производительность обработки данных, требующих больших вычислительных ресурсов. Формат синоптических данных – это, как правило, двумерная или трехмерная сетка. Данные такого типа эффективно обрабатываются системами массового параллелизма, где операция над каждой ячейкой может быть выполнена отдельным независимым потоком.

Разрешение сетки данных является определяющим фактором при учете производительности. Большие вычислительные кластеры хорошо справляются с такой задачей, однако, на определённом уровне разрешения сетки ощущается дефицит доступных ресурсов. На сегодняшний день GPU является недорогой высокопроизводительной альтернативой обычным микропроцессорам. Скорость роста производительности GPU намного выше CPU. Графические процессоры, как и их предшественники – системы SIMD, предназначены для использования как системы массивного мелкозернистого параллелизма [1].

GPU имеют большую, выделенную, память для чтения-записи и только чтения, чтобы обеспечить высокую пропускную способность, необходимую для сложных вычислений с плавающей точкой (такие данные являются характерными для синоптики).

На сегодняшний день, исследования в области ускорения процесса обработки синоптических данных ведутся в

направлении перспективных технологий массового параллелизма, таких как CUDA, MPI и OpenMP [2-5]. Однако само по себе применение этих технологий по отдельности, не всегда может быть целесообразно. Так как эти технологии обладают разным уровнем зернистости параллелизма, то в зависимости от типа данных и способа их обработки необходимо применять гибридные (комбинированные) методы параллелизма. Например, разделить большую географическую область синоптических данных, на подобласти между узлами кластера (рис. 1). В случае необходимости детализации какой-либо подобласти, для расчета может быть использован более мелкозернистый параллелизм, например, параллелизм GPU.

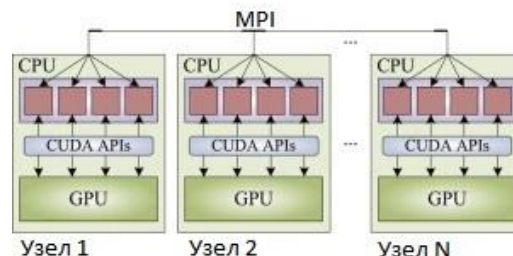


Рис 1: Схема кластера MPI+OpenMP+CUDA.

Таким образом, задача обработки синоптических данных может быть эффективно решена при помощи систем гибридного параллелизма.

## 2. СИСТЕМА ОБРАБОТКИ И ВИЗУАЛИЗАЦИИ СИНОПТИЧЕСКИХ ДАННЫХ

В рамках работы посвященной вопросам обработки и визуализации синоптических данных, лабораторией машинной графики ИАПУ ДВО РАН была создана модульная программная система обработки и визуализации синоптических данных [6-7].

Система представляет собой модульный программный комплекс. Основной частью комплекса является графическая программная оболочка, с помощью которой пользователь может управлять данными, а именно: загрузкой, обработкой, процессом и способом визуализации. Также имеется ряд программных инструментов для проведения визуального анализа данных и управлением хронологией динамических данных.

Пользовательский интерфейс построен по принципу минимализма. Все неинформативные для пользователя процессы (в основном это алгоритмически-функциональная

часть) – скрыты. Пользователь видит лишь результат визуализации и инструменты для его настройки.

С целью обеспечить возможность добавления новых типов данных, методов обработки и визуализации, все процедуры работы с данными (загрузка, обработка, визуализация) вынесены во внешние модули, имеющие унифицированный интерфейс, спецификацию взаимодействия и потоков данных. Таким образом, чтобы адаптировать всю систему для работы с новым типом данных достаточно создать соответствующие модули.

Все данные проходят иерархическую процедуру обработки, состоящую из трех этапов, соответственно модулям:

1. Загрузка данных – этап, на котором учитывается специфика формата, при этом структура данных приводятся к универсальной программной структуре, разработанной для системы.
2. Обработка данных – необязательный этап изменения исходных данных в соответствии с заданными алгоритмами. Применяется, как правило, для выделения областей интереса, расчета дополнительных атмосферных характеристик, фильтрации и других операций.
3. Визуализация – этап отображения данных на экран. На этом этапе пользователь управляет ракурсом виртуальной камеры, задает параметры визуализации.

Система способна одновременно работать с несколькими наборами статических и динамических данных, а также одновременно обрабатывать и визуализировать один и тот же набор данных разными методами. Важной особенностью является то, что для всех данных соблюдается временная хронология. В выбранный пользователем момент времени обрабатываются только данные, с соответствующей датой. Для выбора интересующего момента времени используется программный инструмент «временная шкала».

## 2.1 Модуль двухуровневой обработки данных

Этап обработки синоптических данных является наиболее трудоемким, так как для его осуществления обычно требуется выполнение операций над множеством (обращение к каждому элементу области сетки и/или обращение к элементам подобластей всей сетки данных). Конкретная сложность задачи определяется алгоритмом обработки.

Одной из трудоемких операций по выявлению областей интереса является процедура поиска температурных аномалий тропического циклона (аномалией считается отклонение от нормального значения) [8]. Анализ аномалий в пространстве и времени позволяет метеорологам судить о физике процесса. В качестве исходных данных для этой задачи выступает трёхмерное скалярное поле распределения температуры. Алгоритм обработки данных достаточно прост: для каждого вертикального уровня сетки данных  $l$  рассчитывается среднее арифметическое значение  $Avg(l)$  по формуле 1:

$$Avg(l) = \frac{\sum_x \sum_y (W_{Rr}(x, y) \cdot f(x, y, l))}{\sum_x \sum_y W_{Rr}(x, y)}, \quad (1)$$

где  $f(x, y, l)$  — значение сетки данных;  $x, y$  — размерность по  $x$  и  $y$  соответственно;

$$W_{Rr}(x, y) = \begin{cases} 1, & r < \sqrt{x^2 + y^2} < R \\ 0 & \end{cases}, \quad (2)$$

$W_{Rr}$  — это весовой коэффициент, определяющий принадлежность элемента кольца «спокойствия», где  $r$  — внутренний радиус кольца, а  $R$  — внешний.

Затем рассчитывается стека аномалий по формуле 3:

$$Anm(l) = f(l) - Avg(l), \quad (3)$$

где  $l$  — текущий вертикальный уровень;  $f(l)$  — исходная сетка данных,  $Avg(l)$  — среднее значение.

Однако, в зависимости от разрешения исходной сетки, задача может быть достаточно трудоемкой. Так же стоит учесть, что в случае визуализации в реальном времени, требования к производительности значительно возрастут.

Для увеличения скорости обработки данных был разработан специальный модуль, способный выполнять вычисления на удаленном кластере, поддерживающем технологию гибридного параллелизма. Существуют различные аппаратные конфигурации кластеров, но для работы модуля, кластер должен поддерживать как минимум два уровня распараллеливания:

- Параллелизм между множеством узлов CPU.
- Параллелизм внутри многоядерной структуры GPU.

Применительно к задаче поиска аномалий тропического циклона, распределение аппаратных ресурсов было выполнено следующим образом: трехмерная сетка данных, загруженная при помощи системы визуализации, передается на кластер, где она делится по вертикальным уровням между потоками MPI кластера. Каждый поток рассчитывает  $Avg(l)$ . Результат и сама сетка передаются в GPU, где вычисляется  $Anm(l)$ . Данные каждого уровня  $l$  собираются в единую стеку и возвращаются в систему визуализации (рис. 2).

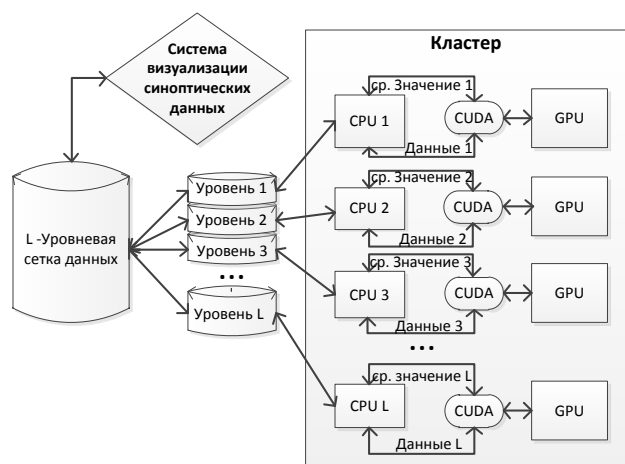


Рис 2: Схема гибридной обработки данных кластером CPU+GPU.

### 3. ОЦЕНКА ЭФФЕКТИВНОСТИ

Чтобы определить эффективность предложенного метода гибридной обработки синоптических данных был проведен ряд численных сравнительных экспериментов различных параллельных архитектур, показывающий временные затраты на обработку данных и накладные расходы. Ниже представлены таблицы поэтапных временных задержек для 4-х реализаций алгоритма поиска аномалий:

Размер решетки	Время выполнения
128x128x64	0.044
256x256x64	0.196
512x512x64	0.737
1024x1024x64	3.079

Таб. 1 Реализация для CPU. Задействован 1 Intel Core i5 центральный процессор. Время в секундах.

Размер решетки	Время выполнения
128x128x64	0.798
256x256x64	0.856
512x512x64	0.968
1024x1024x64	1.379

Таб. 2 реализация для CPU+CUDA. Задействован 1 Intel Core i5 центральный процессор и 1 GPU GeForce GT 540M. Время в секундах.

Размер решетки	Отправка/Получение	Обработка	Суммарно
128x128x64	1.24	1.18	2.42
256x256x64	5.08	1.25	6,33
512x512x64	19.73	1.38	21,11
1024x1024x64	68.24	1.52	69,76

Таб. 3 Реализация для MPI. Задействовано 8 физических узлов, по 8 параллельных потоков на каждом. Время в секундах.

Размер решетки	Отправка/Получение	Обработка	Суммарно
128x128x64	1.31	1.22	2,53
256x256x64	5.15	1.23	6,38
512x512x64	20.12	1.34	21,46
1024x1024x64	69.99	1.46	71,45

Таб. 4 Реализация MPI+ CUDA. Задействовано 8 физических узлов, по 8 параллельных потоков на каждом, по 2 GPU на каждый узел. Время в секундах.

Как можно заметить из первых двух таблиц, использование GPU эффективно только на достаточно больших размерах решетки (от 1024x1024x64). На самой большой решетке из теста удалось получить прирост скорости выполнения в 2.23 раза. В тоже время использование кластерных технологий в рамках предложенных архитектур (Таб. 3, Таб. 4) привело к значительным накладным расходам. Основной проблемой стала передача больших объемов данных между кластером и компьютером.

### 4. ЗАКЛЮЧЕНИЕ

В работе предложен подход высокопроизводительной обработки сеточных синоптических данных на основе двухуровневого параллелизма. Выполнена реализация этого подхода в виде модуля для существующего программного комплекса визуализации синоптических данных. Также проведен ряд вычислительных экспериментов для оценки эффективности гибридного параллелизма в задачах обработки синоптических данных для различных программно-аппаратных архитектур.

Применение GPU параллелизма (как на кластере, так и на персональном компьютере) – эффективно при использовании сеток большой детализации. Результаты вычислительных экспериментов показали, что применение гибридного кластерного параллелизма в рамках предложенного подхода позволяет добиться существенного ускорения при условии хранения исходных данных на кластере, в противном случае возникают огромные накладные расходы при обмене большими объемами данных между персональным компьютером и кластером.

### 5. ССЫЛКИ

- [1] Michalakes, J. and M. Vachharajani: GPU Acceleration of Numerical Weather Prediction. Parallel Processing Letters Vol. 18 No. 4. World Scientific. Dec. 2008. pp. 531--548.
- [2] Souto R.P., Osthoff C., Vilasbôas F., Oliveira A. Silva Dias P.L.: Hybrid Cluster Programming Models Evaluation for Atmospheric Simulation Model Application. Laboratório Nacional de Computação Científica (LNCC)
- [3] Osthoff C., Souto R.P., Vilasbôas F., Grunmann P., Silva Dias P.L., Boito F., Kassick R., Pilla L., Navaux P., Schepke C., Maillard N., Panetta J., Pais Lopes P., Walko R. Laboratório Nacional de Computação Científica (LNCC) Atmospheric Model Applications, ISBN: 978-953-51-0488-9, 2012R

- [4] Demeshko I., Maruyama N., Tomita H., Matsuoka S.: Multi-GPU Implementation of the NICAM Atmospheric Model. Euro-Par Workshops 2012: 175-184Sa
- [5] Sipkov' V., L'uchny A., Ga'z'ak M., Experiments with a Hybrid-Parallel Model of Weather Research and Forecasting (WRF) System
- [6] Бобков В.А., Мельман С.В. Система визуализации пространственных полей синоптических объектов // Тр. междунар. конф. по компьютерной графике ГРАФИКОН-2006. М.: МГУ, 2007. [С.@264-268](#).
- [7] Мельман С. В. Визуализация объемов в задачах анализа физических полей синоптических объектов // Информационные технологии, №1, 2008, с. 62-66.
- [8] Kidder, S. Q., M. D. Goldberg, R. M. Zehr, M. DeMaria, J. F. W. Purdom, C. S. Velden, N. C. Grody, and S. J. Kusselson. Satellite analysis of tropical cyclones using the Advanced Microwave Sounding Unit (AMSU).

Черкашин Александр – аспирант ИАПУ ДВО РАН.  
[alexandr@cherkashin.ws](mailto:alexandr@cherkashin.ws)



# Метод построения навигационной системы мобильного робота на основе использования бортовой веб-камеры

Дмитрий Юхимец, Александр Новицкий  
Дальневосточный Федеральный Университет, Владивосток, Россия  
Институт автоматизации и процессов управления ДВО РАН, Владивосток, Россия  
{undim, alexzander}@dvo.ru

## Аннотация

В данной работе предлагается метод построения навигационной системы мобильного робота (МР) на основе использования информации с бортовой камеры. При этом в качестве устройства получения видеоданных выступает веб-камера, которая имеет низкую стоимость, что приводит к удешевлению системы в целом.

**Ключевые слова:** Мобильный робот, Навигация, Веб-камера, Видеопоток, Обработка изображения, Формирование траектории.

## 1. ВВЕДЕНИЕ

Одним из важных факторов, определяющих эффективность функционирования МР, является точность работы их навигационной системы. При этом одним из наиболее эффективных способов повышения указанной точности является использование видеоданных, поступающих с бортовой веб-камеры МР. Одним из наиболее распространенных методов использования видеоданных для навигации является применение стереокамер. Стереокамеры позволяют помимо графического представления окружения МР получать и карту глубин, по которым, в свою очередь, можно определить расстояние до объекта на кадре.

Пример такой системы описан в работе [3] на примере камеры MSVM-III. Указанная система основана на тринокулярной стерео машине, которая выдает карту глубины разрешением 320x240 пикселей. Однако в силу высокой стоимости указанное решение не является общедоступным.

Другой подход описан в работе [2], где навигационная система МР использует всенаправленную камеру и инфракрасные датчики. Указанный метод обладает высокой эффективностью, однако недостатком его является дороговизна и низкая распространенность всенаправленных камер, а также необходимость использования специфических и сложных алгоритмов обработки видеoinформации.

Еще одним примером использования веб-камеры является МР X80 [4]. Указанное решение отличается не высокой ценой, однако это обуславливается отсутствием бортового компьютера, так как все расчеты проводятся на внешних системах, на которые данные передаются посредством Wi-Fi. Основным недостатком такого подхода является низкая автономность получаемой системы.

Таким образом, в настоящее время существует потребность в создании методов построения навигационных систем МР на основе использования видеоданных, которые строились бы на

недорогих стандартных аппаратных и программных средствах.

## 2. НАВИГАЦИЯ МОБИЛЬНОГО РОБОТА НА ОСНОВЕ ИСПОЛЬЗОВАНИЯ БОРТОВОЙ ВЕБ-КАМЕРЫ

Основной задачей, которая решается в данной работе, является формирование траектории движения МР и сигналов обратных связей для его системы управления на основе данных, получаемых с бортовой веб-камеры. При этом указанная задача рассматривается для двух режимов движения: движение по целевым точкам, обозначенным маркерами, и движение по траектории, задаваемой цветной линией.

### 2.1 Алгоритм работы системы

В общем виде алгоритм работы навигационной системы на основе использования данных, получаемых с бортовой веб-камеры, состоит из следующих этапов.

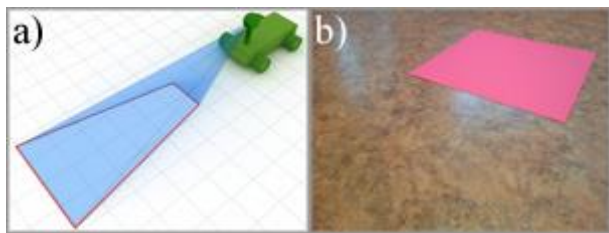
1. Получение кадра с бортовой камеры.
2. Устранение перспективных искажений.
3. Вычисление соотношения между пиксельными координатами и координатами в системе координат (СК) МР.
4. Для режима движения по целевым точкам:
  - a. выделение маркеров;
  - b. классификация выделенных маркеров.
5. Для режима движения по криволинейной траектории:
  - a. выделение криволинейной траектории;
  - b. расстановка контрольных точек на выделенной криволинейной траектории;
  - c. построение кубического сплайна по полученным контрольным точкам.
6. Использование полученной информации для управления МР.

Таким образом, результатом работы алгоритма, в зависимости от режима движения МР, будет являться массив кубических сплайн-функций или классифицированных маркеров с их координатами относительно МР. При этом для реализации алгоритма используются функции из свободной библиотеки с открытым исходным кодом *OpenCV*.

### 2.2 Устранение перспективных искажений

На изображении, полученном с веб-камеры, всегда присутствуют перспективные искажения, пример которых показан на рис. 2.1. На этом рисунке (а) красным контуром

отмечена область, которую охватывает камера при съемке. Как видно, эта область имеет трапециевидную форму, в то время как конечный кадр является прямоугольником.



**Рис 2.1:** Схематичное изображение области, которую охватывает камера (а); пример кадра, полученного с бортовой камеры (б)

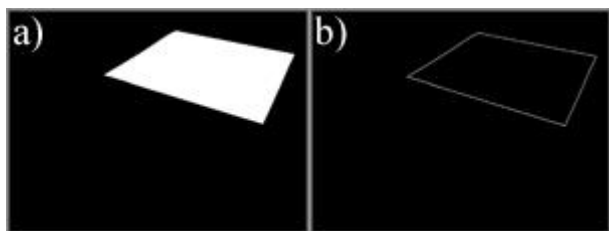
Процесс устранения перспективных искажений заключается в нахождении такого преобразования, которое позволит перейти от прямоугольного кадра, получаемого с видеокамеры, к форме области, охватываемой камерой. Для нахождения указанного преобразования используется эталонный маркер, а сам алгоритм состоит из следующих шагов.

На первом шаге происходит выделение калибровочного маркера из общего фона на основе информации о его цвете. Для этого последовательно используются следующие функции:

- *cvSplit* разбивает изображение на три цветовых составляющих RGB пространства;
- *cvInRangeS* отсеивает в изображении все, что выходит за рамки заданных пороговых значений;
- *cvAnd* суммирует слои, полученные на предыдущем этапе.

В результате формируется бинарное изображение, которое содержит только изображение калибровочного маркера. Пример работы указанной функции показан на рисунке 2.2 (а).

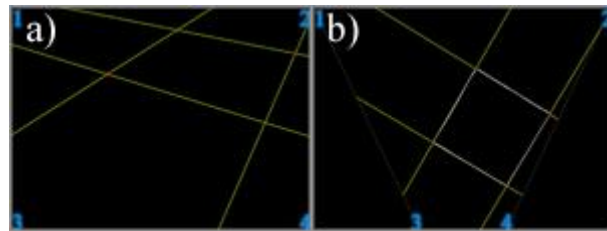
На втором шаге проводится выделение границ эталонного маркера на изображении, полученном на предыдущем шаге. Для этого используется фильтр Кенни, реализованный в функции *cvCanny*. Результат ее работы показан на рис. 2.2 (б).



**Рис 2.2:** Результаты отсеивания по пороговым значениям (а) и работы функции *cvCanny* (б)

На третьем шаге происходит выделение прямых линий на полученном изображении и нахождение такого преобразования, которое переводит полученную на изображении форму маркера в форму, соответствующую эталону. Для этого сначала с помощью преобразований Хафа, реализованных в функции *cvHoughLines2*, проводится выделение прямых линий (см. рис. 2.3 (а)). Далее на основе информации о геометрической форме эталонного маркера с помощью функции *cvWarpPerspective* происходит изменение

угла между линиями, проходящими через точки 1,3 и 2,4 (см. рис. 2.3 (б)).



**Рис 2.3:** Результаты работ функций *cvHoughLines2* (а) и *cvWarpPerspective* (б)

Например, в случае, когда эталонный маркер имеет форму квадрата, изменение угла между указанными линиями можно прекратить тогда, когда угол между двумя любыми линиями, составляющими контур эталонного маркера и имеющими точку пересечения, станет равным 90°.

### 2.3 Вычисление коэффициента масштабирования между пиксельными координатами на кадре и координатами в системе мобильного робота

После устранения перспективных искажений необходимо определить соответствие между пиксельными координатами на изображении и координатами в СК МР. Для этого используется бинарное изображение, содержащее выделенный эталонный маркер (см. рис. 2.2(а)). Это изображение можно рассматривать как матрицу, каждому элементу которой в соответствие ставится цвет пикселя, в данном случае 0 – черный, 1 – белый. Сложив все элементы матрицы, мы получаем количество белых пикселей, которыми представлен калибровочный маркер. Эта величина *S* соответствует площади, занимаемой маркером, при этом, если маркер имеет квадратную форму,  $S^{1/2}$  будет соответствовать длине его стороны в пикселах. Таким образом, масштабирующий коэффициент между пиксельными координатами и координатами в СК МР можно вычислить по следующей формуле:

$$K = \frac{\sqrt{S}}{A},$$

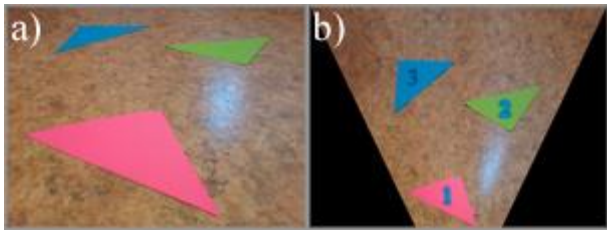
где *A* – размер стороны эталонного маркера в мм, а пиксельные координаты некоторой точки на изображении можно пересчитать в координаты в СК МР по следующим формулам:

$$x_c = (x_n - \frac{W}{2}) \cdot K, y_c = (H - y_n) \cdot K,$$

где  $x_n$  и  $y_n$  – пиксельные координаты, а *W* и *H* – ширина и высота кадра соответственно.

### 2.4 Режим движения по целевым точкам

В режиме движения по целевым точкам навигационная система МР должна определить нужный маркер, который задает целевую точку и его положение в системе координат МР. Следует отметить, что в данной работе рассматривается ситуация, когда маркеры расположены на плоскости движения мобильного робота и отличаются от общего фона достаточно ярким, контрастным цветом (см. рис. 2.4).



**Рис 2.4:** Кадр с маркерами до (а) и после (б) устранения перспективных искажений

Процесс выделения маркеров из изображения, полученного с бортовой видеокамеры, аналогичен процессу выделения эталонного маркера, описанному выше. При этом одновременно происходит разделение маркеров по цветам. Результат выделения маркеров из изображения на рис. 2.4 (б) показан на рис. 2.5.



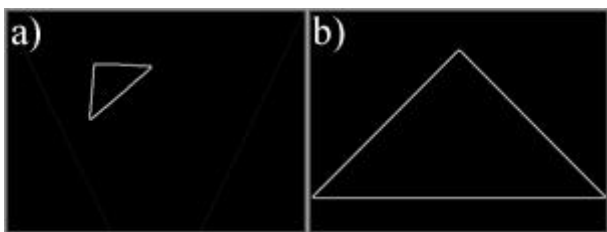
**Рис 2.5:** Набор бинарных изображений, полученный после выделения маркеров

Каждое из этих изображений уже классифицировано относительно цвета и дальше будет рассматриваться только в своем классе.

После начального разбиения по цветовым классам необходимо классифицировать маркеры по форме, так как предполагается, что изначально известны все эталонные геометрические формы маркеров, которые передаются алгоритму в виде бинарных изображений.

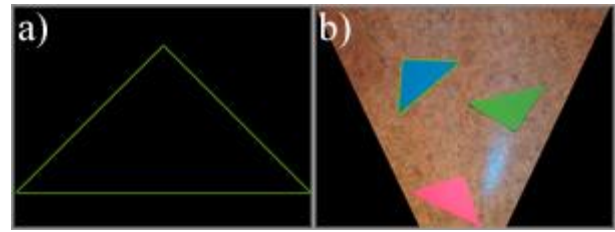
Задача классификации выделенных маркеров по геометрической форме сводится к выделению контуров этих маркеров и их сравнению с эталонами.

Выделение контуров маркеров на изображении происходит в два этапа. На первом этапе с помощью функции *cvCanny* происходит выделение линий (результат показан на рис. 2.6), а на втором этапе с помощью функции *cvFindContours* происходит выделение всех замкнутых контуров на изображении, сформированном функцией *cvCanny*.



**Рис 2.6:** Результат выполнения функции *cvCanny*

В результате прохода в цикле всех замкнутых контуров, найденных на кадре, полученном с бортовой камеры, будет найден контур, который имеет наименьшую ошибку, выдаваемую функцией *cvMatchShapes*, отвечающей за сравнение моментов [1] – суммарных характеристик контуров, рассчитанных интегрированием всех пикселей контура. Найденный маркер обведен зеленой рамкой на рис. 2.7



**Рис 2.7:** Результат выполнения функции *cvMatchShapes*

При этом в качестве координат маркера принимается среднее арифметическое координат последовательности точек контура:

$$x_c = \frac{(\sum_{i=0}^N x_i)}{N}, y_c = \frac{(\sum_{i=0}^N y_i)}{N},$$

где  $N$  – количество контрольных точек контура, а  $x_i$  и  $y_i$  – координаты одной точки контура.

После определения координат маркера эти координаты передаются в СУ МР для формирования сигналов управления его исполнительными устройствами.

## 2.5 Режим движения по криволинейной траектории

Криволинейной траекторией МР в данной работе является линия, нанесенная в плоскости перемещения МР. Также, как и в случае движения по целевым точкам, в данном режиме необходимо выделить указанную линию. Для этого, как было описано ранее, выполняется коррекция перспективных искажений и цветовая фильтрация. Результат выполнения этих операций представлен на рисунке 2.8.

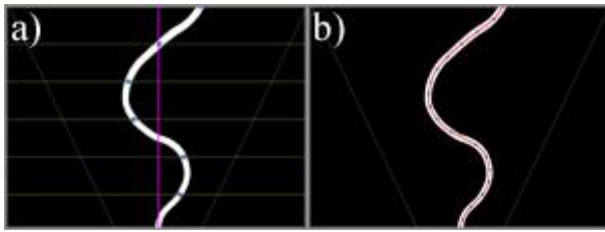


**Рис 2.8:** Пример криволинейной траектории

После получения бинарного изображения необходимо сформировать аналитическое описание траектории, для чего целесообразно использовать описание в виде кубического сплайна.

Для построения кубического сплайна необходимо получить набор опорных точек, через которые будет проходить указанный сплайн. Для этого на бинарном изображении траектории с заданным шагом по вертикальной координате изображения (рис. 2.9 (а), зеленые горизонтальные линии) расставляются опорные точки (рис. 2.9 (а), синие точки), которые лежат на средней линии выделенной траектории.

Координаты сформированных опорных точек используются для определения коэффициентов сплайнов, описывающих траекторию на каждом отрезке между двумя соседними опорными точками. На рисунке 2.9 (б) белым цветом изображена криволинейная траектория, полученная с бортовой камеры, синим – отмечены опорные точки, а красным – показан интерполирующий кубический сплайн, проходящий через эти опорные точки.



**Рис 2.9:** Расстановка опорных точек сплайна (а), сравнение кривой на плоскости и сплайна, построенного по опорным точкам (б)

Далее полученное аналитическое описание траектории параметризуется в функции времени, а пиксельные координаты траектории пересчитываются в координаты СК МР и передаются в его систему управления. После достижения предпоследней опорной точки цикл формирования траектории повторяется.

### 3. АПРОБАЦИЯ АЛГОРИТМА

В качестве платформы, на которой проводилось тестирование разработанного алгоритма навигации, выступал мобильный робот, оснащенный бортовым компьютером на базе смартфона Efika MX и бортовой веб-камерой Logitech HD Webcam C270 (рис. 3.1). В процессе тестирования указанный мобильный робот перемещался в закрытом помещении со стационарным освещением, не оснащенном специальным навигационным оборудованием.



**Рис 3.1:** Мобильный робот на базе смартфона Efika MX

Для оценки вычислительной эффективности разработанного алгоритма была исследована производительность и точность работы указанного алгоритма в зависимости от разрешения исходного изображения.

Результаты исследований показали, что зависимость производительности алгоритма от разрешения изображения – линейная. При этом при разрешении 320x240 пикселей алгоритм на бортовом компьютере МР позволяет формировать навигационную информацию 10-15 раз в сек., что достаточно для эффективного управления движением МР.

В таблице 3.1 приведена зависимость погрешности в вычислении линейных размеров маркера от разрешения полученного кадра. Максимальная погрешность отмечена на разрешении 320x240 пикселей и составляет всего около двух миллиметров, что допустимо в задачах управления движением МР. Также было проведено тестирование работы СУ МР на основе данных, формируемых разработанным алгоритмом, которое показало, что МР точно проходит по заданной последовательности маркеров. Это подтверждает вычислительную эффективность и точность работы алгоритма.

Разрешение	Размер в пикселях	Вычисленный размер, мм	Погрешность, мм
1280 X 960	508,84	296,8252	0,159622
1024 X 768	407,71	296,2707	0,714161
800 X 600	318,94	298,0269	1,042024
640 X 480	253,21	295,1834	1,801406
320 X 230	126,61	294,8458	2,139078

**Табл 3.1:** Зависимость погрешности от разрешения исходного кадра

### 4. ЗАКЛЮЧЕНИЕ

В работе был разработан метод построения навигационной системы МР на основе бортовой веб-камеры. Основное достоинство предложенного подхода заключается в использовании стандартных аппаратных средств и свободного программного обеспечения, что позволяет существенно снизить стоимость всей системы в целом. Разработанный алгоритм позволяет организовать движение МР в двух режимах: движение по целевым точкам, обозначенным маркерами, и движение по криволинейной траектории. Тестирование алгоритма на реальном МР показало, что он обладает высокой вычислительной эффективностью, что обеспечивает его работоспособность на бортовом компьютере МР, а также высокой точностью определения координат маркеров и заданной траектории.

### 5. БЛАГОДАРНОСТИ

Работа была выполнена при поддержке РФФИ (грант №11-07-98502).

### 6. ССЫЛКИ

- [1] Bradski G., Kaehler A. Learning OpenCV / O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. 2008. С. 251 - 255.
- [2] Hsieh. C., Wang M., Kao L., Lin H. Mobile Robot Localization and Path Planning Using an Omnidirectional Camera and Infrared Sensors / Proc. of Int. Conf. Systems, Man and Cybernetics, San Antonio, 2009, pp. 1947 – 1952.
- [3] Jia Y., Li M., An L., Zhang X. Autonomous navigation of a miniature mobile robot using real-time trinocular stereo machine/ Proc. of Int. Conf. Robotics, Intelligent Systems and Signal Processing, 2003, vol. 1, pp. 417 - 421.
- [4] Popa A. S., Popa M., Silea I., Varlan A. Indoor Navigation of a Wheeled Mobile Robot / Proc. of Intl. Conf. Human System Interactions, Krakow, 2008, pp. 1009 – 1014.

### Об авторах

Дмитрий Юхимец – к.т.н, ИАПУ ДВО РАН. Его адрес: [undim@iacp.dvo.ru](mailto:undim@iacp.dvo.ru).

Александр Новицкий – аспирант ИАПУ ДВО РАН. Его адрес: [alexzander@dvo.ru](mailto:alexzander@dvo.ru).

# Сопровождение людей в системах многокамерного видеонаблюдения для спортивных игр

Павел Батанов, Владимир Кононов, Антон Конушин  
Факультет вычислительной математики и кибернетики

Московский Государственный Университет имени М.В.Ломоносова, Москва, Россия  
{pbatanov,vkononov,ktosh}@graphics.cs.msu.ru

## Аннотация

Видеонаблюдение в спорте применяется уже продолжительное время, в особенности, в профессиональных состязаниях. Но, в большинстве своем, задачи, решаемые в этой области, полагаются на высококачественное дорогостоящее оборудование, доступное только для лидеров в мире спорта или ручную разметку данных.

В данной работе описывается алгоритм для сопровождение спортсменов на поле с использованием более доступного оборудования. Описанный подход может быть легко масштабирован на использование произвольного числа камер, наблюдающих одно и то же игровое поле.

**Ключевые слова:** видеонаблюдение, сопровождение людей, трекинг, спортивные игры.

## 1. ВВЕДЕНИЕ

В современных командных видах спорта одной из проблем является сбор данных о передвижениях игроков во время спортивных мероприятий для последующего анализа или отображения положения игроков в реальном времени. Такие данные как график скорости передвижения игрока, статистика его положения на поле, расстояние, которое игрок преодолел с той или иной скоростью, являются ценной информацией, как во время тренировки, так и во время матча для повышения интереса зрителя. Также эти данные могут использоваться при найме новых спортсменов в коллектив в качестве характеризующих спортивных показателей

В данный момент описанная задача сопровождения решается либо с помощью системы дорогостоящего оборудования, либо через использование дешевого наемного ручного труда для разметки уже отснятого видео. Частично данную задачу можно решить использованием высокотехнологичной формы для спортсменов, например, датчиков в бутсах, которые подсоединяются к компьютеру после матча для выгрузки информации. Однако, такой подход не может быть использован для вычисления положения игрока на поле и, тем более, для получения и обработки данных в режиме реального времени. Поэтому задача сопровождения людей в видеозаписях спортивных игр является актуальной.

## 2. ПОСТАНОВКА ЗАДАЧИ

### 2.1 Входные данные

Видеопоследовательности  $u_i$ , содержащие запись спортивной трансляции с нескольких неподвижных и откалиброванных камер, с объектами площадью не менее 100 пикселей:

$$y(t) = \langle y_i(t), \quad i = 1..numCameras, t = 0..numFrames \rangle$$

Объединение всех ракурсов видеопоследовательности покрывают полностью игровое поле. Видеозаписи также синхронизированы по времени

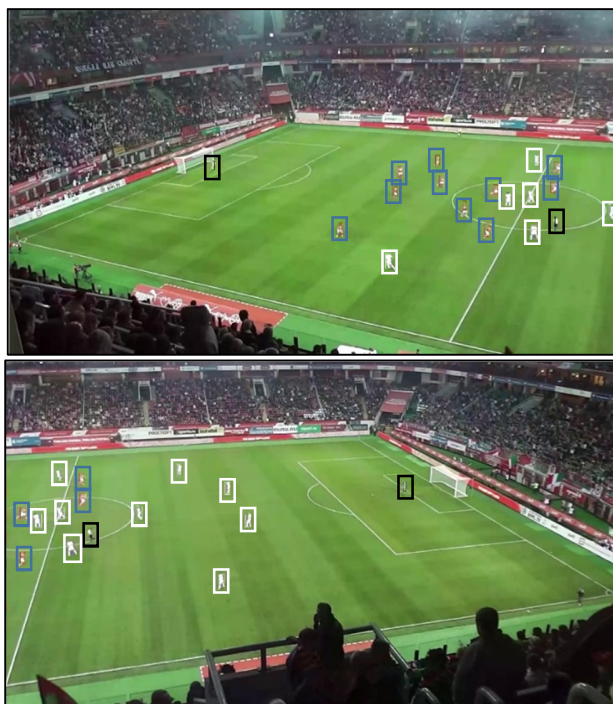


Рис 1: Примеры кадров из входных данных

### 2.2 Выходные данные

- Измеренные характеристики всех сопровождаемых объектов на всех кадрах последовательности
$$x(t) = \langle x_i(t), \quad i = 1..numObjects, t = 0..numFrames \rangle$$
$$x_i(t) = \langle u_i(t), v_i(t), w_i(t) \rangle$$
  - Положение ( $u_i(t)$ ) на плоскости поля
  - Скорость ( $v_i(t)$ )
  - Размеры описывающей фигуры ( $w_i(t)$ )
- Индикация возможных ошибок работы алгоритма для дальнейшего исправления с участием оператора
  - Сопровождение объекта более чем одной меткой («склеивание меток»)
  - Потеря объекта сопровождения

### 2.3 Критерии качества

#### 2.3.1 Точность алгоритма

В качестве основной метрики качества работы алгоритма будет выступать метрика MOTP [1]

$$MOTP = \frac{\sum_t \sum_i d_{i,t}}{\sum_t c_t}$$

- $d_{i,t}$  – метрическое расстояние между реальным и измеренным положением  $i$ -го объекта в момент времени  $t$
- $c_t$  – количество объектов в момент времени  $t$

#### 2.3.2 Время работы

Для использования алгоритма в режиме реального времени в спортивных состязаниях важна скорость обработки каждого поступившего кадра. Современные системы видеонаблюдения, в основном, используют в качестве рабочей частоты значения в 12-30 кадров в секунду. Таким образом, время обработки кадра на современном оборудовании должно производиться за 30-100 мс, в противном случае алгоритм будет работать с задержками либо с пониженной точностью (пропуском кадров). Также стоит отметить, что при более высокой частоте поступления кадров смещение сопровождаемых объектов между кадрами становится меньше, что позволяет регулировать быстродействие алгоритма за счет снижения размера пространства поиска возможных положений объекта.

## 3. ПРЕДЛОЖЕННЫЙ МЕТОД

#### 3.1.1 Фильтр частиц

Фильтр частиц [2] представляет собой последовательный рандомизированный метод для оценки сложных Байесовских сетей. Под сложными сетями понимаются сети с непрерывными состояниями или сложными зависимостями между переменными (например, в линейных динамических системах). Цель этого метода – оценить скрытые значения (положение объекта) по наблюдаемым данным (изображениям с разных ракурсов)  $p(x_t|y_0, \dots, y_t)$ .

Для работы с фильтром частиц процесс движения объектов моделируется как Марковская цепь первого порядка следующим образом:

$x_0, x_1 \dots$  - Марковский процесс первого порядка такой, что  $x_t|x_{t-1} \sim p_{x_t|x_{t-1}}(x|x_{t-1})$ , где  $x_0$  – известное начальное распределение. Для нашей задачи состоянием было выбрано положение объекта  $z$ , его скорость  $v$  и размеры ограничивающего цилиндра  $w$ .

$y_0, y_1 \dots$  - Наблюдения, условно независимы, при условии известности  $x_0, x_1 \dots$ , иными словами  $y_t$  зависит только от  $x_t$ :  $y_t|x_t \sim p_{y_t|x_t}(y|x_t)$

В частности,

$$\begin{aligned} x_t &= g(x_{t-1}) + m \\ y_t &= h(x_t) + z \end{aligned}$$

где  $m$  и  $z$  – взаимно независимые и одинаково распределенные случайные величины с известной плотностью вероятности,  $g()$  и  $h()$  – известные функции.

Функция  $x_t = g(x_{t-1}) + m_t$  представляет собой модель движения объекта. В качестве модели движения для нашей

задачи была выбрана линейная модель со сглаживанием скоростей с параметром  $\alpha$ :

$$\begin{aligned} u_t &= u_{t-1} + v_{t-1} \cdot \Delta t + m_u \\ v_t &= \alpha \cdot v_{t-1} + (1 - \alpha) \cdot (u_{t-1} - u_t) + m_v \\ w_t &= w_{t-1} + m_w \end{aligned}$$

где  $w_i$  – шум соответствующей компоненты состояния.

Модель движения дает априорную информацию о том, где может находиться объект в следующий момент времени. Для того, чтобы оценить  $p(x_t|y_0, \dots, y_t)$ , как и во многих рандомизированных методах (например, [3]), генерируется набор гипотез  $x_t^L$  (так же называемые частицами, отсюда название метода – «фильтр частиц»).

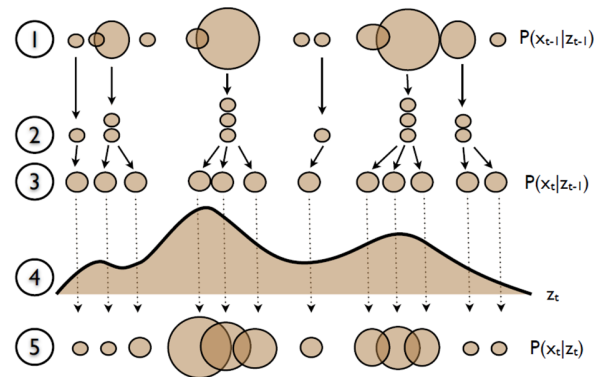


рис. 1 Схема фильтра частиц. Шаги фильтрации: (1) апостериорная функция  $t-1$ ; (2) генерация новых частиц; (3) построение гипотез (априорная функция); (4) измерения; (5) апостериорная функция  $t$ ; [8]

Тогда моменты могут быть оценены в соответствии с фильтрующим распределением следующим образом [4]:

$$\int f(x_k)p(x_k|y_0, \dots, y_k)dx_k \approx \frac{1}{P} \sum_{L=1}^P f(x_k^{(L)})$$

В генерации новых гипотез участвуют старые с вероятностью, пропорциональной их весу на предыдущем этапе [5]. Вес гипотезы оценивается из вероятности того, что гипотеза верна. В этом случае большая часть новых гипотез будет сгенерирована из частиц с большими весами. Но так же существует вероятность генерации и из частиц с малым весом.

После этапа генерации созданный набор частиц подлежит оценке с помощью модели фона и классификатора.

#### 3.1.2 Моделирование фона

На данном этапе обработки новых поступивших кадров всех видеопоследовательностей происходит детектирование объектов путем моделирования фона.



Рис. 2 Пример пользовательской маски поля (слева) и выхода алгоритма ViVe (справа)

Моделирование фона выполняется с помощью алгоритма ViBe [6], который способен рассчитывать маску переднего плана последовательно для каждого нового кадра с поддержкой обновляемой модели фона. Дополнительно пользователь может задать глобальную маску, указывающую область кадра (Рис. 2), в которой детектирование производится не будет. Это требуется, если, например, в кадр попадают трибуны.

На выходе этапа моделирования фона получается маска переднего плана для каждого ракурса.

На приведенном примере (Рис. 3) видно, что кроме самих объектов выделяются также и другие динамические объекты, такие как, например, рекламные щиты вдоль поля. Это затрудняет работу при недостатке ракурсов вдали от камер, если положение камер таково, что сопровождаемые спортсмены могут пересекаться с этими объектами. Это выражается в неверной маске переднего плана для них.

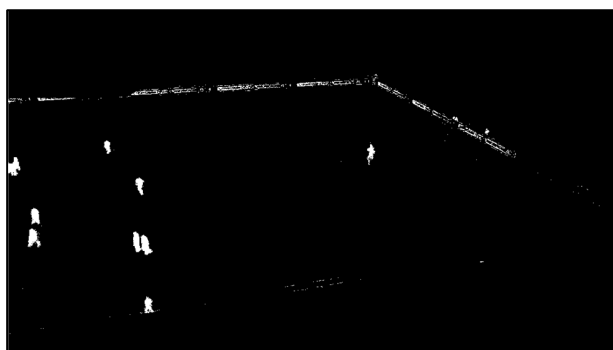


Рис. 3 Пример полученной маски переднего плана

### 3.1.3 Классификация гипотез

Для классификации каждой гипотезы фильтра частиц используется заранее обученный классификатор «случайный лес принятия решений» [7] ( $RF$ ) и полученная на предыдущем шаге маска переднего плана. Для обучения классификатора используются фрагменты изображений, полученных в ходе первоначального выделения объектов пользователем.

В качестве признаков фрагментов строится трехмерная гистограмма цветов пикселей фрагмента в пространстве RGB. Размерность гистограммы –  $8 \times 8 \times 8$

Выходом классификатора является вектор голосов за принадлежность фрагмента изображения к определенному классу (Рис. 4):

- Игрок первой команды
- Игрок второй команды
- Вратарь первой команды
- Вратарь второй команды
- Судья
- Газон

Классы вратарей были добавлены в связи с тем, что в некоторых командных видах спорта форма вратарей сильно отличается от основной формы команды.



Рис. 4 Примеры объектов разных классов

Для классификации гипотеза  $p$  проецируется на каждый ракурс  $c_i$ . После этого извлекается фрагмент кадра и участок маски переднего плана, соответствующий проекции гипотезы на данном ракурсе.

$$[patch(p, c_i), mask(p, c_i)] = projectParticle(p, c_i)$$

Если на некотором ракурсе не существует проекции гипотезы, то данный ракурс помечается для данной гипотезы как недействительный, чтобы в дальнейшем отбросить его при объединении данных.

Если для данного ракурса участок маски переднего плана, на которую проецируется гипотеза, содержит менее половины пикселей переднего плана, то данная гипотеза помечается как недействительная для всех ракурсов.

Для фрагмента изображения строится трехмерная гистограмма  $hist$ , которая подается на вход описанному классификатору. На выходе получается вектор голосов. Из этого вектора формируется значение, соответствующее доле голосов за целевой класс метки. Это значение становится весом гипотезы для данного вида.

$$votes = RF(hist(patch(p, c_i)))$$

$$votes = \langle v_j, j = 1..numClasses \rangle$$

$$weight(p, c_i) = v_{objectClass}$$

Далее вес гипотезы умножается на процент маски переднего плана, попавшей во фрагмент изображения, соответствующего рассматриваемой гипотезе. Таким образом стимулируются гипотезы, покрывающие только объект.

$$weight(p, c_i) = weight(p, c_i) \cdot \sum mask(p, c_i) \cdot size(mask(p, c_i))$$

На выходе этапа измерения получается массив векторов с весами всех гипотез для каждого ракурса (или спец. значениями, обозначающими недействительность ракурса или гипотезы).

### 3.1.4 Объединение данных

После того, как для каждого вида был сформирован вектор весов гипотез, происходит этап объединения данных со всех ракурсов:

- Гипотезы, которые недействительны для всех ракурсов, получают вес 0.
- Гипотезы, которые, у которых есть хотя бы один действительный ракурс, получают вес, равный максимальному из весов для каждого ракурса этой гипотезы:

$$weight(p) = \max_{i=1..numCameras} weight(p, c_i)$$

Функция максимума была выбрана для того, чтобы обрабатывать перекрытия объектов при достаточном количестве ракурсов.

На данном шаге получается массив весов для каждой гипотезы:

$$weight(p), p = 1..numParticles$$

Из этого вектора берется подмножество гипотез с весами, не меньшими, чем  $0.7 \cdot \max_p weights(p)$ , на основе которых делается оценка путем усреднения их состояний.

## 4. ЭКСПЕРИМЕНТАЛЬНАЯ ОЦЕНКА

Алгоритм был протестирован на собранной тестовой выборке, которая представляет собой 300 размеченных кадров футбольного матча, снятого с двух ракурсов.

Алгоритм запускался с использованием 96 гипотез для каждой метки.

Алгоритм при данных параметрах показал метрику MOTP на размеченных данных равную 87.77 см

Время работы алгоритма составляет 4 секунды на кадр на процессоре Intel Core i7-3770K для версии в среде MATLAB с использованием встроенных средств многопоточности.

На C++ время работы алгоритма составило в среднем 80мс на один кадр, что позволяет обрабатывать видеопоследовательности с суммарной частотой до 24 кадров в секунду.

Предложенный методы был сравнен с методом [8], как наиболее схожим по области применения современным алгоритмом, на тех же данных с теми же параметрами.

Метрика MOTP для алгоритма [8] составила 224.11 см.

## 5. ЗАКЛЮЧЕНИЕ

В нашей работе мы предложили усовершенствованный алгоритм сопровождения людей в видеопоследовательностях, который применим для работы с видеозаписями спортивных трансляций.

## 6. БЛАГОДАРНОСТИ

Работа была выполнена при частичной поддержке гранта 10255p/16855 программы «УМНИК» фонда содействия развитию малых форм предприятий в научно-технической сфере и гранта Президента Российской Федерации для молодых ученых - кандидатов наук МК-4644.2012.9.

## 7. ССЫЛКИ

- [1] K. Bernardin и R. Stiefelwagen, «Evaluating multiple object tracking performance: The CLEAR MOT metrics.» *Journal on Image and Video Processing - Regular*, 2008.
- [2] M. Arulampalam, S. Maskell, N. Gordon и T. Clapp, «A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking.» *IEEE Transactions on Signal Processing*, pp. 174-188, 2002.
- [3] I. Murray, «Markov chain Monte Carlo.» 2009. [В Интернете]. Available: <http://homepages.inf.ed.ac.uk/imurray2/teaching/09mlss/slides.pdf>.
- [4] Д. Ветров, «Лекция 11. Методы Монте-Карло. Фильтр частиц.» 2009. [В Интернете]. Available: <http://courses.graphicon.ru/files/courses/smsa/2009/lectures/lecture10.pdf>.
- [5] A. Doucet, «Sequential Importance Sampling Resampling.» *Departments of Statistics & Computer Science*, 2010. [В Интернете]. Available: [http://www.cs.berkeley.edu/~jordan/courses/260-spring10/readings/samsi\\_lec3.pdf](http://www.cs.berkeley.edu/~jordan/courses/260-spring10/readings/samsi_lec3.pdf). [Дата обращения: 2012].
- [6] O. Barnich и M. Van Droogenbroeck, «ViBe: A universal background subtraction algorithm for video sequences.» *IEEE Transactions on Image Processing*, pp. 1709-1724, June 2011.
- [7] L. Breiman, «Random Forests.» *Machine Learning*, т. 45, № 1, pp. 5-32, 2001.
- [8] E. Morais, S. Goldenstein, A. Ferreira и A. Rocha, «Automatic tracking of indoor soccer players using videos from multiple cameras.» в *SIBGRAPI*, Ouro Preto, 2012.

## Об авторах

Павел Батанов – выпускник факультета ВМК МГУ имени М.В. Ломоносова. Его адрес: [pbatanov@graphics.cs.msu.ru](mailto:pbatanov@graphics.cs.msu.ru).

Владимир Кононов – аспирант факультета ВМК МГУ имени М.В. Ломоносова. Его адрес: [vkonoenov@graphics.cs.msu.ru](mailto:vkonoenov@graphics.cs.msu.ru)

Антон Конушин – доцент факультета ВМК МГУ имени М.В. Ломоносова. Его адрес: [ktosh@graphics.cs.msu.ru](mailto:ktosh@graphics.cs.msu.ru)



# Визуализация средствами графических процессоров (GPGPU) смоделированных атомных структур аморфных сплавов

Александр Дубинец, Евгений Пустовалов, Александр Федорец, Станислав Полищук, Владимир Ткачѳв  
Кафедра компьютерных систем

Дальневосточный федеральный университет, Владивосток, Россия  
{dubinec.av, pustovalov.ev, fedorec.an, polischuk.sv}@dvfu.ru, vova292@gmail.com

## Аннотация

Данная работа посвящена моделированию и визуализации моделей атомных структур аморфных сплавов средствами GPU. Модель аморфного сплава представляется как набор атомов. Визуализация производится в двух вариантах: двумерном, как электронно-микроскопическое изображение и трехмерном как 3d-модель атомов сплава. В первом, случае высокая скорость расчета кадра достигается путем использования возможностей параллельных вычислений, с помощью программного интерфейса CUDA. Для визуализации объемных моделей используется расширение OpenGL, Vertex Array Object (VAO).

**Ключевые слова:** Визуализация, GPU, CUDA, OpenGL, аморфные сплавы.

## 1. ВВЕДЕНИЕ

Аморфные металлические и нанокристаллические сплавы (АМС) – перспективный класс металлических материалов, обладающих уникальным сочетанием магнитных, электрофизических, механических и коррозионных свойств. АМС привлекают значительное внимание физиков, работающих в области, как фундаментальных исследований, так и прикладных разработок.

В процессе исследования АМС методами просвечивающей электронной микроскопии, мы получаем изображения структуры, на которых необходимо проанализировать морфологию и атомную структуру. Одним из эффективных инструментов анализа является, моделирование, которое позволяет, создавать модели с заданными параметрами, после чего производится визуализация объекта, как электронно-микроскопического изображения и 3d модели.

В настоящее время требования к программному обеспечению по качеству генерируемых изображений постоянно растут, моделирование, и визуализация атомных структур остается актуальной задачей. Среди наиболее эффективных методов визуализации, выступает: использование графического процессора (GPGPU) в качестве вычислителя, полная загрузка модели на видеоустройство, позволяет избежать медленного непрерывного копирования данных по PCI-E шине.

## 2. ДОСТИЖЕНИЯ В ПРЕДМЕТНОЙ ОБЛАСТИ

На данный момент, существуют следующие аналоги для визуализации моделей атомных структур аморфных сплавов. Программный пакет JEMS [1], который предназначен для представления электронно-микроскопических изображений, но в силу своей реализации не позволяет воспроизводить электронно-микроскопическое изображение с высокой скоростью. Программный пакет, CaRIne Crystallography [2], предназначенный для визуализации трехмерных моделей

атомных структур, не рассчитан на отображение моделей файлов более 5 Мбайт, содержащих около 150000 атомов.

## 3. ВЫБОР ТЕХНОЛОГИИ

Для реализации изменения и обработки моделей атомов был создан редактор реализованный средствами языка программирования C++, который обеспечивает относительно высокую скорость считывания и обработки данных.

Так как необходимо визуализировать статичные данные: координаты, цвета и индексы, которые в процессе отображения не будут изменяться, то наиболее оптимальное решение, использование расширения OpenGL, ARB\_vertex\_buffer\_object (VBO).

```
glEnableClientState( GL_VERTEX_ARRAY );
glEnableClientState( GL_COLOR_ARRAY );
glBindBuffer( GL_ARRAY_BUFFER, cId );
glColorPointer( 3, GL_UNSIGNED_BYTE, 0, 0 );
glBindBuffer( GL_ARRAY_BUFFER, vId );
glVertexPointer( 3, GL_FLOAT, 0, 0 );
glBindBuffer( GL_ELEMENT_ARRAY_BUFFER, iId );
glDrawElements( GL_TRIANGLES, Count,
GL_UNSIGNED_INT, 0 );
glBindBuffer( GL_ELEMENT_ARRAY_BUFFER, 0 );
glBindBuffer( GL_ARRAY_BUFFER, 0 );
glDisableClientState( GL_VERTEX_ARRAY );
glDisableClientState( GL_COLOR_ARRAY );
```

**Листинг 1:** Пример активации буферов без использования VAO

Для упрощения процесса кодирования целесообразно использовать следующее расширение OpenGL, ARB\_vertex\_array\_object (VAO) [3], которое позволяет упростить процедуры активации буферов до одной команды.

```
glBindVertexArray( vao );
glDrawElements( GL_TRIANGLES, n, GL_UNSIGNED_INT, 0 );
glBindVertexArray( 0 );
```

**Листинг 2:** Пример активации буферов с использованием VAO

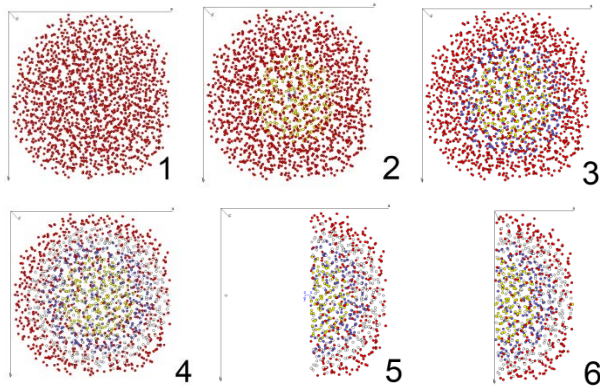
В качестве альтернативной технологии визуализации, можно рассмотреть дисплейные списки, которые выдают большую производительность на статичных данных, однако основной недостаток которых – длительное время на построение списка, и его полное перестроение при любой модификации данных.

## 4. АМОФНЫЕ СТРУКТУРЫ

### 4.1 Моделирование

При моделировании атомной структуры аморфного сплава, производится работа с файлом модели [4], в котором каждый атом представляется набором параметров: название элемента,

степень окисления, три координаты, радиус и коэффициент заполнения. Основная идея заключается в изменении параметров элементов с помощью, элементарных стереометрических объектов: сфера, плоскость, sin-плоскость, тетраэдр и цилиндр. Все это позволяет: создавать флуктуации плотности в заданных областях, внедрять кристаллические структуры в аморфные, изменять морфологию поверхности модели.



**Рисунок 1:** 1 вырезана сфера железа; 2 включение фосфора; 3 добавлен слой никеля; 4 добавлен слой кислорода; 5 срез половины сферы; 6 нормировка ячеек

#### 4.2 2D визуализация

Визуализация моделей средствами GPGPU [5], производится в несколько этапов [6]. Производится вычисление проекций потенциалов отдельных атомов (1) и последующее формирование потенциала образца путем линейной суперпозиции потенциалов каждого атома в образце (2):

$$v_z(X) = \int_{-\infty}^{+\infty} V_a(x, y, z) dz = 4\pi^2 a_0 e \sum_{i=0}^3 a_i K_0(2\pi r \sqrt{b_i}) + 2\pi^2 a_0 e \sum_{i=0}^3 \frac{c_i}{d_i} \exp(-\pi^2 r^2 / d_i) \quad (1)$$

где  $X = (x, y)$  - положение атома в плоскости, перпендикулярной к оптической оси микроскопа;

$K_0(x)$  - модифицированная функция Бесселя нулевого порядка;  $a_0$  - радиус орбиты Бора для электрона;  $a_i, b_i, c_i, d_i$  - коэффициенты, значения которых зависят от атомного номера и приводятся в соответствующих таблицах [7].

$$v_z(X) = \sum_{j=1}^N v_{z_j}(X - X_j) \quad (2)$$

Вычислить функцию прохождения электронов через исследуемый объект (3):

$$t(X) = \exp[i\sigma v_z(X)] \quad (3)$$

где  $\sigma$  - постоянная взаимодействия электронного пучка с образцом.

Взять преобразование Фурье от функции прохождения (4).

$$T(k) = FT[t(X)] \quad (4)$$

Умножить функцию (4) на передаточную функцию объектива (5), чтобы получить волновую функцию в обратной фокальной плоскости (6).

$$H_0(k) = \exp[-i\chi(k)]A(k) \quad (5)$$

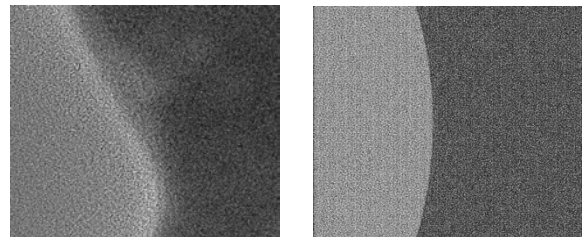
$$\chi(k) = \pi\lambda k^2 (0.5C_s \lambda^2 k^2 - \Delta f)$$

где  $\Delta f$  - дефокусировка;  $C_s$  - коэффициент сферической aberrации;  $A(k)$  - функция апертуры.

$$\psi_i(k) = FT^{-1}[H_0(k)T(k)] \quad (6)$$

Подсчет квадрата модуля волновой функции изображения (в реальном пространстве), чтобы получить конечную интенсивность (7).

$$g(x) = |\psi_i(X)|^2 \quad (7)$$

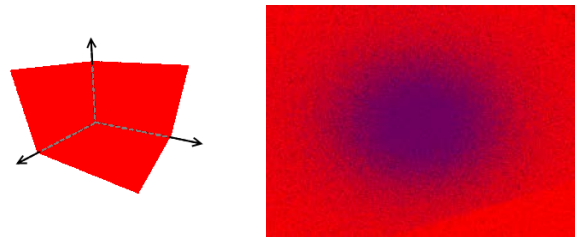


**Рисунок 2:** Электронно-микроскопические изображения: слева реальное, справа смоделированное

Производя визуальное сравнение между реальным и смоделированным изображением, показанных на рисунке 2 мы можем сделать заключения о типе дефекта структуры в реальном объекте.

#### 4.3 3D визуализация

Перед вычислением проекции потенциалов модели, необходимо убедиться в корректности дефектов внесенных в модель, так как процесс создания проекций очень затратный по времени и ресурсам. Визуализация реализуется средствами OpenGL 2.1 с помощью расширения ARB\_vertex\_array\_object (VAO). Данные вершин, цветов и индексов, считываются, полностью загружаются в память видеоустройства, и удаляются из системной памяти.



**Рисунок 3:** Визуализация модели с избытком фосфора в центре, физический размер модели 51\*51\*51 нм, количество атомов 23000000, fps 40.

### 5. РЕЗУЛЬТАТЫ

Все расчеты производились на станции NVIDIA GTX 690. Примерное время обработки модели 51\*51\*51 нм составляет 2-3 минуты. Расчет проекций потенциалов для этой же модели составляет около 5-6 минут. Замеры производительности программ при визуализации приведены в таблицах 1 и 2.

Размеры выводимой области, пх.	Частота кадров, кадр/с	Время кадра, мс
128*128	900	0,11
256*256	300	0,33
1024*1024	60	1,3

**Таблица 1:** Результаты измерений для двумерной визуализации

Количество атомов, атомов.	Частота кадров, кадр/с	Время кадра, мс
125	3500	0,32
650000	3000	0,33
2300000	40	2,5

**Таблица 2:** Результаты измерений для трехмерной визуализации

Возможность средствами операционной системы запускать одновременно нескольких вычислительных потоков на CPU позволяет ускорить в несколько раз процесс создания множества модифицированных моделей.

## 6. ЗАКЛЮЧЕНИЕ

В статье был представлен метод визуализации моделей атомных структур аморфных сплавов средствами GPU. В результате исследования был реализован программный комплекс, который позволяет модифицировать и визуализировать модели в двумерном виде как электронно-микроскопическое изображение и трехмерное, как модель атомов сплава. Процесс моделирование-визуализация происходит практически в реальном времени, что позволяет быстро изменять модели для достижения наибольшего сходства между реальным образцом и смоделированным. Все это позволяет более эффективно интерпретировать результаты экспериментов, позволяя получать материалы с заданными техническими характеристиками.

## 7. БЛАГОДАРНОСТИ

Исследование выполнено при финансовой поддержке Министерства образования и науки, госконтракты № 16.552.11.7059 и 14.740.11.1015.

## 8. ССЫЛКИ

[1] *Simulation of diffraction patterns and high resolution images:*

<http://cimewww.epfl.ch/people/stadelmann/jemsWebSite/jems.html>

[2] *The crystallographic software for research and teaching:*  
<http://carine.crystallography.pagespro-orange.fr/>

[3] *OpenGL VAO programming Guide:*  
<http://www.opengl.org/documentation/books/>

[4] Кристаллография: Лабораторный практикум / Под ред. Проф. Е.В. Чупрунова: Учеб. Пособие для вузов. – М.: Издательство физико-математической литературы, 2005. – 412 с. – ISBN 5-94052-103-7

[5] David B. Kirk, Wen-mei W. Hwu. *Programming Massively Parallel Processors*. 2012. p. 514.

[6] Earl J. Kirkland. *Advanced Computing in Electron Microscopy*. 2010. p. 289.

[7] P.A. Doyle and P.S. Turner. *Relativistic Hartree-Forck x-ray and electron scattering factors*. *ActaCryst.*, A24:390-397, 1968.

## Об авторах

Александр Дубинец – студент магистратуры ДВФУ. Его адрес: [dubinetc.an@dvfu.ru](mailto:dubinetc.an@dvfu.ru).

Евгений Пустовалов – доцент кафедры компьютерных систем ШЕН ДВФУ, к.ф.-м.н. Его адрес: [pustovalov.ev@dvfu.ru](mailto:pustovalov.ev@dvfu.ru).

Александр Федоренко – студент магистратуры ДВФУ. Его адрес: [fedorec.an@dvfu.ru](mailto:fedorec.an@dvfu.ru).

Станислав Полищук – аспирант ДВФУ. Его адрес: [polischuk.sv@dvfu.ru](mailto:polischuk.sv@dvfu.ru).

Владимир Ткачëв – студент ШЕН ДВФУ. Его адрес: [vova292@gmail.com](mailto:vova292@gmail.com).

# Index of Authors

- Barina D., 83  
Belyaev S., 65
- Daming Shi, 99  
Debelov V., 50  
Dolenšek J., 39  
Dolgovssov B., 36
- Eruhimov V., 32
- Frolov V., 75  
Fujun Zhao, 109
- Goncharenko I., 43  
Gruzdev A., 75  
Gudkov V., 60
- Ignatenko A., 75
- Jiancai Wang, 109  
Jianting Shi, 109
- Karthik Mahesh Varadarajan, 95  
Khryashchev V., 71  
Kobrtsek J., 56  
Kogan A., 32  
Kozlov D., 50  
Krylov A., 87
- Lei Xu, 12
- Martyushev E., 24  
Mazurok B., 36  
Mestetskiy L., 91  
Milet T., 56  
Minaev S., 36  
Mongus D., 39
- Nasonov A., 87
- Paltashev T., 48  
Peciva J., 56  
Peng Liu, 103  
Potapova E., 28  
Priorov A., 71, 79
- Rupnik M., 39
- Shemyakov A., 71  
Shmaglit L., 71  
Shubnikov V., 65  
Starka T., 56  
Stepanenko O., 20  
Stožer A., 39
- Tumanov K., 79
- Vincze M., 28  
Volokhov V., 79  
Vostryakov K., 75
- XiangLong Tang, 103  
Xianglong Tang, 99  
XiaoZheng Wang, 103  
XuDong Zhao, 103  
Xudong Zhao, 99
- Yanqin Zhao, 106  
Yongqiang Zhang, 99
- Zemcik P., 56, 83  
Zillich M., 28  
Zimovnov A., 91
- Špelič D., 39  
Žalik B., 39
- Артиков А., 152, 275  
Артиков Т., 152, 275  
Афанасьев В., 215  
Афанасьева А., 178
- Балахонцева А., 250  
Барладян Б., 148  
Батанов П., 333  
Бекетова И., 219  
Бобков В., 238  
Бобков В.А., 114  
Бобков В.В., 244  
Бобкова А., 238, 244  
Бологова Н., 302  
Бондарев А., 127  
Букатов А., 282
- Васёв П., 174

Визильтер Ю., 219  
Волобой А., 148, 186, 199  
Востряков К., 207  
Вяткин С., 140

Гаврилов Н., 195  
Гаганов В., 211  
Галактионов В., 127, 186, 207  
Глазкова М., 254  
Годоба А., 250  
Горбачев В., 226  
Гордеев Д., 157  
Городилов М., 140, 166, 171  
Гридчина Е., 282  
Грудинин С., 286  
Гудков В., 264, 306, 314  
Гурьева М., 280

Дебелов В., 191  
Денисов Е., 199  
Дмитриев С., 278  
Долговесов Б., 131, 140, 152, 166, 171  
Дорофеев К., 264  
Дубинец А., 337

Ершов С., 144

Жданов Д., 144  
Жиляков Е., 230

Заставной Б., 282  
Заставной Д., 282  
Зюзин В., 238, 244

Игнатенко А., 178, 211, 215

Кайнарова Е., 226  
Камаев А., 298  
Каратеев С., 219  
Клименко С., 13  
Козлов Д., 191, 321  
Колосовский М., 316  
Кононов В., 333  
Конущин А., 333  
Коростелев Е., 152, 171  
Крючкова Е., 316  
Кудряшов А., 114  
Кузьмин А., 247  
Кузьмицкий Н., 234  
Кумков С., 174  
Кумсков М., 241, 247  
Курбатов С., 222

Лебедев А., 211  
Левашов А., 259  
Лепихова Д., 314

Лобзин А., 222  
Ломов Н., 310  
Лукин А., 254

Мазурок Б., 152, 166  
Малин И., 294  
Мамаев Н., 254  
Матвеев И., 182  
Машенцев В., 114, 271  
Метелев И., 226  
Морозов Б., 131, 166  
Морозов М., 161

Нагапетян В., 118  
Нгуен Тьен, 250  
Новицкий А., 329

Овчинников А., 122

Перминов И., 267  
Поздняков С., 144  
Полищук С., 337  
Поршнева С., 238, 244  
Прохоров Е., 241  
Пустовалов Е., 337

Роньшин Ю., 114

Свистунов С., 203  
Синицин В., 254  
Соловьёв И., 290  
Суслов А., 306

Тисевич И., 215  
Ткачёв В., 337  
Турлапов В., 195  
Туфанов И., 161

Федорец А., 337  
Фролов В., 207  
Фроловский В., 286

Харламов А., 207  
Хахалин Г., 222  
Хоа Фан, 122

Черкашин А., 325  
Черноморец А., 230

Шапиро Л., 148  
Шмаков Е., 174  
Шулькин Е., 266

Щеголева Н., 135

Юдин И., 182  
Юрин Д., 254, 259  
Юхимец Д., 329

Яковлева Е., 226

УДК 004.9  
ББК 32.973.26-018.2  
Г78

**ГрафиКон'2013:** 23-я Международная конференция по компьютерной графике и зрению: Владивосток, Институт автоматике и процессов управления ДВО РАН: Труды конференции.  
ISBN 978-5-8044-1402-4

УДК 004.9  
ББК 32.973.26-018.2

Подготовка оригинал-макета:  
*М.А. Морозова*

Отпечатано с оригинал-макета, подготовленного  
в Институте автоматике и процессов управления ДВО РАН,  
минуя редподготовку в «Дальнаука» ДВО РАН

Подписано к печати 20.08.2013 г.  
Печать офсетная. Бумага офсетная.  
Формат 60x84/8. Усл. п. л. 43,0. Уч.-изд. л. 41,92.  
Тираж 100 экз. Заказ 82

Издательство «Дальнаука» ДВО РАН  
690041, г. Владивосток, ул. Радио, 7

Отпечатано в Информационно-полиграфическом  
хозрасчетном центре ТИГ ДВО РАН  
690041, г. Владивосток, ул. Радио, 7