

The 20th International Conference
on Computer Graphics and Vision

GraphiCon'2010

September 20-24, 2010
St. Petersburg, Russia

Conference Proceedings

20-я Международная Конференция
по Компьютерной Графике и Зрению

ГрафиКон'2010

20-24 сентября, 2010
Санкт-Петербург, Россия

Труды Конференции

Санкт-Петербургский государственный университет
информационных технологий, механики и оптики

In partnership with Intel corp.



Sponsored by NVIDIA corp.



Sponsored by Microsoft Research



Organized by The National Research University of Information Technologies, Mechanics and Optics



Organized by Moscow State University



Organized by Graphics & Media Lab



Organized by Children Computer Club of St. Petersburg



In cooperation with Eurographics Association



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



Supported by Nizhny Novgorod Foundation for Education and Research Assistance

Information Sponsor



Organizing Committee

Conference Chairs

Vladimir Vasilev (Saint-Petersburg University ITMO)
Vladimir Galaktionov (Keldysh Institute of Applied Mathematics)
Timour Paltashev (Northwestern Polytechnic University)

Organizing Committee Chairs

Alexey Ignatenko (Lomonosov Moscow State University)
Alexander Tropchenko (Saint-Petersburg University ITMO)

International Program Committee Co-Chairs

Gunter Knittel (Tuebingen University)
Alexander Tropchenko (Saint-Petersburg University ITMO)
Anton Konushin (Lomonosov Moscow State University)

International Program Committee

Valery Adzhiev (UK)	Alexei A. Efros (USA)	Gregory Nielson (USA)
Mark W. Bannatyne (USA)	Roland Fleming (Germany)	Victor Ostromoukhov (France)
Brian A. Barsky (USA)	Amy Gooch (Canada)	Dimitri Plemenos (France)
Yuri Bayakovski (Russia)	Atanas Gotchev (Finland)	Boris Prokopenko (USA)
Alexander Belyaev (UK)	Stefan Gumhold (Germany)	Kari Pulli (USA)
Sergey B. Berezin (Russia)	Stefan Guthe (USA)	Martin Reiser (Singapore)
Valery Bobkov (Russia)	Zahid Hussain (USA)	Alexander Reshetov (Russia)
Alexander Bondarev (Russia)	Mikhail V. Iakobovski (Russia)	Michael N. Rychagov (Russia)
Gary Bradski (USA)	Andres Iglesias Prieto (Spain)	Vladimir Savchenko (Japan)
Ilya V. Brailovskiy (USA)	Konstantine Iourcha (USA)	Philipp Slusallek (Germany)
Pere Brunet (Spain)	Denis Ivanov (Russia)	Alexei Sourin (Singapore)
Alexey Chernyavskiy (Russia)	Joaquim A. Jorge (Portugal)	Oliver Staadt (Germany)
Alexander Chipizhko (Russia)	Andrey Krylov (Russia)	Daniel Thalmann (Switzerland)
Sabine Coquillart (France)	Victor Lempitsky (UK)	Oleg Ushmaev (Russia)
Steve Cunningham (USA)	Alexey Lukin (Russia)	Dmitry P. Vetrov (Russia)
Marc Daniel (France)	Tom Malzbender (USA)	Alexey Voloboy (Russia)
Victor A. Debelov (Sciences)	Leonid Mestetskiy (Russia)	Daniel Weiskopf (Germany)
Michael Doggett (Sweden)	Mikhail Mikhailiuk (Russia)	Dmitry V. Yurin (Russia)
Roman Durikovic (Slovakia)	Karol Myszkowski (Germany)	

Address: GraphiCon'2010, St. Petersburg State University of Information Technologies, Mechanics and Optics, Birzhevya lane, 14, St.Petersburg, 199034, Russian Federation, Phone: +7 812 328 0916, +7 812 3280 256, Fax: +7 812 323 7175,
E-mail: gc2010@graphicon.ru.

Contents

Preface	10
---------------	----

Technical Section

English Section

Section 1. Geometry processing

Adaptive Data Hiding: A Hybrid Based High-capacity Approach for 3D Models <i>Shih-Chun Tu, Charlie Irawan Tan and Wen-Kai Tai</i>	14
Adaptive Context Modeling for Efficient Image and Elevation Data Compression <i>Egor Yusov</i>	22
An Efficient Collision Detection Algorithm For Point Cloud Models <i>Mauro Figueiredo, João Oliveira, Bruno Araújo, João Pereira</i>	30
Advanced Mesh Repairing <i>Alexander Emelyanov</i>	38

Section 2. Rendering and visualization

Low Cost Adaptive Anti-Aliasing for Real-Time Ray-Tracing <i>Maxim Shevtsov, Mikhail Letavin and Alexey Rukhlinskiy</i>	45
Biased global illumination via Irradiance Caching and Adaptive Path Tracing on GPUs <i>Vladimir Frolov, Alexander Kharlamov, Alexey Ignatenko</i>	49
Modeling BRDF by a Probability Distribution <i>Aydın Öztürk, Murat Kurt, Ahmet Bilgili</i>	57
Pulse of the city: Visualizing Urban Dynamics of Special Events <i>Andrea Vaccarim, Mauro Martino, Francisca Rojas, Carlo Ratti</i>	64
An Efficient Slice-Based Water Simulation Approach for Breaking Waves <i>Chung-Nan Lee, Jyun-Ming Wu, Yan-Zheng Huang, Da-Jing Zhang-Jian</i>	72

Section 3. Image recognition

Animated thumbnail for still image <i>Iliia V. Safonov, Victor V. Bucha</i>	79
New Ridge Count Model based on Fingerprint Image Topology <i>Vladimir Gudkov</i>	87
Automated Atlas-Based Segmentation of Nissl-Stained Mouse Brain Slices <i>Olga Senyukova, Alexey Lukin, Dmitry Vetrov</i>	92
Improvement of background subtraction by mask constraints <i>Vadim Konushin, Anton Konushin</i>	97

Automatic Extraction of Regular Grids from Rectified Facade Image <i>Anton Yakubenko, Ivan Mizin, Anton Konushin</i>	100
Generalized gamma mixtures for supervised SAR image classification <i>Vladimir A. Krylov, Josiane Zerubia</i>	107

Section 4. Image processing

Efficient Super-Resolution Up-Conversion Algorithm for HDTV <i>Vadim Vashkelis, Natalia Trukhina, Ivan Chirkov</i>	111
Robust Space/Spatial-Frequency based Filtering of Images in the presence of heavy tailed noise <i>Irena Orović, Nikola Žarić, Srdjan Stanković</i>	116
A Novel Smart Bilateral Filter for Ringing Artifacts Removal in JPEG2000 Images <i>Vladimir Khryashchev, Ilya Apalkov, Lev Shmaglit</i>	122
Edge Detection Using Reaction-Diffusion Equation with Variable Diffusion Coefficient <i>Alexander M. Denisov, Andrey S. Krylov, Victoria Yu. Medvedeva</i>	129
Scale Space Edge Curvature Estimation and Its Application to Straight Lines Detection <i>Ekaterina V. Semeikina, Dmitry V. Yurin</i>	133
Short Reference Image Quality Estimation Using Modified Angular Edge Coherence <i>Dmitry V. Sorokin, Andrey S. Krylov</i>	137

Russian Section

Секция 1. Свет в фотореалистичной визуализации

Эффективный метод расчета освещения для сцен с большой глубиной зеркальных отражений и преломлений <i>Константин Востряков</i>	142
Построение спектрального представления из RGB данных в задачах спектрального моделирования <i>Дмитрий Жданов, Игорь Потемин</i>	144
Спектральная трассировка лучей в задачах построения фотореалистичных изображений <i>Б.Х. Барладян, К.А. Востряков, В.А. Галактионов, Д.Д. Жданов, И.С. Потемин, Л.З. Шапиро</i>	148
Моделирование распространения света в тонком красящем слое с высокой концентрацией частиц <i>А.Г. Волобой, С.В. Ершов, Э.С. Клышинский, С.Г. Поздняков</i>	155

Секция 2. Сегментация и обработка изображений

Метод адаптации цифровых изображений для людей с аномальным цветовосприятием (дихроматов) <i>В.А. Ковалёв, Э.В. Снежко, В.И. Архипов</i>	163
Методики выделения связанных компонент в штриховых бинарных изображениях <i>Максим Стержанов</i>	169
Морфологическая сегментация изображений земной поверхности с использованием стековых деревьев <i>Юрий Визильтер, Владимир Горбацевич</i>	175
Агломеративный метод сегментации иероглифического текста <i>А.В. Кугаевских</i>	181
Инструментальная реализация алгоритмов реконструкции искаженных изображений <i>К.А. Кирьянов</i>	188
Определение положения объекта на изображении по фрагментам границы <i>И.С. Симакин</i>	192

Секция 3. Видеонаблюдение

Встроенная видеоаналитика для детектирования и сопровождения объектов при помощи многомасштабных признаков <i>Н.В. Птицын</i>	200
Распределенная система видеомониторинга для интеллектуального пространства <i>А.Л. Ронжин, М.В. Прищепа, В.Ю. Будков, А.А. Карпов, А.Л.Ронжин</i>	207
Отслеживание контура лабораторной мыши в видеопоследовательности на основе метода ветвей и границ <i>Михаил Синдеев, Антон Конушин</i>	214

Секция 4. Скелеты и отрезки

К построению эффективного решения задачи пересечения отрезков <i>Тарас Вознюк, Василий Терещенко</i>	218
Скелет многоугольной фигуры – представление плоским прямолинейным графом <i>Леонид Местецкий</i>	222
Топологическая модель многоуровневой улично-дорожной сети на основе скелета <i>Дмитрий Задонский, Елена Макарова, Иван Мехедов</i>	230

Секция 5. Визуализация и GPU

Верификация алгоритмов фотореалистического рендеринга кристаллов <i>В.А. Дебелов†, Д.С. Козлов</i>	238
Анализ алгоритмов оптимизации времени отображения электронных карт в формате HP-GL <i>Ю.Л. Кетков, З.А. Матвеев</i>	246

Моделирование каустик в реальном времени <i>Денис Боголепов, Дмитрий Сопин, Вадим Турлапов</i>	253
Визуализация полупрозрачных объектов трехмерных сцен реального времени <i>Е.И. Коростелев, Б.С.Долговесов, Б.С. Мазурок</i>	257
Алгоритм восстановления поверхности из облака точек на графическом процессоре <i>Дмитрий Козлов, Вадим Турлапов</i>	263
Моделирование монитора кругового обзора <i>И.В. Валиев, А.Г. Волобой</i>	269

Секция 6. Вопросы визуализации

Методы быстрого формирования изображения в параллельной системе постобработки результатов научных вычислений ScientificView <i>А.Л. Потехин</i>	273
Поиск оптимального пути на множестве преград в компьютерных играх <i>Василий Терещенко, Денис Янчик, Дмитрий Пустовойтов</i>	280
Разработка инструментария для показа стерео-презентаций из файлов в KML-формате <i>Александр Бобков, Илья Казанский, Станислав Клименко</i>	285
Комплекс виртуальной имитации зубочелюстной системы <i>А.Г. Смирнов, С.В. Клименко, Д.А. Ростков</i>	292

Biometry Workshop

Алгоритм автоматического выделения лица на термографических изображениях <i>Наталья Баши, Людмила Шульга</i>	300
Алгоритмы биометрической идентификации личности человека на основе инвариантных признаков профильного изображения лица <i>Мумтозали Тухтасинов</i>	304
Fingerprint image quick processing <i>Vladimir Gudkov, Maxim Bokov</i>	308
Система идентификации дикторов по голосу для конкурса NIST SRE 2010 <i>Ю.Н. Матвеев, К.К. Симончик</i>	315
An Adaptive Algorithm of Iris Image Key Points Detection <i>Elena A. Pavelyeva, Andrey S. Krylov</i>	320
Averaging of Fingerprint Template with Respect to Elastic Deformations <i>Oleg Ushmaev, Artem Arutyunyan</i>	324

Young Scientists School

Оценивание ожидаемой работоспособности алгоритмов измерения координат объектов <i>Павел Бабаян, Сергей Смирнов</i>	329
Использование информации о цвете в алгоритме выделения лиц на изображениях <i>Л.А. Шмаглит, М.Н. Голубев</i>	331
Адаптивный метод выделения движущихся объектов <i>Рустам Салимзибаров, Артем Разумков, Александр Коробков</i>	333
Анализ использования адаптивного дискретного косинусного преобразования для улучшения качества изображений <i>О.Н. Гуцина, Е.А. Аминова</i>	335
Восстановление информации о смежности вершин для полигональных сеток, полученных с помощью метода маршрутирующих кубов <i>Ростислав Хлебников</i>	337
Сегментация отсканированных документов <i>Алексей Вилькин, Марта Егорова</i>	339
Удаление аддитивного белого гауссова шума из цифровых изображений на основе анализа главных компонент <i>Владимир Волохов, Евгений Сергеев</i>	342
Формирование и идентификация скрытых голографических водяных знаков в JPEG-изображениях <i>Владислав Гендин</i>	344
Алгоритм модификации типового трехмерного портрета по заданным фотоизображениям <i>Александра Соловьева</i>	346
Построение трехмерной модели порового пространства горных пород <i>Александр Чигорин, Антон Конушин</i>	348
Сtereo-визуализация научных и медицинских объемных данных трассировкой лучей в реальном времени <i>Николай Гаврилов, Александра Белокаменская</i>	350
Интерактивная обучающая система «Освещение в 3ds Max» <i>Татьяна Бобышева</i>	353
Разработка графического движка для компьютерной игры на GLSL <i>Александр Веселов</i>	355
Об одной реализации метода фотонных карт для визуализации каустик <i>Данила Ульянов</i>	358
Редактор для создания реалистичных физически корректных моделей эффектов для ландшафта <i>Александр Иванюк, Алексей Потапов, Алексей Щекин</i>	360
Авторский указатель	362

Dear Participant,

We would like to welcome you at GraphiCon'2010, a major international conference on Computer Graphics, Computer Vision, Image and Video processing in Russia. This anniversary 20th event will be hosted by the Saint-Petersburg State University of Information Technologies, Mechanics and Optics on September 20-24, 2010.

This year we have a strong program consisting of scientific papers, carefully selected by International Program Committee for oral presentations, young scientist school and special interest session on biometry. Keynote talks will be given by Dr. David Kirk (NVidia Fellow), Prof. Tomas Akenine-Möller (Lund University) and Prof. Marcus Gross (ETH Zurich).

The International Program Committee was formed of 56 members representing 14 countries all over the world. Being top experts in the respective areas, all of them have done a good job reviewing on average 6 papers out of 113 submitted. We express thanks to the committee members, who served at considerable personal sacrifice and with impressive collective wisdom. The final decision was based on at least two reviews of each manuscript and ended up with 48 works selected for oral presentation and 6 papers for biometry special session. Also we have decided to organize a young scientist school, where 15 papers will be presented.

Keeping traditions, this year GraphiCon has not only scientific, but also extensive educational program consisting of 2 full-day tutorials and master classes 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 also like to thank our main sponsors - Intel Corporation, NVidia and Microsoft, volunteers organized by Graphics and Media Lab, LMSU, and everyone who made this event happen.

We do hope you will enjoy the conference,
Anton Konushin, on behalf of GraphiCon'2010 Organizing Committee



Technical Section

GraphiCon'2010

September 20-24, 2010
Saint-Petersburg, Russia

English Section

Adaptive Data Hiding: A Hybrid Based High-capacity Approach for 3D Models

Shih-Chun Tu, Charlie Irawan Tan and Wen-Kai Tai
 Department of Computer Science and Information Engineering
 National Dong Hwa University, Taiwan, Republic of China.
 tusjtu@ms01.dahan.edu.tw, d9421003@ems.ndhu.edu.tw, wktai@mail.ndhu.edu.tw

Abstract

A hybrid data hiding approach, combining permutation steganography and the spatial domain approach, is proposed in this paper. Message is partially embedded in the cover model by permutation steganography first, and then the rest of message is embedded in the vertex by modifying vertex position limited in a voxel. The number of bits to be embedded in the vertex is adaptive in light of the degree of model distortion and visual perception to the stego model preferred.

As compared with previous data hiding methods for 3D models, our capacity is up to twice as large as that of previous work. Also, with the adaptability, the proposed hybrid approach is flexible with a tradeoff between capacity and distortion. Both embedding and extraction procedures are simple to implement and running efficiently.

Keywords: *data hiding, spatial domain, permutation steganography.*

1. INTRODUCTION

Steganography, the art of hiding one message in another, has been used over many centuries. Steganography ranges from simple hidden messages that can be deciphered by shifting each letter by a number of positions in the alphabet to watermarks that can be extracted from an image by a specific method, and has been applied widely in many topics related to information security. Steganography has even been used recently in 3D data, with similar purposes to those of 2D data.

In this paper, we propose a hybrid approach for data hiding based on permutation steganography and the vertex modification in the spatial domain. The 3D model consists of the vertex set and face set. The permutation steganography by Tu et al. [1] is applied to embed message in the vertex and face. Namely, the vertex permutation and the face permutation represent the embedded message. Then, we modify the position of the vertex in the vertex permutation within a range inside a predefined voxel to further embed more bits in each component of the vertex. The axis aligned bounding volume composed of voxels is built for the cover model. The voxel is subdivided into a number of units in each dimension as well. The unit where a vertex is located at inside the voxel indicates the embedded message bitstream for each dimension. Eventually, one part of the message is embedded in the first embedding stage, and another in the second. When extracting the message, either the extraction procedure of permutation steganography [1], or modifying vertex position can be applied first. Message extracted by the procedure of modifying vertex position is the second part of embedding message, and, of course, by that of permutation steganography is the first part.

The permutation steganography is distortionless for the stego model, but modifying vertex position causes model distortion related to the number of bits to be embedded in and the scale of

the cover model. Therefore, the capacity of our hybrid approach is adaptive by trading distortion for capacity. Model distortion is measured using normalized Hausdorff distance and visual perception. As experimental results show, our capacity is the highest than previous work of data hiding. Also, our approach is simple to implement with time complexity $O(n)$.

The rest of this paper is structured as follows. Section 2. surveys related work. Section 3. describes the proposed method. Experimental results are shown in Section 4.. Finally, we conclude and point out possible future work in Section 5..

2. RELATED WORK

Steganography for 3D polygonal meshes was pioneered by Ohbuchi et al. [2], who introduced watermarking on 3D polygonal meshes. Since then many ideas to steganography have been proposed. Most methods are to slightly perturb the vertex positions of the mesh for hiding messages either in spatial domain [3, 4, 5, 6, 7, 8, 9, 2, 10, 11, 12, 13, 14, 15, 16], or in the spectral domain, [17, 18, 19, 20, 21, 22]. Spatial methods tend to have higher capacity and lower computation costs at the expense of weak robustness. Spectral methods are more robust but have limited capacity and involve serious computations. They are more appropriate for data protection applications, such as watermarking, than for data hiding. Recently, Chao et al. [23] presented a very high-capacity and low-distortion 3D steganography approach based on a novel multilayered embedding scheme to hide secret messages in the vertices of 3D polygon models. Their approach can hide 21 to 39 bits/vertex.

Some methods, [17, 23, 20, 13], utilize 3D models defined as point set. Polygonal meshes provide fewer vertices than point set models, but have face information that can be used as the alternative medium. Higher vertex numbers allow a model to hide more information, but require more space and computing power to handle. Recently several methods, [6, 11], hide messages in the connectivity of the mesh by rearranging vertices and faces relative to a reference ordering derived from the mesh geometry. Their techniques are lossless because the cover and stego models are the same.

Permutation steganography [24, 25, 26, 27] gives optimal capacity for hiding information through reordering of n primitives that have a known reference ordering. Permutation steganography is of the optimal capacity, up to $O(\log(n!)) = O(n \log n)$ bits, which is much better than the results of the previous work for 3D polygonal meshes but at the expense of computation time $\Omega(n^2 \log^2 n \log \log n)$. Recently, two proposed methods, Bogomjakov et al. [28] and Tu et al. [1], are very simple to implement and perform efficiently, $O(n)$. Both methods guarantee the minimal capacity, one bit per element less than the theoretical optimum, and are robust and resistant to any kind of attacks on the polygonal mesh because the reference ordering is obtained by using the traversal of Edgebreaker mesh compression algorithm [29] based on the mesh connectivity alone. Obviously, those approaches are lossless.

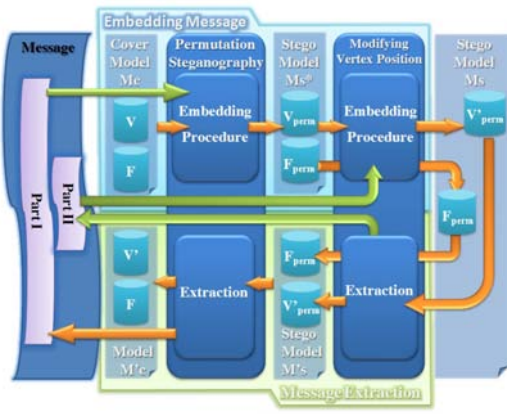


Figure 1: The framework of hybrid data hiding approach.

3. PROPOSED APPROACH

The framework of our approach is shown in Figure 1. There are two stages both for message embedding and extraction procedures. When embedding messages for the cover model, we first apply permutation steganography [1], to hide message, and then modify the rearranged embedding vertices to hide more message. To extract message from the stego model, either the extraction procedure of [1] or modifying vertex position can be first used to extract message, and then another follows to extract message left.

3.1 Embedding Message

3.1.1 Embedding by Permutation Steganography

Permutation steganography hides the message in a cover model by rearranging the order of vertices in the model with respect to a canonical reference ordering. We apply the method proposed by Tu et al. [1], which improves the work of Bogomjakov et al. [28], to hide the message in the first stage of the embedding procedure.

Given a cover model, $M_c = (V, F)$, where V is the set of vertices and F is the set of faces, the first edge of the first face in the model is selected as the initial vertex and edge to obtain two reference orderings by using Edgebreaker algorithm [29] respectively. Note that the binary trie [30] search structure is built for the embedding primitive, in which the internal node branches the search traversal by the message bitstream and the leaf node keeps the primitive each indexed in a non-decreasing order from left to right. Figure 2 shows an example of binary trie with $n = 11$ embedding primitives at the leaf node (red) where each is indexed and associated with the corresponding embedded bitstream shown below the index.

At each step i , a primitive at position p is chosen from the remaining $n - i$ primitives of the reference ordering and output it as the next primitive of the permutation. The position p is the index of the leaf node reached by the binary trie traversal according to the next $k + 1$ bits, $k = \lfloor \log_2(n - i) \rfloor$, in the embedding message. The binary trie is a complete binary tree so the leaf node is either at level $\lceil \log_2(n - i) \rceil$ or $\lceil \log_2(n - i) \rceil - 1$. If the number of leaf nodes at the highest level, $e = ((n - i) - 2^k) \times 2$, is larger than the integer value of the next $k + 1$ bits, then the primitive at position p will be outputted as the next primitive of the permutation. Otherwise, the primitive at level $\lceil \log_2(n - i) \rceil - 1$ reached by next k bits in the embedding message will be outputted. Note that the output primitive is actually removed by replacing it with the last primitive in the remaining primitive so that the remaining $n - i - 1$ primitives

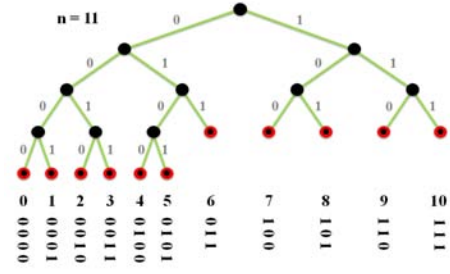


Figure 2: An example of the binary trie search structure with 11 embedding primitives at the leaf node (red). The index and the embedded bitstream are shown below the leaf node.

can be still indexed sequentially.

After this first stage of embedding procedure, we have the quasi stego model $M_{s^*} = (V_{perm}, F_{perm})$.

3.1.2 Embedding by Modifying Vertex Position (MVP)

The vertex in the output primitive permutation V_{perm} can be embedded in more message bitstream by modifying its position.

The axis aligned bounding volume of M_{s^*} is determined. Given k_x, k_y , and k_z bits to be embedded in x, y , and z components of the vertex respectively, the volume is then subdivided into $n_{voxel}^x \times n_{voxel}^y \times n_{voxel}^z$ voxels, where $n_{voxel}^i = l_{BV}^i / (2^{k_i} \times l_v^i)$, l_{BV}^i is the side length of the bounding volume, and l_v^i is the unit length of 2^{k_i} units in the voxel for $i = x, y$ and z . The next k_i bits in the embedding message are embedded in the vertex by modifying the coordinate of its i component to the unit with the index equal to the integer value of k_i in the component of the belonging voxel for all $i = x, y$ and z . Figure 3 illustrates the voxelized bounding volume and the subdivision for a voxel. The embedding procedure of MVP is summarized in Algorithm 4.

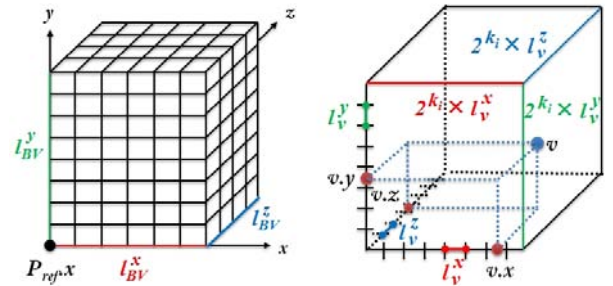


Figure 3: The axis aligned bounding volume of the cover model is conceptually subdivided into voxels, and the voxel is again subdivided into 2^{k_i} units for each component x, y , and z . The vertex v is embedded in k_x, k_y , and k_z bits in x, y , and z components respectively.

3.2 Message Extraction

There are two stages of message extraction. First, the message extraction procedure of permutation steganography [1] is to extract the message. In the second stage, we extract the message which is embedded by MVP. Note that either stage can be used to extract message first, and then another.

```

Input: perm[]
Output: output[]
// perm[] is the output primitive permutation from permutation
// steganography.
// p_ref is the minimal vertex of the bounding volume.
// n is the number of the vertices.
// peek(k_i) peeks next k_i bits from the embedding message.
for i = 0 to n - 1
    v = perm[i]
    output[i].x ← p_ref.x + ⌊  $\frac{v.x - p_{ref}.x}{2^{k_x} \times l_v^x}$  ⌋ × (2^{k_x} × l_v^x)
                    + INT(peek(k_x)) × l_v^x;
    output[i].y ← p_ref.y + ⌊  $\frac{v.y - p_{ref}.y}{2^{k_y} \times l_v^y}$  ⌋ × (2^{k_y} × l_v^y)
                    + INT(peek(k_y)) × l_v^y;
    output[i].z ← p_ref.z + ⌊  $\frac{v.z - p_{ref}.z}{2^{k_z} \times l_v^z}$  ⌋ × (2^{k_z} × l_v^z)
                    + INT(peek(k_z)) × l_v^z;
end

```

Figure 4: Pseudo code of the embedding procedure of *MVP*.

3.2.1 Extraction by Permutation Steganography

The message extraction procedure of [1], is applied to extract the message in the stego model. Given a permutation of n primitives in the stego model, again the same reference ordering as one from the cover model is computed. And, we build the binary trie search structure for the primitives in the reference ordering. At each step i , we choose the next primitive from the primitive permutation to extract the message bitstream by finding the position (index) p where the primitive is located in the remaining $n - i$ primitives of the reference ordering. If the number of primitive (leaf) nodes in the binary trie at the highest level, $e = ((n - i) - 2^k) \times 2$, at step i is larger than p , then p represents the extracted $k + 1 = \lfloor \log_2(n - i) \rfloor + 1$ bits. Otherwise, one of the primitives at level $\lfloor \log_2(n - i) \rfloor - 1$ is the candidate for extracting next k bits. In this case, p is updated as $p - e/2$ to represent the extracted k bits.

3.2.2 Extraction by Modifying Vertex Position (MVP)

In this stage, we extract the embedded message from the vertices in the stego model. Actually, for each component x , y , and z , the index of the unit in the voxel where the vertex is located at is the integer value of the embedded message. Given the maximal embedding bits, k_i , the minimal vertex of the bounding volume of the cover model, p_{ref} , and the unit length, l_v^i , in the voxel, the embedding message m_i can be extracted by

$$m_i = \frac{v.i - p_{ref}.i - \lfloor \frac{v.i - p_{ref}.i}{2^{k_i} \times l_v^i} \rfloor \times (2^{k_i} \times l_v^i)}{l_v^i}, i = x, y, z.$$

4. EXPERIMENTAL RESULTS

All experiments were performed with several polygonal models of different sizes on a PC with an Intel Core 2 1.87GHz processor and 2GB main memory to verify and evaluate our proposed approach. In all experiments, unless otherwise specified, the decimal precision for the vertex coordinate of all testing models is about 6 decimal digits so the unit length l_v^i is set to 1×10^{-6} , and $k_x = k_y = k_z = k$ bits.

Over 1000 randomly generated embedding messages are used to measure the average capacity and normalized Hausdorff distance of the testing models for permutation steganography [1], and *MVP* respectively. The statistics of the measured capacity is shown in Table 1. For permutation steganography, the average capacity of Bogomjakov et al. [28] is the highest, nearly optimum $\log_2 n!$, than

that of previous work. The average capacity improved by Tu et al. [1] is about 0.63 bits/vertex. For spatial domain, Cheng et al. [6] proposed a multilevel embedding procedure and a 3D model representation rearrangement procedure to hide 9 bits/vertex. Chao et al. [23] proposed a novel multilayered embedding scheme that can hide up to $3n_{layers}$ bits/vertex in normalized models, where n_{layers} ranges from 7 to 13. The proposed *MVP* approach can hide 27 to 48 bits/vertex in 3D models. The capacity of *MVP* is much higher than that of Cheng et al. [6] in 3D models, but is less than that of Chao et al. [23] in normalized 3D models about 10 bits/vertex. However, as you can see, Our hybrid data hiding approach produces much higher hiding capacity, $\approx 2\log_2 n!$, than all of the previous work.

The visual perception for the cover model and stego models of the all testing models were shown from Figure 7 to 12 in that subfigures (b) and (c) show the stego models with different k . Subfigure (b) shows that the stego model with the maximal k bits embedded by using *MVP* has little distortion yet is almost unperceivable visually. As a model distortion comparison, subfigure (c) shows the stego model with distortion that can be easily detected when one likes to trade model distortion for embedding more bits. Subfigures (d), (e) and (f) show the close-up views of the subfigures right on the top. Regarding to the visual perception, subfigure (e) is similar to subfigure (d), but the distortion seen in subfigure (f) is manifest.

The number of bits to be embedded in the vertex at the second stage of message embedding is adaptive. But the more bits to be embedded in the vertex the more distortion to the stego model. The normalized Hausdorff distance [31] is commonly used to measure the model distortion such as Metro [32], M.E.S.H. [33], Cheng et al. [6], etc. We measure the average normalized Hausdorff distance (*NHD*) for all testing models and obtain the reasonable *NDH* each for the testing model as shown in Table 1. As experiments show, the distortion is visually acceptable when the value of *NHD* is around 1×10^{-4} . Figure 5 shows the normalized Hausdorff distance as a function of embedding k bits in each component of the vertex by *MVP*. Note that different model scale presents different sensitivity of model distortion to the increasing of embedding bits. The Armadillo is a large scale model and is of acceptable distortion when k is up to 16 resulting totally high capacity 97.71 bpv. Figure 6 illustrates the adaptability using Armadillo model as an example. Our approach adapts the capacity by the model distortion preferred.

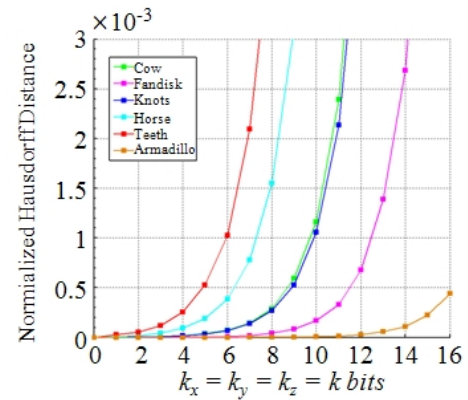


Figure 5: The average normalized Hausdorff distance as a function of embedding k bits in each component of the vertex by *MVP*.

Table 2 shows timing statistics in milliseconds of the embedding and extraction procedures of permutation steganography [1], and *MVP* for testing models respectively. The time complexity of *MVP*

Model			Capacity (bpv) [bits]				normalized
name	#verts	#faces	Tu et al. [1]	<i>MVP</i>	k	bpv	Hausdorff distance
Cow	2,904	5,804	92,918 (32.00)	87,120 (27.00)	9	57.00	5.92×10^{-4}
Fandisk	6,475	12,946	229,686 (35.47)	233,100 (36.00)	12	71.47	6.79×10^{-4}
Knots	23,232	46,464	952,798 (41.01)	627,264 (27.00)	9	68.01	5.26×10^{-4}
Horse	48,485	96,966	2,142,566 (44.19)	872,712 (18.00)	6	62.19	3.87×10^{-4}
Teeth	116,604	233,204	5,595,303 (47.99)	1,749,060 (15.00)	5	62.99	5.25×10^{-4}
Armadillo	172,974	345,944	8,597,908 (49.71)	8,302,752 (48.00)	16	97.71	4.42×10^{-4}

Table 1: The statistics of measured average capacity and normalized Hausdorff distance for testing models. The number in column k means that the model distortion is still visually unperceivable after k bits are embedded in each component of the vertex, meanwhile the normalized Hausdorff distance measured is shown in the last column for each model.

Model			Timings (msecs)				Total
			Embedding		Extraction		
name	#verts	#faces	Tu et al. [1]	<i>MVP</i>	Tu et al. [1]	<i>MVP</i>	
cow	2,904	5,804	6	1	3	$\ll 1$	10
fandisk	6,475	12,946	13	1	9	1	24
knots	23,232	46,464	56	4	39	2	101
horse	48,485	96,966	124	9	87	5	225
teeth	116,604	233,204	311	27	225	17	580
armadillo	172,974	345,944	481	40	341	24	886

Table 2: Timing statistics for proposed embedding and extraction procedures of *MVP* and permutation steganography [1], for testing models.

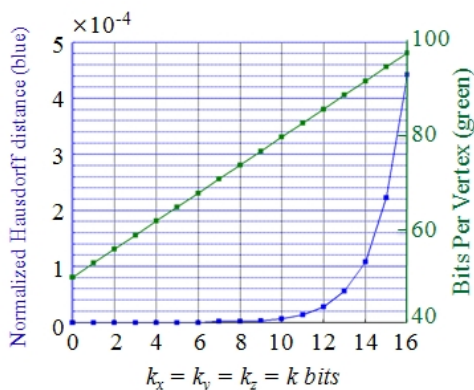


Figure 6: Adaptability illustration for Armadillo model. Given a preferred *NHD*, the maximal k bits to be embedded is adaptively determined.

and permutation steganography [1], are both $O(n)$. Namely, the proposed hybrid approach is $O(n)$. Note that even for Armadillo model, the embedding and extraction procedures can be done in one second.

5. CONCLUSION AND FUTURE WORK

A hybrid data hiding approach has proposed, which combines permutation steganography and *MVP*. Embedding message by permutation steganography [1], for the vertex and face in the cover model and then modifying the vertex position by *MVP*, our method improves the capacity of data hiding on 3D models up to $2\log_2 n!$. Moreover, the capacity is adaptive regarding to the degree of model distortion making our method flexible. Our method is simple to implement and is efficient, $O(n)$, running in a second for all testing models.

The distortionless approach with high capacity is one of the key

concerns for data hiding. In the future, it is worth exploring what optimal number of units in each component of the voxel is to make the model distortion less. Also, we like to embed more permutations in the cover model. Namely, the primitive arrangement represents more primitive permutations, hopefully, $\log_2 n!$, $s \geq 2$. *MVP* is not a robust approach. To improve the robustness, we would try to find a way for modifying vertex position on the basis of reference ordering.

6. REFERENCES

- [1] Shih-Chun Tu, Wen-Kai Tai, Martin Isenburg, and Chin-Chen Chang, "An improved data hiding approach for polygon meshes," *The Visual Computer*, 2009, DOI: 10.1007/s00371-009-0398-1.
- [2] R. Ohbuchi, H. Masuda, and M. Aono, "Watermarking three-dimensional polygonal models," in *MULTIMEDIA'97: Proceedings of the fifth ACM international conference on Multimedia*, 1997, pp. 261–272.
- [3] Nicolas Aspert, E. Drelie, Y. Maret, and Touradj Ebrahimi, "Steganography for three-dimensional polygonal meshes," in *Proceedings of SPIE*, Nov. 2002, vol. 4790, pp. 211–219.
- [4] Oliver Benedens, "Geometry-based watermarking of 3d models," *IEEE Computer Graphics and Applications*, vol. 19, no. 1, pp. 46–55, 1999.
- [5] F. Cayre and B. Macq, "Data hiding on 3-d triangle meshes," *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 939–949, April 2003.
- [6] Yu-Ming Cheng and Chung-Ming Wang, "A high-capacity steganographic approach for 3d polygonal meshes," *The Visual Computer*, vol. 22, no. 9, pp. 845–855, 2006.
- [7] Yu-Ming Cheng and Chung-Ming Wang, "An adaptive steganographic algorithm for 3d polygonal meshes," *The Visual Computer*, vol. 23, no. 9-11, pp. 721–732, 2007.

- [8] Jae-Won Cho, Rémy Prost, and Ho-Youl Jung, “An oblivious watermarking for 3-d polygonal meshes using distribution of vertex norms,” *IEEE Transactions on Signal Processing*, vol. 55, no. 1, pp. 142–155, 2007.
- [9] Y. Maret and T. Ebrahimi, “Data hiding on 3d polygonal meshes,” in *MM&Sec’04: Proceedings of the ACM 2004 workshop on Multimedia and security*, 2004, pp. 68–74.
- [10] Emil Praun, Hugues Hoppe, and Adam Finkelstein, “Robust mesh watermarking,” in *Siggraph 1999, Computer Graphics Proceedings*, 1999, pp. 49–56.
- [11] C.-M. Wang and Y.-M. Cheng, “An efficient information hiding algorithm for polygon models,” *Computer Graphics Forum*, vol. 24, no. 3, pp. 591–600, 2005.
- [12] C.-M. Wang and P.-C. Wang, “Data hiding approach for point-sampled geometry,” *IEICE Transactions on Communications*, vol. E88-B, no. 1, pp. 190–194, 2005.
- [13] C.-M. Wang and P.-C. Wang, “Steganography on point-sampled geometry,” *Computers & Graphics*, vol. 30, no. 2, pp. 244–254, 2006.
- [14] H.-T. Wu and Y.-M. Cheung, “A reversible data hiding approach to mesh authentication,” in *WI’05: the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, 2005, pp. 774–777.
- [15] Z. Yu, H. S. IP Horace, and L. F. Kwok, “A robust watermarking scheme for 3d triangular mesh models,” *Pattern Recognition*, vol. 36, no. 11, pp. 2603–2614, 2003.
- [16] S. Zafeiriou, A. Tefas, and I. Pitas, “Blind robust watermarking schemes for copyright protection of 3d mesh objects,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 5, pp. 596–607, 2005.
- [17] D. Cotting, T. Weyrich, M. Pauly, and M. Gross, “Robust watermarking of polygonal meshes,” in *SMI’04: International Conference on Shape Modeling and Applications*, 2004, pp. 233–242.
- [18] S. Kanai, H. Date, and T. Kishinami, “Digital watermarking for 3d polygons using multiresolution wavelet decomposition,” in *Proceedings of Sixth IFIP WG 5.2 GEO-6*, 1998, pp. 296–307.
- [19] R. Ohbuchi, A. Mukaiyama, and S. Takahashi, “A frequency domain approach to watermarking 3d shapes,” *Computer Graphics Forum*, vol. 21, no. 3, pp. 373–382, Sept. 2002.
- [20] R. Ohbuchi, A. Mukaiyama, and S. Takahashi, “Watermarking a 3d shape model defined as a point set,” in *CW’04: Third International Conference on Cyberworlds*, 2004, pp. 392–399.
- [21] R. Ohbuchi, S. Takahashi, T. Miyazawa, and A. Mukaiyama, “Watermarking 3d polygonal meshes in the mesh spectral domain,” in *GRIN’01: No description on Graphics interface 2001*, 2001, pp. 9–17.
- [22] M. G. Wagner, “Robust watermarking of polygonal meshes,” in *Proceedings of Geometric Modeling and Processing*, 2000, pp. 201–208.
- [23] Min-Wen Chao, Chao-Hung Lin, Cheng-Wei Yu, and Tong-Yee Lee, “A high capacity 3d steganography algorithm,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 2, pp. 274–284, March 2009.
- [24] Stephen Forrest, “Html steganography tool,” <http://wandership.ca/projects/deogol/intro.html>.
- [25] David Glaude, “Hiding data into the palette of a gif,” <http://users.skynet.be/glu/sgpo.htm>.
- [26] M. Kwan, “Gif colormap steganography,” <http://www.darksided.com.au/gifshuffle/>.
- [27] Donovan Artz, “Digital steganography: Hiding data within data,” *IEEE Internet Computing*, vol. 5, no. 3, pp. 75–80, 2001.
- [28] Alexander Bogomjakov, Craig Gotsman, and Martin Isen-burg, “Distortion-free steganography for polygonal meshes,” *Computer Graphics Forum*, vol. 27, no. 2, pp. 637–642, 2008.
- [29] Jarek Rossignac, “Edgebreaker: Connectivity compression for triangle meshes,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 1, pp. 47–61, 1999.
- [30] Ellis Horowitz, Sartaj Sahni, and Dinesh Mehta, *Fundamentals of Data Structures in C++*, Silicon Press, 2006.
- [31] Normand Gregoire and Mikael Bouillot, “Hausdorff distance between convex polygons,” <http://cgm.cs.mcgill.ca/~godfried/teaching/cg-projects/98/normand/main.html>.
- [32] Paolo Cignoni, Claudio Rocchini, and Roberto Scopigno, “Metro: measuring error on simplified surfaces,” *Computer Graphics Forum*, vol. 17, no. 2, pp. 167–174, June 1998.
- [33] Nicolas Aspert, Diego Santa-Cruz, and Touradj Ebrahimi, “Mesh: Measuring error between surfaces using the hausdorff distance,” in *IEEE International Conference on Multimedia and Expo Proceedings*, 2002, pp. 705–708, <http://mesh.berlios.de/>.

ABOUT THE AUTHOR

Shih-Chun Tu is currently a doctoral student of Computer Science and Information Engineering department at National Dong Hwa University, Taiwan, R.O.C.

Charlie Irawan Tan is currently a doctoral student of Computer Science and Information Engineering department at National Dong Hwa University, Taiwan, R.O.C.

Wen-Kai Tai is currently an associate professor of Computer Science and Information Engineering department at National Dong Hwa University, Taiwan, R.O.C. His current research interests include data hiding, real-time rendering, visibility, and techniques for gaming.

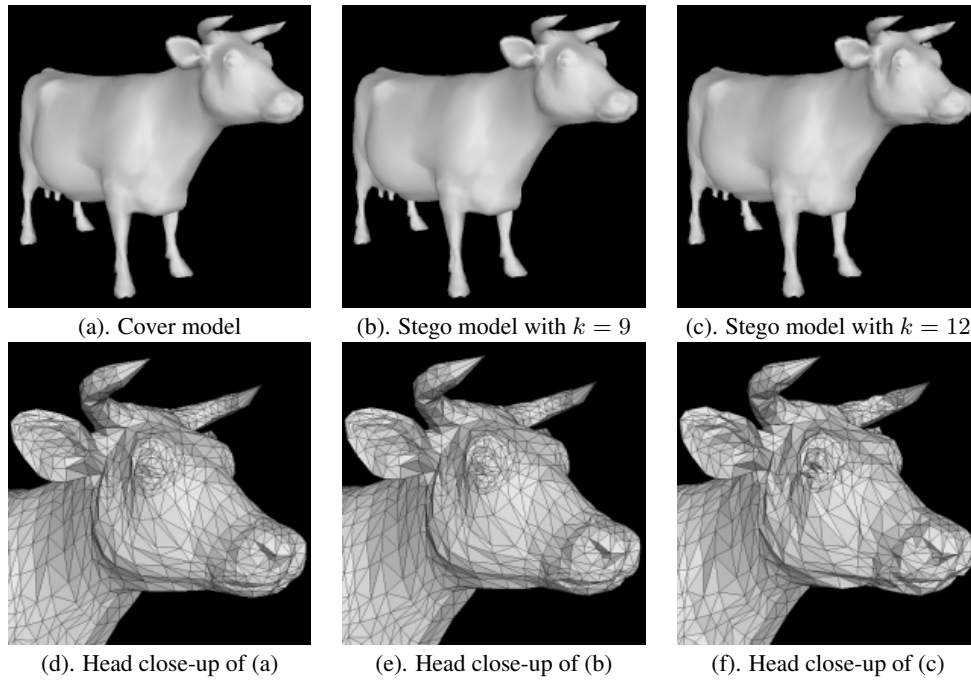


Figure 7: The visual perception for the cover model (a), Cow, its stego models for different value of k ((b) and (c)), and three close-up views (bottom) at head regarding to the figure on the top.

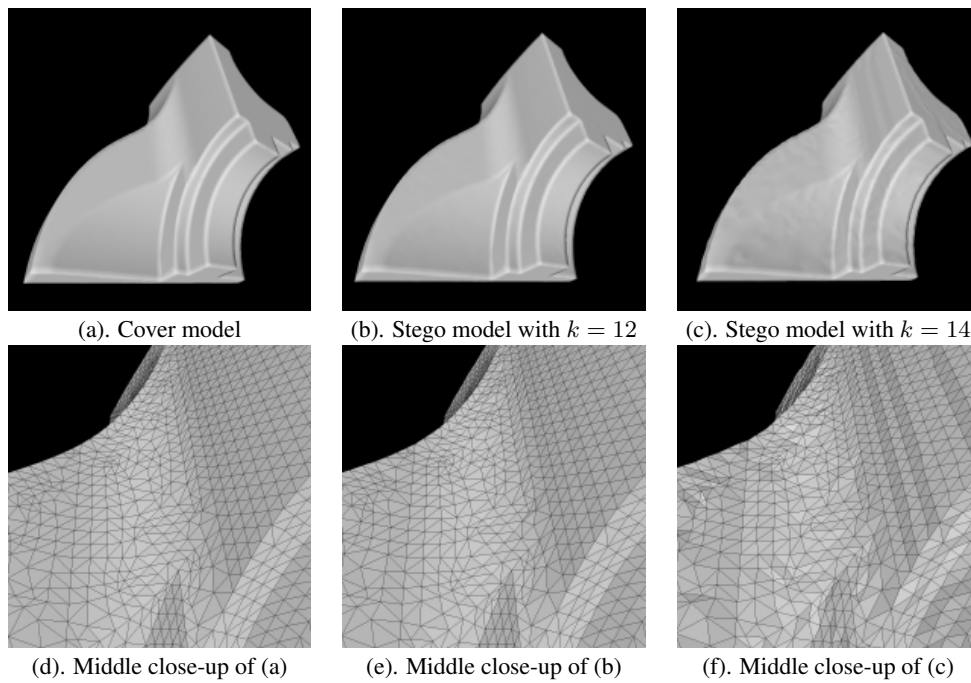


Figure 8: The visual perception for the cover model (a), Fandisk, its stego models for different value of k ((b) and (c)), and three close-up views (bottom) at middle part regarding to the figure on the top.

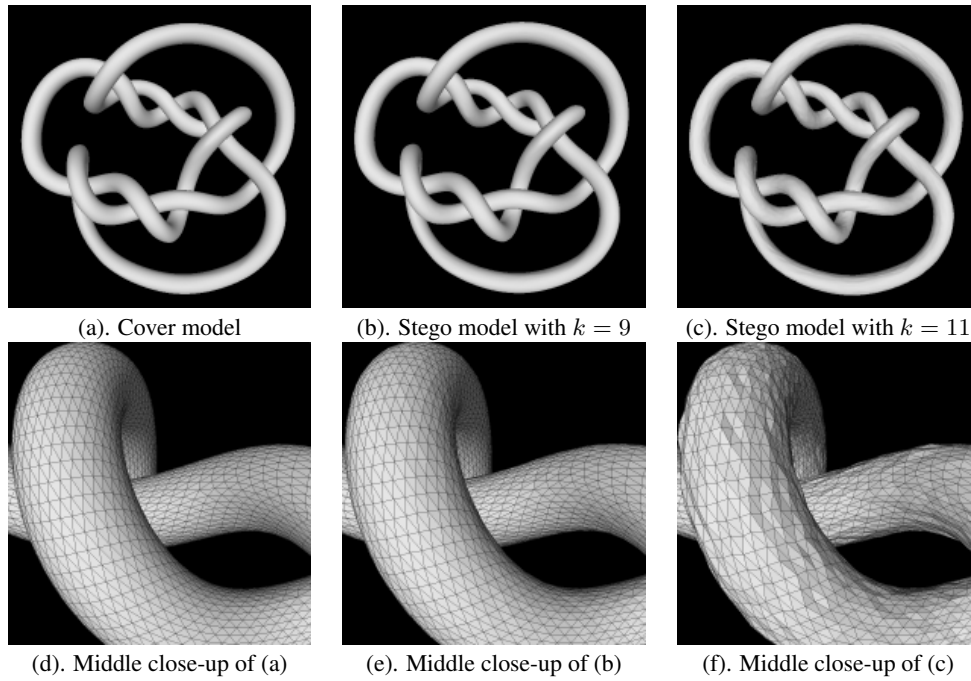


Figure 9: The visual perception for the cover model (a), Knots, its stego models for different value of k ((b) and (c)), and three close-up views (bottom) at middle part regarding to the figure on the top.

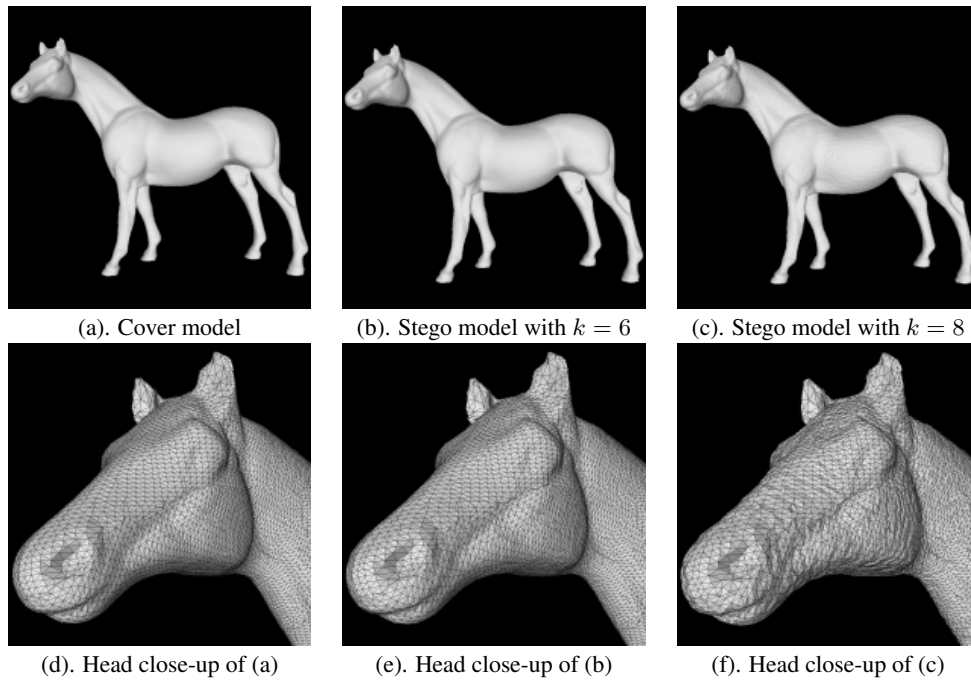


Figure 10: The visual perception for the cover model (a), Horse, its stego models for different value of k ((b) and (c)), and three close-up views (bottom) at head regarding to the figure on the top.

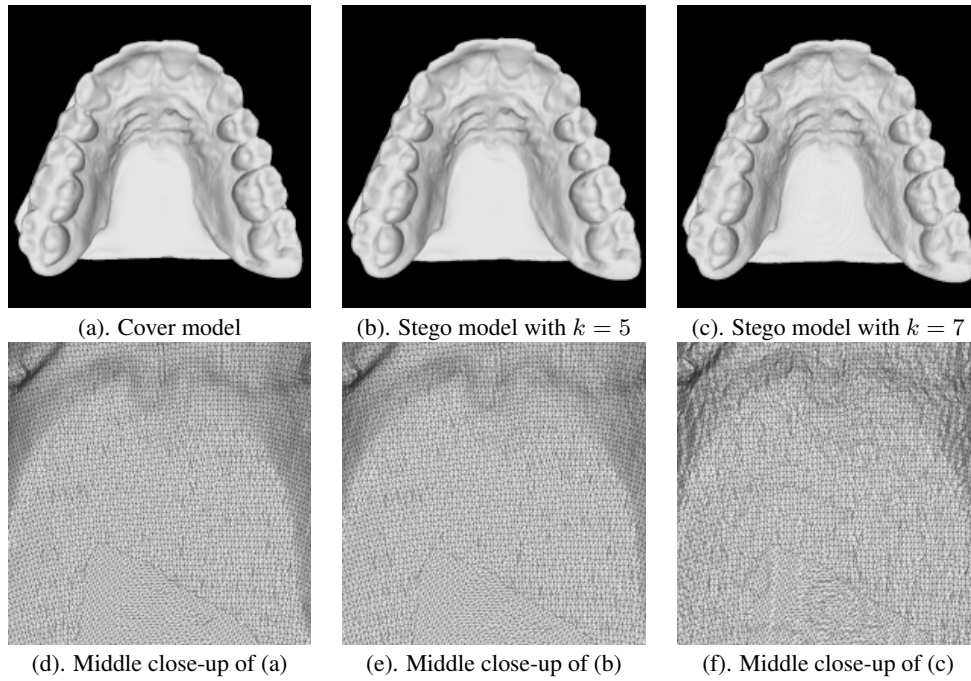


Figure 11: The visual perception for the cover model (a), Teeth, its stego models for different value of k ((b) and (c)), and three close-up views (bottom) at middle part regarding to the figure on the top.

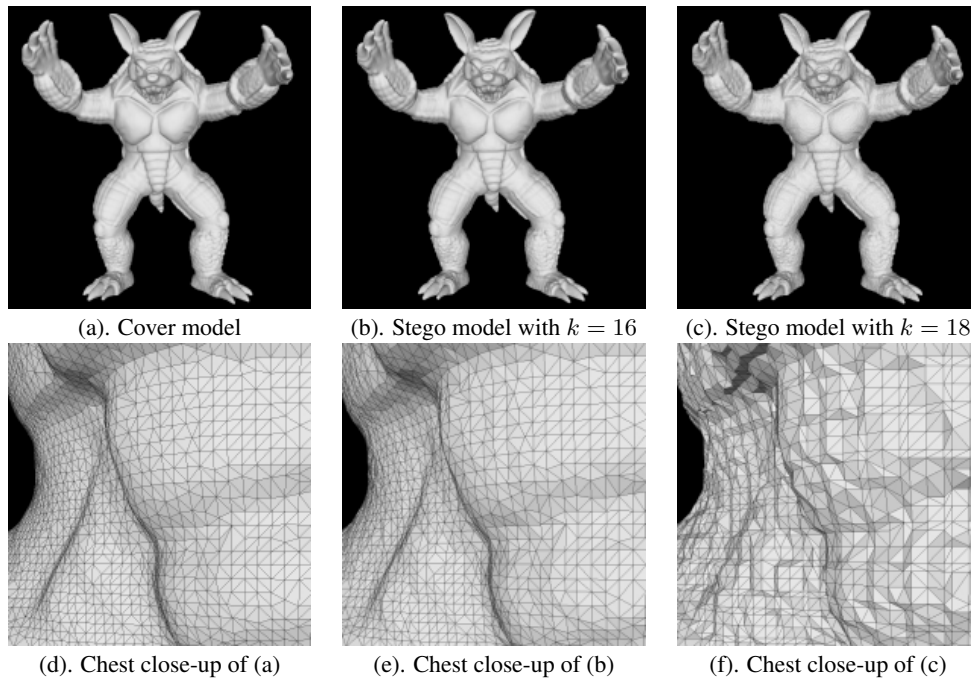


Figure 12: The visual perception for the cover model (a), Armadillo, its stego models for different value of k ((b) and (c)), and three close-up views (bottom) at chest regarding to the figure on the top.

Adaptive Context Modeling for Efficient Image and Elevation Data Compression

Egor Yusov

Intel Corporation

Nizhny Novgorod, Russia

egor.yusov@gmail.com

Abstract

In this paper, a new wavelet zero tree-based image compression algorithm that is based on exploitation of smart adaptive context models is proposed. The models are derived from wavelet transform properties and are exploited to improve efficiency of arithmetic coding. They are intuitively clear and do not require any preliminary training. To author's knowledge, the proposed algorithm is comparable to or surpasses all previous zero-tree based encoders in terms of R-D performance. At the same time, the computational complexity of the algorithm remains low because it bypasses bit-plane coding and processes each coefficient in only one pass. The near-lossless algorithm extension that is based on lossy plus residual coding and provides a guaranteed maximum absolute error bound is presented. A new hierarchical compressed multiresolution terrain model designed for efficient elevation data storage and retrieval that exploits the presented compression technique is proposed. Special care is taken to guarantee seamless stitching of neighboring patches.

Keywords: *Compression, Wavelet Transform, Context Modeling, Terrain.*

1. INTRODUCTION

Rapid evolution of digital data acquisition technologies in the past years led to the exponential growth of digital content size. As a result, efficient compression techniques that reduce the storage requirements with no or minimal information loss have becoming more and more important. Image compression applications are well known and include digital camera, medical imaging, internet browsing, to name a few.

Compact representation of digital elevation data sets is another area where efficient compression techniques are required. This problem is especially important for such applications as geographical information systems, flight simulators, virtual environments, computer games etc. Satellite scans of a terrain region can contain billions elevation samples potentially requiring storage of up to terra bytes. Processing such huge uncompressed data sets is a very complex task, especially in the context of real-time terrain visualization, because the data size can exceed not only the main memory, but even the disk capacity.

In this paper, we present a new image compression method that improves the compression performance of previous algorithms such as SPIHT [SP96] and LTW [OM03] by using smart adaptive context modeling for more efficient arithmetic coding. At the same time, since algorithm is zero-tree based and bypasses bit-plane coding, its temporal complexity remains low. We applied the presented compression technique to construct hierarchical compressed multiresolution terrain representation that can be exploited in real-time terrain rendering applications. Special care is taken to assure seamless patch connection across borders.

2. RELATED WORKS

2.1 Wavelet-based image compression methods

During last years, wavelet-based image compression techniques have becoming more and more popular since they provide better compression performance and at the same time do not suffer from artifacts typical to the Discrete Cosine Transform – based image compression algorithms. All wavelet-based image coders have the same workflow. To remove correlation between neighboring pixels, at the first stage of the compression process, the image is transformed from spatial domain to a combined spatial-frequency wavelet domain [ABM92]. After the first step of wavelet decomposition, the image is transformed into the lower resolution representation (LL subband) and three detail subbands called: horizontal (LH), vertical (HL) and diagonal (LL) details (fig. 1). The same transform is further applied to the LL subband and so on.

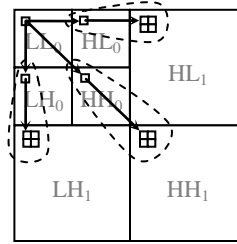


Fig. 1: Subbands of two-stage wavelet decomposition.

We denote $s_{i,j}^{(l)}$ to be the coefficient of subband s where $s \in \{ll, lh, hl, hh\}$ of the level l located at spatial position (i,j) . In the literature, coefficient $s_{i,j}^{(l)}$ is called significant with respect to threshold τ , if its magnitude exceeds τ , i.e. $|s_{i,j}^{(l)}| \geq \tau$ and it is called insignificant otherwise.

Wavelet coefficients resulting from full wavelet decomposition are quantized at the next stage (here some information is lost) and are encoded. Due to hierarchical nature of wavelet transform, quantized wavelet coefficients can be organized into three quad trees growing through the LH, HL and HH subbands (see fig. 2).

The fundamentally new method for wavelet coefficient trees encoding was presented by J.Shapiro [Sha93] in his EZW algorithm. The main contribution of [Sha93] is introduction of a zero tree structure. Wavelet coefficients tree is called *zero tree* with respect to threshold τ if all its nodes are insignificant with respect to τ . If we define $T_{i,j}^{(l)} = \{(i, j, l_r)\}$ to be the set of positions of all nodes belonging to the tree rooted at node (i, j) on level l , including the root, then the node $s_{i,j}^{(l)}$ ($s \in \{lh, hl, hh\}$) is called the *zero*

tree root with respect to threshold τ , if it does not yet belong to the zero tree and $\forall (i_r, j_r, l_r) \in T_{i,j}^{(l)} \rightarrow |s_{i_r, j_r}^h| < \tau$.

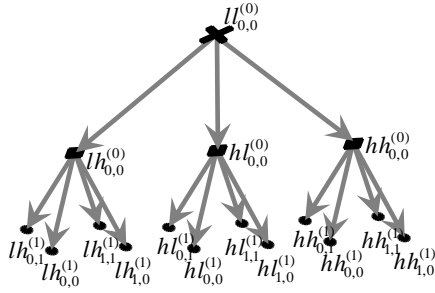


Fig. 2: Wavelet coefficients quad trees.

The main idea behind zero tree coding is the observation that if a coefficient is insignificant at a coarse scale in a multiresolution representation, then it is likely that its descendants at the finer scales are also insignificant. Thus single symbol is enough to encode all zero coefficients in a zero tree. One important feature of EZW algorithm is that it generates embedded bit stream, in which all encodings of the same image at lower bit rates are embedded in the beginning of the bit stream at higher bit rate.

EZW laid down the foundations for the new class of wavelet-based image compression techniques – the zero tree coders. The next method in this trend is called SPIHT and was presented by A. Said and W. Pearlman [SP96]. SPIHT is a highly refined version of EZW and due to smart set partitioning rules exploited it achieves remarkably higher compression ratios compared to EZW.

Though code embedding is a very useful feature, it significantly complicates the algorithm since each coefficients is reconstructed in several passes, one for each bit plane. However many applications (such as the digital camera or elevation data storage) do not require that feature and the data need to have pre-defined quality. So a number of algorithms tried to give up quality scalability in favor of higher execution speed.

One of such methods named SWEET was presented by J. Andrew [And97]. SWEET is solely based on energy clustering in transformed image in frequency and space and exploits block partitioning as an alternative to zero-tree coding to separate significant coefficients from large sets of insignificant ones. In contrast to SPIHT and EZW, SWEET does not produce embedded code: it outputs all bits of the coefficient magnitude up to some minimum bit plane number n_{min} as well as its sign as soon as the coefficient is identified as significant. Since SWEET avoids complicated list processing, it is much simpler and faster than SPIHT and at the same time it demonstrates comparable compression efficiency.

To improve temporal performance of the wavelet encoders/decoders, J. Oliver and M. Malumbres [OM03] proposed the algorithm that is based on a structure called wavelet coefficients lower tree (LTW), which is actually the zero-tree of pre-quantized wavelet coefficients with respect to threshold $2^{rplanes}$ ($rplanes$ is the number of least significant bit planes to drop). The main speed improvement in LTW is achieved by eliminating bit-plane encoding (similar to SWEET). As a result, the compressed bit stream is not embedded, but it is resolution scalable, which means that the information corresponding to coarser image representation goes first. The encoding process consists of two passes. On the first pass, wavelet coefficients are labeled using special three labels:

LOWER, ISOLATED_LOWER and LOWER_COMPONENT. The first two labels directly correspond to Zero Tree Root and Isolated Zero labels used by Shapiro in EZW [Sha93]. The last label indicates that the coefficient belongs to a lower tree. On the second pass, the coefficients are scanned from the lowest resolution to the highest resolution subbands, and for each coefficient its label as well as the number of bits required to represent its magnitude are arithmetically coded. They are then followed by the least significant bits representing the coefficient magnitude (MSB is omitted) and its sign. The main advantage of the LTW algorithm is its simplicity. LTW does not exploit lists and reconstructs coefficients in one pass which leads to significant speed improvement.

Previous zero-tree coders do not fully take advantage of adaptive context modeling. EZW exploits simple Markov conditioning based on significance of previous coefficient in scan and parent coefficient significances. SPIHT uses adaptive context modeling to jointly encode significance of 4 sibling coefficients. LTW also exploits simple arithmetic coding algorithm, however no details are presented in [OM03]. It was shown in other works [CO97], [Wu97] that exploitation of high-order context modeling can yield significant compression performance improvement. However, algorithms presented in [CO97], [Wu97] are not zero-tree based. Besides ECECOW algorithm [Wu97] requires extensive trainings to initialize its coding structures. In this work, we tried to take best from both worlds – combine adaptive context modeling with zero-tree coding to improve compression performance keeping at the same time the algorithmic complexity low.

2.2 Compressed multiresolution terrain models

Though large-scale terrain visualization has long history, only a few recent works concentrate on efficient elevation data compression methods. The geometry clipmaps approach [LH04] exploits regular grid pyramid data structure that enables applying the lossy image compression technique [Mal00] to the terrain height map. However, this method cannot provide a guaranteed error bound, which becomes especially apparent on high-variation terrains such as the Grand Canyon. Another method that utilizes terrain elevation data compression is presented by E. Gobbetti, et al., [GMC+06] and is called C-BDAM. It exploits a wavelet-based two stage near-lossless compression technique presented by S. Yea and W. Pearlman [YP06]. A problem of seamless stitching neighboring patches is not covered by C-BDAM. Another method presented by C. Dick, et al. [DSW09], mainly focuses on compressing adaptive triangulation in a way that enables GPU-based decompression. The method achieves a moderate compression factor of 8-9. Elevation data compression techniques are not considered by C. Dick, et al.

The remained of this paper is organized as follows. In section 3, our image compression technique is described. In section 4, we present compressed multiresolution terrain representation based on the proposed image compression algorithm. Section 5 presents experimental results. Section 6 concludes the paper.

3. COMPRESSION ALGORITHM DESCRIPTION

The compression method we developed belongs to the class of zero-tree coders and extends the ideas of EZW [Sha93], SPIHT [SP96] and LTW [OM03] algorithms. As in EZW, the quantized wavelet coefficients are scanned from the lowest frequency subband to the highest frequency in our algorithm, and in each subband, coefficients are scanned in zigzag order. While EZW and SPIHT perform multiple bit-plane passes, our method encodes

each coefficient in only one pass. It encodes each coefficient magnitude using at most n_{bp} bits. It thus is similar to SWEET [And97] and LTW [OM03] algorithms, which discard a fixed number of least significant bits in quantized wavelet coefficients and generate non-embedded but resolution scalable bit stream.

3.1 Quantization

We use a uniform quantizer with a dead zone (fig. 3). The quantization step q is determined as $q = M / 2^{n_{bp}}$ where $M = \max_{s \in \{lh, hl, hh\}, i, j, l} |s_{i,j}^{(l)}|$ is the maximum wavelet coefficient magnitude of all detail subbands (LH, HL and HH) across all scales and n_{bp} is the maximum number of bits allotted to represent the coefficient magnitudes. The quantizer divides the range of all magnitudes $[0, M]$ onto $N = 2^{n_{bp}}$ uniform quantization bins B_n such that $B_n = \{s : nq \leq |s| < (n+1)q\}$.

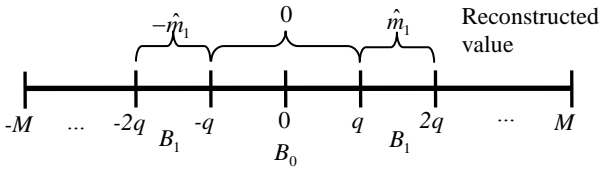


Fig. 3: Uniform quantizer with a dead zone.

We denote $n_B(s) = \lfloor |s| / q \rfloor$ to be the number of the bin which coefficient s falls into ($\lfloor \cdot \rfloor$ denotes the integer part). We will refer to this number later as quantized magnitude. Binary representation of $n_B(s)$ can be treated as a sequence of n_{bp} binary decisions where the first bit indicates whether the coefficient magnitude falls into the $[0, M/2)$ range (0 bit) or into the $[M/2, M]$ range (1 bit). The next bit refines the uncertainty interval to length $M/4$ (00: $[0, M/4)$, 01: $[M/4, M/2)$, 10: $[M/2, 3M/4)$, 11: $[3M/4, M]$) and so on. Magnitudes of all coefficients $s_{i,j}^{(l)}$ falling into the same quantization bin B_n are reconstructed equally as \hat{m}_n and at the decoder, the coefficients are restored as follows:

$$\hat{s}_{i,j}^{(l)} = \text{sign}(s_{i,j}^{(l)}) * \hat{m}_{i,j}^{(l)}$$

where $\text{sign}(s_{i,j}^{(l)}) = +1$ if $s_{i,j}^{(l)} > 0$ and $\text{sign}(s_{i,j}^{(l)}) = -1$ if $s_{i,j}^{(l)} < 0$; $\hat{m}_{i,j}^{(l)}$ is the reconstruction level for the quantization bin which $s_{i,j}^{(l)}$ falls into (with the number $n_B(s_{i,j}^{(l)})$). The question is what value to use as a reproduction level \hat{m}_n for the bin B_n . If the probability distribution function (PDF) $f(t)$ of the coefficient magnitudes was known, the optimal reproduction levels for each bin would be placed at the centroid of the distribution for that bin:

$$\hat{m}_n = E(nq \leq |s_{i,j}^{(l)}| < (n+1)q) = \frac{\int_{nq}^{(n+1)q} tf(t)dt}{\int_{nq}^{(n+1)q} f(t)dt} \quad (1)$$

Exact distribution is unknown; however it is usually assumed that wavelet coefficients have Laplacian distribution with zero mean, that is the coefficient magnitudes distribution can be well approximated by the following PDF:

$$f(t) = \alpha e^{-\alpha t} \quad (2)$$

Given that assumption, the reconstruction levels \hat{m}_n can be calculated as follows:

$$\hat{m}_n = \frac{\int_{nq}^{(n+1)q} \alpha t e^{\alpha t} dt}{\int_{nq}^{(n+1)q} \alpha e^{\alpha t} dt}$$

Reducing this formula gives the following equation for \hat{m}_n :

$$\hat{m}_n = nq + \frac{1}{\alpha} - \frac{q}{e^{q\alpha} - 1} \quad (3)$$

The unknown parameter α in (3) can be estimated based on the magnitudes of wavelet coefficients. This could be done using one of the methods from mathematical statistics. Since our goal is to minimize reconstructed image error, we exploit different approach. It is difficult to find exact α that minimizes the distortion in image space without performing extensive calculations. We instead calculate α such that it minimizes the distortion in wavelet domain. Due to bi-orthogonality of wavelet transform, this method also gives close to optimal solution in image space. We thus minimize the following error function that gives mean square error in wavelet domain:

$$D(\alpha) = \sum_{n=1}^{N-1} \sum_{s \in B_n} (|s| - \hat{m}_n)^2 \rightarrow \min$$

It can be seen from (3) that all reproduction levels \hat{m}_n are equally shifted from the quantization bin B_n lower bound nq :

$$\hat{m}_n = nq + \beta \quad \text{where} \quad (4)$$

$$\beta = \frac{1}{\alpha} - \frac{q}{e^{\alpha q} - 1}. \quad (5)$$

We thus need to find the optimal shift β that minimizes the following function:

$$D(\beta) = \sum_{n=1}^{N-1} \sum_{s \in B_n} (|s| - (nq + \beta))^2 \rightarrow \min$$

The minimum of this function can be derived from the root of the following equation: $\frac{dD(\beta)}{d\beta} = 0$

which gives the following formula for optimal β :

$$\beta = \frac{1}{N_s} \sum_{n=1}^{N-1} \sum_{s \in B_n} (|s| - nq) \quad (6)$$

Where N_s is the total number of significant coefficients. It follows from (6) that the optimal shift β for reproduction levels is the average magnitude shift from the quantization bins' lower bounds. This is intuitively reasonable result.

Given β , the distribution parameter α can be calculated by reverting equation (5), however in fact, we do not need α since we only interested in optimal shift β . Thus $\beta^{(l)}$ parameters are calculated separately for all decomposition levels in our algorithm and are encoded as side information. Since β falls in the range $[0, q]$, the following symbol is encoded using $nbits_{\beta}$ bits:

$$\tilde{\beta}^{(l)} = \lfloor \beta^{(l)} / q \cdot 2^{n_{bits_p}} \rfloor.$$

At decoder, the parameter is reconstructed as follows:

$$\hat{\beta}^{(l)} = \lfloor (\tilde{\beta}^{(l)} + 0.5) / 2^{n_{bits_p}} \cdot q \rfloor.$$

We use 7 bits ($n_{bits_p} = 7$) for encoding $\beta^{(l)}$.

Wavelet coefficients of the lowest frequency LL subband in our algorithm are quantized and are arithmetically encoded using separate model for each bit position. Since large values are less possible, arithmetic coding reduces the compressed data size.

The compression performance of different wavelet-based algorithms is primarily determined by how efficiently the quantized coefficients are encoded. In the next subsections, we will describe our adaptive context models that improve compression efficiency of arithmetic coding. To distinguish our method from others, we call it ACMZW (Adaptive Context Modeling Zero-tree Wavelet coder).

3.2 Utilizing adaptive context modeling to predict the coefficient magnitude

Similar to LTW, our method encodes number of bits $n_{bits}(s)$ required to represent each significant coefficient s quantized magnitude $n_B(s)$ using arithmetic coding [WNC87] and then transmits the magnitude bits followed by the sign. To exploit information carried by already encoded neighboring coefficients, we utilize adaptive context modeling. For this purpose, we first predict the reconstructed coefficient magnitude. Wavelet transform localize energy in both frequency and spatial domains: WT coefficients of similar magnitudes statistically cluster in frequency subbands and in spatial locations. As a result, the reconstructed coefficient $\hat{s}_{i,j}^{(l)}$ estimated magnitude can be well derived from the magnitudes of the coefficients in the context, containing four neighbors (one to the left ($\hat{s}_{i-1,j}^{(l)}$)) and three from above ($\hat{s}_{i-1,j-1}^{(l)}$, $\hat{s}_{i,j-1}^{(l)}$ and $\hat{s}_{i+1,j-1}^{(l)}$) and one direct ancestor ($\hat{s}_{i/2,j/2}^{(l-1)}$) (see fig. 4).

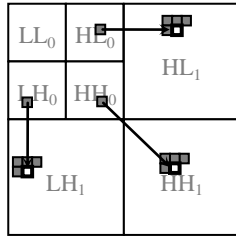


Fig. 4: 5-coefficients contexts used to estimate the coefficient's magnitude.

We use the following expression to calculate the expected coefficient magnitude $\hat{m}_{i,j}^{(l)}$:

$$\hat{m}_{i,j}^{(l)} = \sqrt{w_0(\hat{m}_{i-1,j}^{(l)})^2 + w_1(\hat{m}_{i-1,j-1}^{(l)})^2 + w_2(\hat{m}_{i,j-1}^{(l)})^2 + w_3(\hat{m}_{i+1,j-1}^{(l)})^2 + w_4\sqrt{(\hat{m}_{i/2,j/2}^{(l-1)})^2}} \quad (7)$$

In the expression above, the first summand represents the energy localization property in space: it is expected that neighboring coefficients will have a similar energy. The second summand represents the energy localization property in frequency: the ancestor's energy is distributed among its descendants.

Weights w_i can be optimized for each subband for particular image. However, this would require additional computations and a lot of side information to be sent, so we use constant weights instead, which are optimized for a test set of images. Since LH subband exhibits predominantly horizontal structures, we use the following coefficients for this subband:

$$(w_0, w_1, w_2, w_3, w_4) = (0.4, 0.15, 0.3, 0.15, 0.125).$$

HL subband exhibits predominantly vertical structures, so we apply the following weights for this subband:

$$(w_0, w_1, w_2, w_3, w_4) = (0.3, 0.15, 0.4, 0.15, 0.125).$$

HH subband does not exhibit explicit structures, and we exploit the following weights:

$$(w_0, w_1, w_2, w_3, w_4) = (0.25, 0.25, 0.25, 0.25, 0.125).$$

Due to the zig-zag scanning order, all coefficients in the context are evaluated first, so $\hat{m}_{i,j}^{(l)}$ is always properly calculated. The number of bits $n_{bits}(n_B(\hat{m}_{i,j}^{(l)}))$ required to represent the quantized predicted magnitude $n_B(\hat{m}_{i,j}^{(l)})$ gives the adaptive context model number to be used in arithmetic coder. To further improve compression performance, we encode the maximum number of bits required in each level of wavelet decomposition and use separate context sets (consisting of $max_bits(level)$ models) for each level. After the number of bits $n_{bits}(n_B(s_{i,j}^{(l)}))$ required to represent the exact quantization bin number $n_B(s_{i,j}^{(l)})$ is encoded with the appropriate model (which is given by $n_{bits}(n_B(\hat{m}_{i,j}^{(l)}))$ and decomposition level), all the coefficient magnitude refinement bits (which are the bits in binary representation of the $n_B(s_{i,j}^{(l)})$) excepting the most significant one are transmitted. We determined that exploiting arithmetic coding for refinement bits also improves compression performance. This is achieved by using a separate arithmetic model for each bit position and for each number of bits required to represent bin number.

3.3 Utilizing adaptive context modeling to predict insignificant coefficient sets

While the LTW algorithm exploits the EZW-like coding style to encode large sets of insignificant coefficients, with additional symbols corresponding to degree-1 zero tree, we adopt the more advanced SPIHT-style coding method. Since SPIHT is a degree-two zero tree coder (see [CP07]), while LTW is a degree-one zero tree coder, the exploitation of SPIHT-style coding gives additional benefit [CP07]. In [SP96], the set of coordinates of all offsprings (direct descendants) of the node is called *set type A*. The set of all descendants of the node excepting its offsprings is called *set type B*. Set is defined to be significant if it contains at least one significant coefficient. SPIHT defines smart set partitioning rules that are used to efficiently separate significant and insignificant coefficients (see [SP96] for more details).

SPIHT jointly encodes the significance of four neighboring sets of type A and uses conditioning on the significance of set type A to encode the significance of set type B. In our algorithm, we implemented different and more efficient approach. As with magnitudes, we predict the significance of sets of type A and B based on already encoded information (fig. 5) and taking into account space and frequency localization properties of wavelet transform.

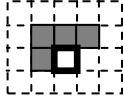


Fig. 5: 4-coefficients context used to predict set type A and set type B significance.

Due to the space localization property, it can be expected that the more significant neighboring sets of type A (B) the coefficient has, the more probable its set type A (B) is significant. Due to the frequency localization property, it is expected that the more energy is concentrated in this spatial location, the more probable the coefficient's set type A (B) is significant. Combining these two considerations, we derive the following expressions to calculate context model number to encode the set type A (B) significance:

$$CM_A = A_{i-1,j}^{(l)} + A_{i-1,j-1}^{(l)} + A_{i,j-1}^{(l)} + A_{i+1,j-1}^{(l)} + (4, \text{if } |s_{i,j}^{(l)}| > 2q)$$

$$CM_B = B_{i-1,j}^{(l)} + B_{i-1,j-1}^{(l)} + B_{i,j-1}^{(l)} + B_{i+1,j-1}^{(l)} + (4, \text{if } |s_{i,j}^{(l)}| > q)$$

where $A_{i,j}^{(l)} = 1$ ($B_{i,j}^{(l)} = 1$) if set type A (B) is significant and $A_{i,j}^{(l)} = 0$ ($B_{i,j}^{(l)} = 0$) otherwise. Thresholds $2q$ and q were found out empirically for test set of images. The presented equations have the meaning that the greater context model number, the more probable the set is significant.

3.4 Encoding coefficient signs

In [BP01], the sign/significance information is encoded using 3^m symbols for m yet insignificant coefficients in a group of 2×2 sibling coefficients. We figured out that it is more efficient to encode sign separately using one out of 27 context models for each subband. The context model number is defined depending on the sign and significance of three already encoded neighbors in the context shown in fig. 6. There can be 3 possible states for each neighbor: positive, negative or insignificant (zero), which gives 27 possible combinations for three coefficients. Since signs tend to produce different patterns in each subband, we use 3 separate context model sets for each subband (LH, HL and HH).

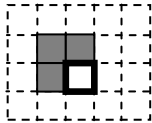


Fig. 6: 3-coefficients context used to encode the sign.

The close approach was implemented in ECECOW algorithm [Wu97], however contexts of ECECOW contain much more coefficients and thus they require much more data to provide reliable probability estimations.

3.5 Lossy plus residuals coding that guarantees the specified tolerance of the reconstructed data

In many applications, such as medical imaging, the decompressed image must satisfy some pre-defined error tolerance in L^∞ sense. For terrain compression, the L^2 error bound is also clearly inappropriate. The L^∞ error bound can not be guaranteed in wavelet domain, so many methods exploit lossy plus residual coding approach [AMC98]. In this scheme, the data is first compressed using the lossy coder and it is then supplemented by the encoded quantized residuals that guarantee a given L^∞ error bound δ

(refer to [AMC98, YP06] for more details). For this purpose, the residual layer R which is the difference between the decompressed image \hat{I} and original image I is calculated: $R = I - \hat{I}$. Since pixels of 8-bit images can only be integer values, the residual layer is quantized using the following rule [AMC98]:

$$\tilde{r}_{i,j} = \lfloor (|r_{i,j}| + \delta) / (2\delta + 1) \rfloor \quad (8)$$

At decompression, the residuals are reconstructed as follows: $\hat{r}_{i,j} = \text{sign}(r_{i,j}) \cdot \tilde{r}_{i,j} \cdot (2\delta + 1)$. Note that before calculating the residual layer, the decompressed image \hat{I} must be rounded to integer values. The resulting decompressed image is then obtained as $\hat{I} + \hat{R}$ and differs from original image I by at most δ .

Quantized residuals $\tilde{r}_{i,j}$ as well as their signs $\text{sign}(r_{i,j})$ are arithmetically encoded in our method using adaptive context modeling. We first estimate average variation in the lossy decompressed image around the residual using the following equation:

$$V_{i,j} = \sqrt{\frac{1}{8} \sum_{m,n=-1}^1 (\hat{i}_{i+m,j+n} - \hat{i}_{i,j})^2}$$

context as that used for encoding set type A/B significance (fig. 5) to take into account already encoded neighboring residuals. Finally, the context model number is determined as follows:

$$CM_{n,j} = \lfloor V_{i,j} / \delta \rfloor + \tilde{r}_{i-1,j} + \tilde{r}_{i-1,j-1} + \tilde{r}_{i,j-1} + \tilde{r}_{i+1,j-1}$$

Note that [AMC98] exploits much more complex method to determine context model number; however our method being much simpler provides comparable compression results as shown in section 5. Residual signs are encoded in the same way as described in section 3.4. In our current implementation, the optimal lossy bitrate is determined iteratively, however it can be estimated during the encoding process as described in [YP06].

4. COMPRESSED MULTIREOLUTION TERRAIN MODEL

4.1 Hierarchically encoding height map using ACMZW algorithm

In this section, we describe compressed multiresolution terrain representation that exploits the presented compression method. At first, the initial height map is prefiltered into a mipmap pyramid much like it is done in the geometry clipmaps framework [LH04] and our earlier approach [YT08]. Each level of the pyramid has two times fewer samples in each direction and a two times longer sample spacing interval and thus approximates the original height map with diminishing accuracy. For pyramid construction, we use normalized Daubechies 9/7 low-pass wavelet filter.

At the next stage, the patch quad tree data structure is constructed by subdividing each level of the pyramid into square blocks having an equal number of samples (64x64, 128x128, 256x256 etc.) as shown in fig. 7. The resulting hierarchy is compressed in a top-down order starting from the coarsest resolution (patch at level 0). During that process, for each patch, special refinement information is encoded that enables reconstructing patch descendant's height maps. For this purpose, the difference layer $D = \{d_{i,j}\}^{2n \times 2n}$ is calculated as the discrepancy between the four child patches' predicted height maps ($H^P = \{h_{i,j}^P\}^{2n \times 2n}$) and their exact height maps ($H = \{h_{i,j}\}^{2n \times 2n}$) as shown in fig. 8: $D = H - H^P$.

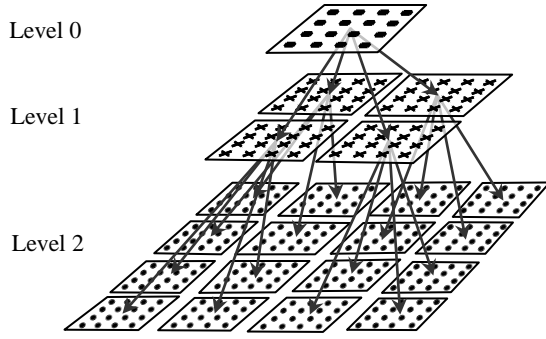


Fig. 7: Patch quad tree data structure.

The predicted height maps H^p are obtained by applying Daubechies 9/7 synthesis low-pass wavelet filter to the parent patch height map. For the coarsest resolution patch (level 0), the predicted height map is defined to be zero. An important aspect here is that the parent patch height map is extracted from a compressed representation instead of using the exact data from the multiresolution representation. This eliminates error propagation from coarser to finer levels.

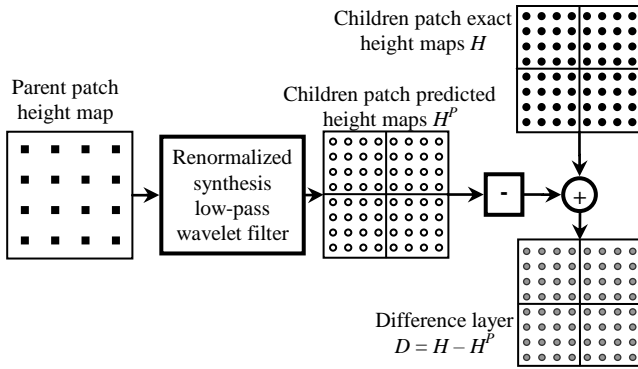


Fig. 8: Calculating the difference layer.

The difference layer D for each group of four sibling patches is compressed using a two-stage lossy plus residuals coding scheme described in section 3.5 with another residual quantization rules.

At first, we attempted to compress levels with exponentially increasing error tolerance such that patches at the finest level are compressed with a user-defined world space error threshold δ , patches at the next-coarser level are compressed with threshold 2δ and so on. However, later we found out that it is more efficient to compress each level of the pyramid using the same tolerance δ . In the latter case, the prediction turns out to be more accurate and as a result, the difference layer is compressed more efficiently. Besides, we use another quantization rule when compressing patches at all coarser levels excepting the finest one:

$\tilde{r}_{i,j} = \lfloor |r_{i,j}| / \delta \rfloor$. Residual reconstruction is performed as follows:

$$\hat{r}_{i,j} = \begin{cases} \text{sign}(r_{i,j}) \cdot (\tilde{r}_{i,j} + 0.5) \cdot \delta, & \tilde{r}_{i,j} \neq 0 \\ 0, & \tilde{r}_{i,j} = 0 \end{cases}$$

This quantizer has the dead zone of length 2δ around zero, but the rest quantization intervals are two times shorter. This also results in a more precise prediction and leads to more compact representation. Besides, since elevation data samples are not integer values, we use more accurate quantizer for the finest level than that given by (8):

$$\tilde{r}_{i,j} = \lfloor (|r_{i,j}| + \delta) / (2\delta + p) \rfloor \quad (9)$$

where p is the height map samples representation precision.

We found out that exploiting two-stage compression yields better results compared to one-stage encoding, where the difference layer is directly compressed.

For each patch in the compressed hierarchy, we store its upper approximation error bound. We recursively calculate it as the sum of three terms: 1) the maximum distance between patch's interpolated height map and the vertices at the next finer resolution; 2) the maximum reconstruction error of the patch descendants and 3) elevation data reconstruction error. This value is used to construct adaptive view-dependent block-based terrain approximation.

4.2 Seamlessly stitching neighboring patches

None of previous elevation data compression techniques take care of seamless elevation data connection and normal map generation. In our representation, neighboring patches have common elevation data samples and each patch is "responsible" for seamlessly stitching transition with its right and top neighbors. The two-stage compression technique presented in section 3.5 guarantees that each elevation data sample's reconstruction error is within the predefined threshold. However, it is not guaranteed, that the common samples of neighboring patches both being within the tolerance are reconstructed equally. To cope with this problem, we introduced special boundary area around four sibling patches' height maps. We call this area "matching boundary" (see fig. 9).

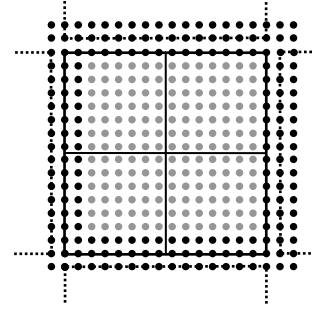


Fig. 9: Matching boundary.

Matching boundary consists of several rings (3 in example). Note that it is sufficient to have one ring to assure seamless geometric stitching, however for seamless normal map stitching more rings are required. Each ring is compressed separately without wavelet transform using the following algorithm. The ring is treated as a one dimensional sequence of elevations h_i , which are quantized using (9). For each quantized elevation \tilde{h}_i (excepting \tilde{h}_0), the difference $d_i = \tilde{h}_i - \tilde{h}_{i-1}$ is encoded using arithmetic coding. We use simple context modeling based on previous quantized difference d_{i-1} to improve compression performance. Signs are also arithmetically encoded using conditioning on previous sign.

Matching boundary consisting of 3 rings assures that the normal maps calculated for neighboring patches perfectly connect.

In terrain rendering systems, sometimes it is required to generate elevation data finer than the original data set define. In our test system, we implemented procedural height map generation algorithm that exploits local surface elevation and slope. The presented matching boundary also assures that procedurally generated height maps for neighboring patches perfectly match.

5. EXPERIMENTAL RESULTS AND DISCUSSION

5.1 Image compression performance

In our tests, we used Daubechies 9/7 bi-orthogonal wavelet filters [ABM92] with 5 decomposition levels. The compression performance gain obtained by applying each modification described in sections 3.2-3.4 for Lena image is presented in Table 1. The table shows compression ratio of the not optimized algorithm and compressed bit stream size after applying each modification as a fraction (in percents) of the bit stream size generated by the basic algorithm.

# bit planes	PSNR, dB	Basic (no opts), bpp	Magn. predict. (sec. 3.2)	Set type A/B signif. prediction (sec. 3.3)	Sign coding (sec. 3.4)	All opts
6	29.936	0.098	96.4%	98.0%	98.9%	93.2%
7	33.187	0.212	95.6%	97.9%	98.2%	91.7%
8	36.343	0.436	94.6%	98.0%	97.9%	90.5%
9	39.652	0.931	92.8%	98.3%	98.2%	89.3%
10	44.413	1.946	92.3%	99.3%	98.9%	90.5%

Table 1: Compression ratio improvements resulting from exploiting presented context models for Lena image.

The results of compressing standard test images, Lena and Barbara in comparison with LTW [OM03] and SPIHT [SP96] methods are presented in Tables 2 and 3. To have fair comparison, we also implemented LTW and SPIHT-AC methods as it is described in original papers. The exact bit rate in our method is achieved by tuning the M value and thus by adjusting the quantization step q .

codec\ rate	LTW [OM03]	LTW (Ours)	SPIHT-AC [SP96]	ACMZW
0.125	31.27	31.01	31.10	31.20
0.25	34.31	33.98	34.11	34.28
0.5	37.35	37.07	37.21	37.39
1.0	40.50	40.13	40.41	40.55
2.0	45.46	44.82	45.07	45.67

Table 2: PSNR (dB) values at various rates for Lena image.

codec\ rate	LTW [OM03]	LTW (Ours)	SPIHT-AC [SP96]	ACMZW
0.125	25.52	25.17	25.23	25.55
0.25	28.33	27.89	27.83	28.33
0.5	31.78	31.34	31.45	31.9
1.0	35.88	35.41	35.69	35.94
2.0	40.74	39.97	40.40	40.9

Table 3: PSNR (dB) values at various rates for Barbara image.

Our implementation of the LTW encoder based on exact formal description of the algorithm presented in [OM03] incorporates all improvements mentioned in [OM03] and a few our modifications that also improve the compression ratio. In Tables 2 and 3, we presented the best results we obtained for our LTW implementation. However despite all modifications, our implementation of the LTW method demonstrated significantly poorer compression performance compared to the results reported in [OM03] (up to 0.77 dB below the reported results (see tables 2 and 3) and 0.43 dB below in average). This is not an issue of the implementation

since LTW algorithm is rather simple from one hand, and our implementation of the SPIHT encoder demonstrates exactly the same compression ratios as stated in [SP96], from the other. The lack between real compression performance and reported in [OM03] is probably the result of adaptive context modeling implemented in real LTW which is only mentioned in [OM03] but no details are presented in the paper. In this work, we thoroughly described adaptive context models we used in our algorithm that enables us to improve the compression performance and achieve the same or higher compression rates compared to LTW. As tables 2 and 3 show, our method demonstrates better compression performance (up to **0.6 dB**) on all bitrates compared to SPIHT-AC [SP96]. It also demonstrates comparable or better (up to **0.21 dB**) performance than reported in [OM03] and significantly better performance than our exact implementation of LTW.

Table 4 shows compression performance for Lena image for different maximum absolute error thresholds compared to other methods (the data is taken from [YP06]). These results are obtained using quantizer (8).

Method\abs err	1	2	4	6	7
JPEG-LS (bpp)	2.72	2.09	1.54	1.24	1.14
CALIC (bpp)	2.59	1.95	1.29	0.96	0.85
[ACM98] (bpp)	2.69	2.02	1.28	0.86	0.73
ACMZW (bpp)	2.68	2.02	1.30	0.88	0.75
lossy + residual	0.47+2.21	0.39+1.63	0.45+0.85	0.39+0.49	0.39+0.36

Table 4: Comparing compression performance for various maximum absolute errors with other methods for Lena image.

As table 4 shows, though we used much simpler context to compress residuals, our method demonstrates compression performance comparable to [ACM98] and comparable to or superior than other methods such as JPEG-LS [WSS00] and CALIC [WM97]. Our method also demonstrates **1.39** to **2.9** times higher compression ratios compared to FBTR method [Zhe04].

5.2 Elevation data compression performance

To test the performance of our terrain compression algorithm, we used two different elevation data sets. The first data set is the Puget Sound being 16384x16384 in size and sampled at 10 m spacing. This data set is used as the common benchmark and is available at [PS]. The second one is the Grand Canyon being 8192 x 8192 in size and sampled at 30 m spacing. The elevation data precision is 0.1 m, so we use $p=0.1$ in (9). The compression results for patch size 256x256 and 64x64 and matching boundary width 3 are presented in Tables 5 and 6. The compression and run-time experiments were done on a workstation with the following hardware configuration: CPU: Intel Core i7 @2.67 GHz (4 cores with 2 hyper threads each); 6.0 GB RAM; GeForce GTX275 graphics card.

Tolerance (m)	Puget 256M			Grand Canyon 64M		
	rms (m)	Compr. rate (bps)	Compr. time (s)	rms (m)	Compr. rate (bps)	Compr. time (s)
1	0.449	0.636	294	0.539	1.963	124
3	1.120	0.287	235	1.445	1.031	80
10	3.099	0.116	223	3.962	0.414	61

Table 5: Compressing Puget 256M and Grand Canyon 64M data sets with patch size 256x256 and matching boundary size 3.

Tolerance (m)	Puget 256M			Grand Canyon 64M		
	rms (m)	Compr. rate (bps)	Compr. time (s)	rms (m)	Compr. rate (bps)	Compr. time (s)
1	0.423	0.837	255	0.531	2.282	93
3	1.077	0.431	229	1.407	1.237	68
10	2.931	0.21	221	3.898	0.54	56

Table 6: Compressing Puget 256M and Grand Canyon 64M data sets with patch size 64x64 and matching boundary size 3.

The compression method presented is up to 2.5 times more efficient than our previous algorithm [YT08]. The same Puget Sound data set was compressed by C-BDAM with 1 m max error to 0.61 bps, as reported in [GMC+06]. Without matching boundary and using patch size 256x256 our algorithm achieved 0.588 bps, which proves high compression potential of the proposed method.

During an interactive flight over the Grand Canyon and Puget Sound data sets with procedural terrain surface texturing and atmospheric effects rendered at 1920x1200 resolution, the frame rates never dropped below 120 fps. The decompression is performed significantly faster (less than 0.1 s to decompress one patch) than compression and in conjunction with the asynchronous rendering algorithm, it provides steady frame rates. These results show that our method can be successfully used in real-time terrain rendering applications.

6. CONCLUSION AND FUTURE WORK

In this paper, we presented a new image compression algorithm called ACMZW that combines smart adaptive context modeling with zero-tree coding and demonstrates top compression performance in the class of zero tree wavelet coders such as EZW, SPIHT and LTW. At the same time, the algorithmic complexity of the algorithm is comparable to that of LTW since it bypasses multiple bit-plane coding.

Our experiments showed that exploiting proposed context models reduces the compressed bit stream size by more than 10%. Our ACMZW method demonstrates up to **0.6 dB** compression performance superiority over SPIHT with arithmetic encoding [SP96] and up to **0.21 dB** superiority over LTW [OM03] on standard test images (Lena, Barbara).

A new hierarchical compressed multiresolution terrain model is proposed that is based on presented image compression technique and can be exploited in high-quality real-time terrain rendering systems. The model introduces special matching boundary region that is compressed in a way that guarantees the borders of neighboring patches perfectly match. The algorithm demonstrates compression performance that is on par with the best known methods.

Our future work will be aimed at exploiting the recent graphics hardware for improving temporal performance of the algorithms. The most computation-intensive parts of the presented compression technique can be implemented entirely on GPU using compute shader, the new capability exposed by DX11-class hardware.

7. REFERENCES

[WNC87] I.H.Witten, R.M.Neal, and J.G.Cleary, "Arithmetic coding for data compression". *Comm. ACM*, vol. 30, No. 6, pp.520-540, June 1987.

[ABM92] M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Proc.*, Vol. 1, No. 2, pp. 205-220, 1992.

[Sha93] J.M. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients," *IEEE Transactions on Signal Processing*, vol. 41, no 12, pp. 3445-3462, Dec. 1993.

[SP96] A. Said and W. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no 3, pp. 243-250, June 1996.

[CO97] C. Chrysafis, A. Ortega, "Efficient Context-Based Entropy Coding for Lossy Wavelet Image Compression", *Proceedings of the Conference on Data Compression*, pp.241-250, March 25-27, 1997.

[Wu97] X. Wu, "High-order context modeling and embedded conditional entropy coding of wavelet coefficients for image compression," *Proc. ACSSC*, Nov. 1997.

[WM97] X. Wu and N.D.Memon, "Context-based, adaptive, lossless image coding," *IEEE Trans. Commun.*, vol. 45 (4), pp. 437-444, Apr. 1997.

[And97] J. Andrew, "A simple and efficient hierarchical image coder," *ICIP'97*, vol. 3, pp 658-661, 1997.

[AMC98] R. Ansari, N.Memon, and E. Ceran, "Near-lossless image compression techniques," *J. Electronic Imaging*, vol. 7, pp. 486-494, Jul. 1998.

[Mal00] H. Malvar. "Fast Progressive Image Coding without Wavelets." *In Data Compression Conference (DCC '00)*, pp. 243-252, 2000.

[WSS00] M. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEGLS," *IEEE Trans. Image Processing*, vol. 9, pp. 1309-1324, Aug. 2000.

[BP01] U. Bayazit and W. A. Pearlman, "Algorithmic Modifications to SPIHT". *IEEE Int. Conf. on Image Processing (ICIP 2001)*, Thessaloniki, Greece., Oct. 2001.

[OM03] J. Oliver and M. P. Malumbres, "Fast and efficient spatial scalable image compression using wavelet lower trees," *In Proc. 2003 IEEE Data Compression Conference*, Mar 2003, pp. 133-142.

[LH04] F. Losasso, H. Hoppe, "Geometry Clipmaps: Terrain Rendering Using Nested Regular Grids". *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2004)* 23(3), pp. 769-776.

[Zhe04] S.V. Zherzdev. "Development of adaptive video information compression algorithms based on hierarchical structures for operation visualization". *PhD Thesis*. Nizhny Novgorod State University, Nizhny Novgorod, Russia.

[YP06] S. Yea and W. Pearlman. "A wavelet-based two-stage near-lossless coder". *In Proc. ICIP (2004)*, pp. 2503-2506.

[GMC+06] E. Gobbetti, F. Marton, P. Cignoni, M. Di Benedetto, and F. Ganovelli. "C-BDAM - Compressed Batched Dynamic Adaptive Meshes for Terrain Rendering". *Computer Graphics Forum*, Volume 25(2006), Number 3.

[Cp07] Y. Cho and W. A. Pearlman, "Quantifying the Performance of Zerotrees of Wavelet Coefficients: Dgree-k Zerotree Model", *IEEE Trans. on Signal Processing*, Vol. 55, Part 1, pp. 2425-2431, June 2007.

[YT08] Egor Yusov, Vadim Turlapov. "JPEG2000-based compressed multiresolution model for real-time large scale terrain visualization". *Proc. of the 18th international Conference on Computer Graphics and Vision "GraphiCon'2008"* pp. 164-171, Moscow, June 23-27, 2008.

[DSW09] C. Dick, J. Schneider, and R. Westermann. "Efficient Geometry Compression for GPU-based Decoding in Realtime Terrain Rendering". *In Computer Graphics Forum*, Vol. 28, No 1, pp. 67-83, 2009.

[PS] Puget Sound elevation data set is available at http://www.cc.gatech.edu/projects/large_models/ps.html

AN EFFICIENT COLLISION DETECTION ALGORITHM FOR POINT CLOUD MODELS

Mauro Figueiredo¹, João Oliveira², Bruno Araújo², João Pereira²

¹ Instituto Superior de Engenharia, Universidade do Algarve, Faro, Portugal
mfiguei@ualg.pt

² IST/INESC-ID, Rua Alves Redol, 9, 1000-029 Lisboa
{joao.oliveira, bruno.araujo, jap}@inesc-id.pt

Abstract

Point clouds models are a common shape representation for several reasons. Three-dimensional scanning devices are widely used nowadays and points are an attractive primitive for rendering complex geometry. Nevertheless, there is not much literature on collision detection for point cloud models.

This paper presents a novel collision detection algorithm for point cloud models. The scene graph is divided in voxels. The objects of each voxel are organized in R-trees hierarchies of Axis-Aligned Bounding Boxes to group neighboring points and filter out very quickly parts of objects that do not interact with other models. The proposed algorithm also uses Overlapping Axis-Aligned Bounding Boxes to improve the performance of the collision detection process. Points derived from laser scanned data typically are not segmented and can have arbitrary spatial resolution thus introducing computational and modeling issues. We address these issues and results show that the proposed collision detection algorithm effectively finds intersections between point cloud models since it is able to reduce the number of bounding volume checks and updates.

Keywords: *Collision detection, virtual environments, surface segmentation, point cloud processing.*

1. INTRODUCTION

Point cloud models are an increasingly attractive representation used as the basis to capture and measure reality rapidly in an increasing number of applications such as environmental surveying, structure analysis and archaeology [1]. Point cloud models also share a remarkable similarity with a very popular computer game representation of the 80s, numerous Sinclair Spectrum games used axonometric projection of point models to convey details of buildings, interiors and avatars. A crucial element to enable laser scanned point models to be used in a similar scenario is collision detection of point clouds. In general interactive virtual environments often need very fast collision detection queries to simulate physical behaviour and to allow the user to interact. However, there is practically no literature on determining collisions between two sets of points.

This paper describes a novel collision detection algorithm for point cloud models.

The scene graph is organized in voxels. To speed up the process of finding collisions, for each voxel, each object is represented by an R-tree data structure of Axis-Aligned Bounding Boxes (AABB) defined in its own local coordinate system. The R-tree organizes spatially its geometry, grouping neighbouring points.

The proposed algorithm is also based in the use of the Overlapping Axis-Aligned Bounding Box (OAABB) to improve

the performance of the collision detection process. In addition a traversal algorithm for collision detection for point clouds that takes advantage of the OAABB is also presented, improving performance by reducing the number of bounding volume checks and updates.

Results show that the proposed approach uses effectively the R-tree structure and the OAABB concept to find intersections between point cloud models at interactive rates. In addition, unlike CAD objects which typically contain object hierarchies and the data is already segmented into surface groups, point data sets derived from laser scanned data do not have such information thus presenting computational issues. We address these issues and present a solution that adapts to point sets derived from different laser scanners and spatial settings.

The paper is organized as follows. Section 2 presents collision detection approaches for the determination of intersections between polygonal and point cloud models. Section 3 describes the VIZIR project that highlighted the need to develop an efficient collision detection algorithm for point cloud models. Section 4 describes the data structures for the representation of the scene graph that are used to improve the performance of the novel collision detection algorithm, which is presented in Section 5. Section 6 presents the evaluation results using CAD models and addresses laser scanned point sets. Conclusions and future work are presented in Section 7.

2. RELATED WORK

Currently, there are many implementations of collision detection schemes for interactive system, most of them only support polygonal models. Frequently, they use bounding volume hierarchies (BVH), spatial subdivision methods and more recently use graphics hardware to accelerate collision detection by hardware. There is a lack of collision detection systems for point cloud models.

Bounding volume hierarchies are frequently used to organize the triangles of an object to improve the performance of the collision detection process, by reducing the number of pairs of bounding volume tests. The classic scheme for hierarchical collision detection is a simultaneous recursive traversal of two bounding volumes trees A and B.

Several types of bounding volumes are available. Bounding spheres can be used [2]. SOLID [3] and OPCODE [4] use axis-aligned bounding boxes (AABB). RAPID [5], V-COLLIDE [6], PQP [7], H-COLLIDE [8], use oriented bounding boxes (OBB). QuickCD [9] and Dop-Tree [10] uses k-dops; and Swift++ [11] uses convex hulls (CH). There are also hybrid approaches like QuOSPOs [12] that use a combination of OBBs and k-dops.

The main advantage of SOLID, OPCODE and Box-Tree is that AABBs are faster to intersect.

RAPID approximates 3D objects with hierarchies of oriented bounding boxes (OBBs). The main advantage of RAPID is that OBBs are better approximations to triangles reducing effectively the number of intersecting operations.

V-COLLIDE solves the broad-phase of the collision detection using a sweep-and-prune operation to find pairs of objects potentially in contact. It uses RAPID to find in the narrow phase which pairs of objects intersect.

H-COLLIDE is a framework to find collisions for haptic interactions. It uses a hybrid hierarchy of uniform grids and trees of OBBs to exploit frame-to-frame coherence. It was specialized to find collisions between a point probe against 3D objects.

The QuickCD and Dop-Tree implementations build a hierarchy tree of discrete orientation polytopes (k-dops). The main advantage of using discrete orientation polytopes is that k-dops are better approximations to the underlying geometry than AABBs with the advantage of its low cost compared to OBBs.

Swift++ builds a hierarchy of convex hulls and intersection is tested using a modified Lin-Canny closest feature algorithm.

He [12] uses a hybrid approach that combines OBBs and k-dops called QuOSPOs. This approach provides a tight approximation of the original model at each level.

Another class of hierarchical data structures used for collision detection are spatial partitioning representations: regular grids [13, 14, 15, 16], octrees [17, 18], BSP-trees [19] and R-trees [20].

Spatial subdivisions are a recursive partitioning of the embedding space occupied by objects. In general, spatial partitioning structures are used as a secondary representation for the collision detection process.

The main idea behind all space partitioning methods is to exploit spatial coherency. For each object, we check for collision only objects of the neighborhood, eliminating comparisons with those objects that are far away and therefore cannot be colliding.

Zyda [13] uses grids to find overlapping objects in the broad phase. García-Alonso [14] also uses uniform grids to find exact collisions between 3D objects for the narrow phase. Teschner [21] use uniform grid subdivision combined with hashing to reduce storage requirements for collision and self-collision detection of deforming objects that consist of tetrahedrons. Eits [22] also uses a spatial grid inspired by the work of Teschner together with 1D hash table to find collisions between deformable tetrahedral models. A hybrid approach is presented by Gregory [8] using regular grids, where each occupied grid cell stores an OBBs tree of those triangles on that cell.

Hubbard [17] approach for finding collisions in real time is based on a *time-critical* computing algorithm and on octrees of spheres. Kitamura [18] algorithm for collision detection uses an octree for each object. Ganovelli [16] also associate an octree of axis aligned bounding boxes with each object, and keeps this hierarchy efficiently and dynamically updated for deformable objects.

Luque [19] uses semi-adjusting BSP-trees for representing scenes composed of thousands of moving objects.

Figueiredo [20] combines AABBs with R-trees to implement an efficient collision detection algorithm that determines intersecting surfaces.

Various approaches have been recently introduced using existing graphics accelerated boards (GPU) [23, 24, 25, 26] or dedicated hardware [27] to accelerate collision detection by hardware.

Algorithms using graphics hardware use depth and stencil buffer techniques to determine collisions between convex [23] and non-convex [24] objects. CULLIDE [25] is also a GPU based algorithm that uses image-space occlusion queries and OBBs in a hybrid approach to determine intersections between general models with thousands of polygons. MRC [26] deals with large models composed of dozens of millions of polygons by using the representation of a clustered hierarchy of progressive meshes (CHPM) as a LOD hierarchy for a conservative errorbound collision and as a BVH for a GPU-based collision culling algorithm.

These GPU-based algorithms are applicable to both rigid and deformable models since all the computations are made in the image-space. Collision detection methods using GPUs have the disadvantage that they compete with the rendering process, slowing down the overall frame rate. Furthermore, some of these approaches are pure image based reducing their accuracy due to the discrete geometry representation.

All these collision methods have been applied only to polygonal objects. Recently Klein [28] presented a novel approach for collision detection of point clouds. They construct a point hierarchy of bounding volumes to enclose the points at different levels of the hierarchy. Points are stored in the hierarchy leaves. Each node stores a sufficient sample of the points plus a sphere covering of a part of the surface. Given two point cloud hierarchies, two objects are tested for collision by simultaneous traversal. At the leaves, an intersection is determined by estimating the smallest distance.

Recently, Kim [31] et. al show the performance benefits of using compression of out-of-core AABBs for collision detection of polygon models that do not fit in main memory, namely they show that the resources of the CPU can be used to compensate the I/O lag of reading uncompressed data structures.

3. VIZIR

The VIZIR project sets out to develop new visualization and interaction algorithms of massive out-of-core data. The 3D model of study consists of approximately 700 laser scans of the Batalha monastery, ~2 billion points, exceeding 100 GBytes.

Collision detection is an important interaction cue to help user navigation in the virtual world. Unfortunately not much work exists with solutions for collision detection with point clouds.

Before the full complexity of the model can be addressed, an efficient and reliable collision detection solution is needed for point clouds.

For this purpose a simple scenario was designed to evaluate different user collisions that can occur whilst navigating and exploring a 3D point cloud model.

In this scenario a subset of the model was chosen that enabled the user's polygonal avatar, which is represented as a point cloud for collision detection purposes, to pass through open doors, walk alongside walls, but is stopped when colliding with the point cloud (Figure 1, 2).

In addition standard collision detection tests were carried out, and collisions with points obtained from CAD models were also tested.

In the next section we present our solution for efficiently detecting collisions with point clouds.



Figure 1: First person view of the user in the scenario of interaction whilst navigating the scanned Batalha monastery model.

4. POINT CLOUD HIERARCHY

This section presents the data structures that the proposed algorithm uses to find collisions in a large environment where 3D models are described as point clouds.

First a uniform grid that divides the scene graph into $N \times N \times N$ cubic cells of equal volume is used, thus building a grid of voxels. Each voxel has a list of the objects and the points occupying that region. To study the various user scenarios described in the previous section the voxel containing the entrance to the monastery was used (Figure 2). In a future scenario, each data structure associated with a voxel could be compressed and neighbour voxels to a user's position loaded and uncompressed into a LRU queue [32].

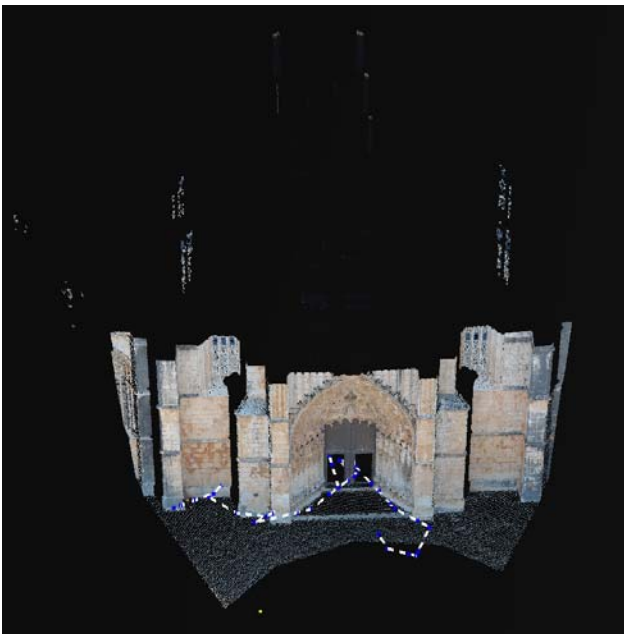


Figure 2: Walkthrough collision test scenarios between the AI avatar model comprised of 3617 points (lower left) and the 587923 point cloud belonging to a single voxel.

To determine colliding objects at each voxel, the approach presented in this paper use R-tree hierarchies of Axis-Aligned Bounding Boxes, to find collisions between pairs of 3D objects defined as clouds of points. Each object is represented by a R-tree data structure in its own local coordinate system. The R-tree hierarchy structure is used to filter out portions of the object that cannot intersect.

The choice of bounding volume type influences performance of the collision detection process. The implementation of the collision detection algorithm presented in this paper uses axis aligned bounding boxes because they are faster to intersect.

It was decided to use R-trees [30] to build bounding volume hierarchies and organize 3D geometry of objects to improving the performance of the collision detection process. R-trees are a good choice for collision detection because first, at any level of the tree, each primitive is associated with only a single node. Secondly, in an R-tree all leaf nodes appear on the same level. Third, because the depth of a R-tree storing n primitives is $\log_m n$, m is the minimum number of children of a node.

The objects of each voxel are represented by an R-tree data structure in its own local coordinate system (Figure 3) to speed up the process of finding collisions. The R-tree is built, grouping neighbouring points. The leaf nodes represent the points that define the object. For two objects, it checks for collisions between points which are in the neighbourhood, eliminating comparisons with those that are far away.

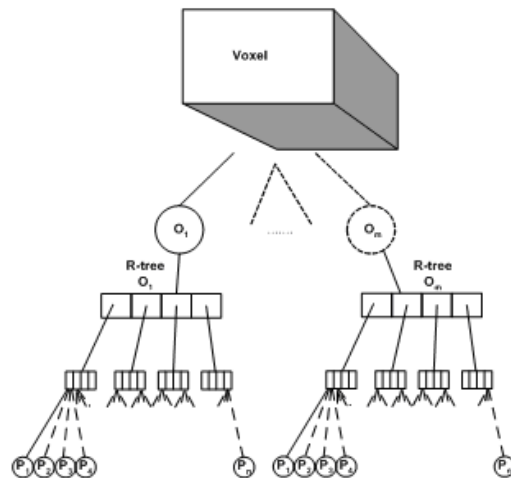


Figure 3: Every object of each voxel is an R-tree of points.

5. COLLISION DETECTION ALGORITHM

This section presents a novel algorithm for determining intersections between pairs of 3D objects defined as point clouds. The approach presented is supported by an R-tree hierarchy of axis-aligned bounding boxes and the Overlapping Axis-Aligned Bounding Box to improve performance by reducing the number of bounding volume intersections.

The collision detection approach uses axis-aligned bounding boxes for four reasons: i) they are fast to intersect; ii) use less memory; iii) hierarchies of AABBs are faster to build; and iv) faster to update.

The points of an object are organized in a hierarchical tree of bounding volumes (BV). To find if two objects are intersecting, the collision detection manager makes a recursive traversal of

two R-tree bounding volumes trees A and B . The approach presented takes advantage of the OAABB and is implemented to avoid visiting the same node several times to improve performance. It visits the nodes of object A once. The OAABB is an approach introduced by [29] to improve collision detection performance. Consider two objects, A and B , whose corresponding axis-aligned bounding boxes are overlapping and therefore are candidates for collision. The OAABB is defined as the volume that is common to two axis-aligned bounding boxes (Figure 4).

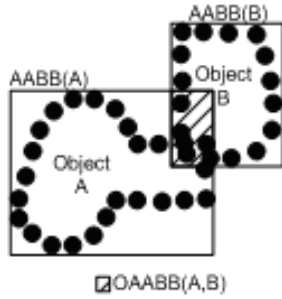


Figure 4: The OAABB is shown for two point cloud models intersecting.

Figure 5 presents the pseudo code of the novel approach.

```

Collide ( A, B )
1:  $AABB_B(A) = M_{B,A} \cdot AABB_A(A)$  //update BV
2: if ( $AABB_B(A)$  do not intersect  $AABB_B(B)$ )
   return
3: Determine  $OAABB_B(A, B)$ 
4: DescendRtree( B,  $OAABB_B(A, B)$ )
5: for each point P(B) inside  $OAABB_B(A, B)$ 
6:   Update point P(B) into coord. system of A  $E_A(B)$ 
7:   DescendRtree( A,  $OAABB_B(A, B)$ ,  $E_A(B)$ )
    
```

Figure 5: Pseudo-code for finding two intersecting objects.

The collision detection algorithm first checks if objects A and B are disjoint (line 1-2 in Figure 5). The bounding volumes of each object are originally computed in the object's local coordinate system, $AABB_A(A)$ and $AABB_B(B)$, respectively. The transformation matrix that converts the local representation of object A into the local coordinate system of object B is defined as $M_{B,A}$. The bounding volume of object A is updated to the coordinate system of object B , by computing the cover axis-aligned bounding box, $AABB_B(A)$. Once the bounding volumes of each object are in the same coordinate system they can be checked for overlap. If this pair of AABBs does not overlap, then the corresponding two objects are not intersecting and the process ends. If they overlap, then the system determines the Overlapping Axis-Aligned Bounding Box, $OAABB_B(A, B)$ of the two objects (line 3 in Figure 5), which is defined in the local coordinate system of object B .

The next step of the collision detection process determines the points from object B inside the OAABB (line 4 of Figure 5). As mentioned before, the points of object B are organized in a bounding volume R-tree. The points of B are stored at the leaf nodes of the R-tree. By descending this R-tree, the points of object B outside the $OAABB_B(A, B)$ are filtered out. Only points at the leaf nodes inside the $OAABB_B(A, B)$ are candidate for collision.

The objects of object B inside the OAABB are transformed into the coordinate system of object A , $E_A(B)$ (line 6).

Then, the collision detection algorithm descends the bounding volume R-tree for object A (line 7 of Figure 5). In this step it finds points of object A inside both the OAABB and points in close proximity of object B .

6. EXPERIMENTAL RESULTS

This section presents the performance evaluation results of the novel collision detection algorithm for point cloud models described in this paper.

6.1. Using Points from CAD Models

This section shows that the proposed collision detection for point cloud models is effective in determining collisions in real time. It is also shown that it compares favorably with other approaches that determine collisions with a model's polygons instead of with a model's vertices.

To evaluate this, two case studies were designed. The first case study, evaluates the performance of the system with a real maintenance application, with interpenetrations between 3D models. The second case study, tests the performance of the collision detection algorithm for very close proximity when there are no intersections.

The first case study represents user operations to assembly the components to build a digger mechanism (Figure 6, left). For this application, it is necessary to allow the user to interactively carry out assembly and disassembly operations on the virtual prototypes in a realistic way. The three-dimensional virtual prototypes need to simulate physical properties realistically and interactively.

The functional modules used by this application are collision detection, constraint recognition, constraint satisfaction, constraint management and constraint motion. The automatic constraint recognition process uses collision detection services for various purposes such as (a) to provide collision response to stop object penetration, (b) to identify colliding parts to support the recognition of assembly relationships between the assembly parts, (c) to simulate constrained motion, (e) to simulate kinematics motion and sliding, thus assisting users to carry out precise object manipulations.

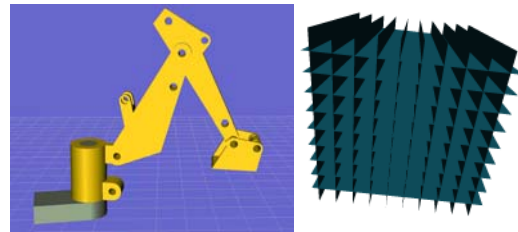


Figure 6: Test case scenario of: left) a Digger model; right) the grid scene.

The second example is a scene with two grids from a collision detection benchmark suite [31] (Figure 6, right). This benchmarking system is used to compare pairwise static collision detection algorithms for rigid objects. This benchmark generates a number of positions and orientations for a predefined distance in close proximity and no interpenetration. It does not test

performance of collision detection approaches when intersections occur.

Table 1 presents the complexity for the digger and grid case studies.

The digger scenario has five parts that are assembled in a sequence of five hundred and seven intersecting steps. At each step it is found an intersection between parts of the scenario that are recognized and assembled appropriately. This experiment was conducted by the user executing the required assembly operations. The executed path was recorded and it was repeated only the intersecting steps to obtain the data values.

The second scenario has two equal grid objects defined each one of them with forty thousand and eighty points. The benchmark generated six thousand and thirty eight steps that positioned the two objects very close to each other, but not touching each other.

With these two case studies, the performance of the novel collision detection approach for cloud point models is evaluated.

Table 1: Complexity of the case studies.

	Digger	Grid
Number of Objects	5	2
Number Points	7356	46080
Number of steps	507	6038

All the experiments run in an Intel Core 2 Duo T7300, 2GByte of RAM memory at 2GHz.

The execution times, presented in this section, include only the time to determine collisions and do not include time for rendering or motion simulation.

Table 2 presents these times. The proposed collision detection algorithm for point cloud models achieves interactive rates in real industrial applications as desired.

Table 2: Collision detection time to find intersecting surfaces.

Time in milliseconds per step to determine intersections	3D models of Point Clouds
Digger	0.03
Grid	0.21

The time to determine the collisions between two objects depends on: (1) the cost of intersecting and updating bounding volumes; and (2) on the number of such operations. Table 3 shows the number of operations executed to determine intersections. This table shows that the number of bounding volume updates is significantly lower than the number of bounding volume intersections. The update of a bounding volume is a more expensive operation than a bounding volume intersection.

Table 3: Operations per step to determine intersections.

Number of operations	Digger	Grid
AABBs tests	149	634
AABBs updates	26	131

Table 4 shows the time and the number of operations executed to run the two test cases with the same collision detection algorithm, but it does not use the OAABB concept. This table presents results for the same traversal scheme to find collisions using only R-Trees.

Table 4: Traversal scheme for collision detection using R-Trees and not using the OAABB.

	Digger	Grid
Time to find intersections (ms)	0.15	3.79
Number AABBs tests	230	4810
Number AABBs updates	174	4625

The performance of the collision detection approach proposed is better when it uses the OAABBs.

From comparison of Tables 3 and 4, it is possible to see that the number of bounding volume checks and updates is reduced significantly by the use of the OAABB.

It is also important to compare the performance of this algorithm with other collision detection systems, although public collision detection toolkits are supported by polygonal models. Table 5 presents the times obtained for the two case studies with the S-CD, PQP, RAPID, OPCODE and Dop Tree collision detection toolkits. The times presented were obtained in the determination of the first intersecting triangle.

Table 5: Time to find first triangle intersection.

Time to find first triangle intersection (milliseconds)	Digger	Grid
PQP	0.94	8.99
RAPID	0.36	6.02
OPCODE	0.08	0.61
Dop Tree	1.26	7.09
S-CD	0.25	2.29

Tables 2 and 5 shows there is an improvement in performance for the collision detection approach supported by point cloud models. This improvement can be explained by the fact that the novel approach presented in this paper is being supported by point cloud models and, in this way, it does not make triangle checks to find intersections, which is an expensive operation. For this reason, there is a difference on the number of intersections determined with a collision detection using polygonal models and the approach described in this paper. However, for the digger case study there was only 1,1% different answers reported, which is a low error probability for the new collision detection algorithm for point cloud models.

6.2. Using Points from Scanned DataSets

Collision detection algorithms designed for polygonal and CAD models can rely on the concept of collision between subset surfaces. This has the advantage that searches for instance can be faster as we are dealing with only subsets rather than the entire model. In addition since we are typically only interested in detecting the collision between surfaces, a small standard tolerance constant is used in the literature. However point clouds derived from laser scans present two main differences: they are not segmented, and points are only samples of the surface, making an actual collision between points a less likely event. Point based rendering algorithms such as QSplat [33] change the thickness and shape of a point splat to better convey visually the underlying surface while viewing in close range. Similarly we use the average closest point distance of a point divided by two to create bounding boxes at point level that ensure collision detection of the surface they represent. In addition for each voxel we use an octree to segment points into smaller working sets.

As mentioned in section 3, we created an interactive system to study various collision scenarios using the Batalha Façade Model. This model was obtained using a laser scan and contains 587923 points. The avatar is the AI model using 3617 points and walks along a predefined path of 40 seconds used to benchmark our collision algorithm and depicted with a white dashed line in Fig. 2.

We performed several partitions of the model into surfaces using an octree data-structure with different levels (4 joining cells of the first level, 8, 64, 512, 4096 cells). The resulting non-empty cells of the octree were used to create an R-tree collision structure of degree 4 per each cell. We run our experiment on a laptop equipped with a Core 2 Duo T9300 2.50 GHz Cpu, 4 Gb of RAM memory, a NVIDIA 8800M GTX graphic card with 512 Mb and running Windows 7 64 bits. Our walkthrough application was implemented in C++ using GLUT and OpenGL libraries.

Table 6, presents the results of our algorithm during the navigation of the avatar model along the predefined path.

The first column defines which partition was used for the model resulting into several surfaces (i.e. set of points organized in a R-tree) presented in the second column. We also present the memory used by the application during the path traversal and the memory size of the R-tree data structure. Finally, the average framerate of the walkthrough as well as the pre-processing time needed to create the collision structure are shown. We note that to run the application for an interactive exploration, we only need a few seconds to load the R-tree as shown in the last column of Table 6.

Table 6: Memory and Timing for several subdivision of the Batalha Model.

Model	#RTrees or Surfaces	App Mem (Mb)	Rtree Mem. (Mb)	Avg. Fps	Pre Processing Time (h:mm)	Load RTree Time(s)
Sub 4	4	170.10	49.03	18.36	4h00	4.67
Oct 8	8	175.93	59.02	19.60	2h00	6.39
Oct 64	12	175.67	56.81	19.86	1h58	6.09
Oct 512	31	173.37	52.08	19.57	1h15	5.62
Oct 4096	95	172.78	52.66	23.78	0h25	5.34

Our application was designed to run a synchronized rendering loop of 30 fps which is sufficient for desktop based real-time visualization. We can notice that our approach provides an average frame rate from 16 up to 24 fps with collision detection (Figure 7). The penalty of the collision test is strongly related with the partition of the model. We should notice that the collision test is defined between the avatar and the scene points. Our experiments have shown that creating more R-trees (one for each non-empty subdivided octree leaf node), segments the object in correspondingly more surfaces with less triangles (Table 6), however as the octree subdivision is not deep (typically level 4), very little extra memory is necessary than when using fewer R-trees with less surfaces and more triangles. More R-Trees with less triangles enable fewer run-time tests and faster average frame rate (Figure 7). In addition, the preprocessing time when using more R-Trees is significantly less as there are less triangles to consider in the subdivision and grouping steps. This marginal increase in memory makes us believe that the approach is suitable and scalable to handle large point data-sets. Figure 7 depicts the variation of the framerate along our 40 second path using the different partitions. The black dashed line corresponds to the framerate obtained with the walkthrough of the path without collision detection. Figure 7 shows that even with collision

detection the navigation is still interactive and the cost of the R-tree traversal is variable due to the spatial partition of the R-tree structure. Depending of the octree level used for the model subdivision into surfaces, the different R-tree do not have the same depth. However the collision detection is faster using smaller point sets, as each R-tree has less primitives to test (Figure 7). This is why we obtain the best results with the surface partition based on 4096 cells of the octree (orange line) with an average framerate of 24 fps. These results show that the partition of the model improves the collision test performance providing interactive collision detection with a model of 587923 points.

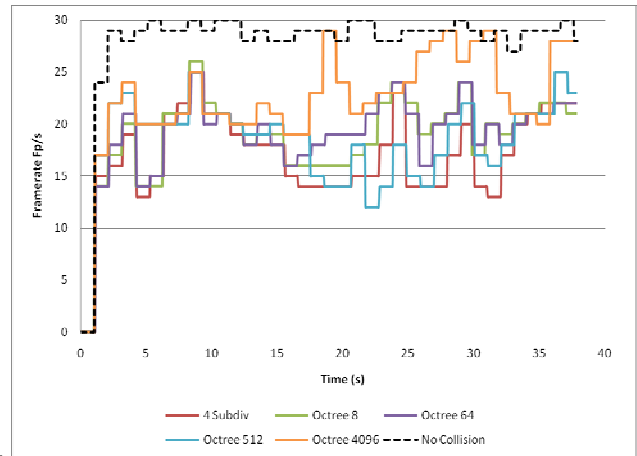


Figure 7: Application framerate when AI model is walking along the path.

7. CONCLUSIONS AND FUTURE WORK

This paper presents a novel approach for collision detection of point clouds. There are many approaches and algorithms to determine collisions between 3D polygonal models. There is very little in the literature about collision between 3D point clouds models. However, point clouds have become a popular shape representation. One of the reasons is due to the fact that 3D scanning devices became affordable and widely used for projects like VIZIR.

The proposed collision detection approach divides the scene graph in voxels. There is a bounding volume R-Tree for each object in a voxel that organizes spatially its point cloud.

To improve the sequential performance, the collision detection manager also uses the overlapping axis-aligned bounding box approach. The OAABB is used to filter out bounding volumes from two R-trees that cannot intersect. It was shown that the use of the OAABB reduces significantly the number of bounding volume checks and updates.

Experimental results show that this implementation is effective in determining interactively intersections between 3D models. In particular we show the improvements that R-trees can offer over just using AABB, we believe that these results can present benefits in an out-of-core setting, since the solution for polygon models developed by Kim et.al [31] al uses AABB.

This collision detection toolkit is publicly available for download at <http://w3.ualg.pt/~mfiguei/>.

For future work we want to integrate the collision detection manager presented in this paper in the VIZIR prototype that is

being developed and evaluate its applicability to very large environments.

In the context of laser scan data, we found that using the average closest point distance of a point divided by two to create the bounding box around each point works well in general. However the sampling density of such models is not the same everywhere thus making the underlining surface more porous for collision. We tested a more conservative approach that used the average closest point distance to ensure overlap between point boxes. This test did not interfere with the framerate. However, traversing tighter areas such as the door of the Monastery became more difficult without colliding. In the future we would like to design an adaptive bounding box size to better handle point sets with heterogeneous sampling density.

We would also like to test the performance of the collision detection algorithm using manually defined point subsets that spatially approximate more closely the underlying real world surfaces.

8. ACKNOWLEDGMENTS

The authors would like to thank Instituto de Gestão do Património Arquitectónico e Arqueológico (IGESPAR) and "Artescan, Tridimensional Digitization" for the model of the Batalha cathedral. The work presented in this paper was funded by the Portuguese Foundation for Science and Technology (FCT), VIZIR project grant (PTDC/EIA/66655/2006). In addition, Bruno Araújo would like to thank FCT for doctoral grant reference SFRH/BD/31020/2006.

REFERENCES

- [1] Oliveira, J., Oliveira A., Boavida, J. Catarino, L., Araújo, B., Pereira, J and Jorge J, 2009. "Value added 3D modelling of Laser scanned and photogrammetric data", Proc. of 17^o Encontro Português de Computação Gráfica.
- [2] Bradshaw, G. and O'Sullivan, C., 2004. "Adaptive medial-axis approximation for sphere-tree construction", *ACM Transactions on Graphics*, 23, 1–26.
- [3] Van Der Bergen, G., 1997. "Efficient Collision Detection of Complex Deformable Models using AABB Trees", *Journal of Graphics Tools*, 2, 4, 1-13.
- [4] Terdiman, P., 2001. "Memory-optimized bounding volume hierarchies", <http://www.codecorner.com/Opcode.pdf>.
- [5] Gottschalk, S., Lin, M. and Manocha, D., 1996. "Obb-tree: A hierarchical structure for rapid interference detection", *Proc. of ACM Siggraph'96*, 171-180.
- [6] Hudson, T., Lin, M., Cohen, J., Gottschalk, S. and Manocha, D., 1997. "VCollide: Accelerated Collision Detection for VRML", *Proc. of VRML*.
- [7] Larsen, E., Gottschalk, S., Lin, M. and Manocha, D. 1999. "Fast Proximity Queries with Swept Sphere Volumes", Technical report TR99-018, UNC.
- [8] Gregory, A., Lin, M.C., Gottschalk, S. and Taylor, R., 1999. "A Framework for Fast and Accurate Collision Detection for Haptic Interaction", *Proc. of the IEEE Virtual Reality*, 38–45.
- [9] Klosowski, J., Held, M., Mitchell, J., Sowizral, H. and Zika, K., 1998. "Efficient Collision Detection using Bounding Volume Hierarchies of k-DOPs", *IEEE Trans. On Visualization and Computer Graphics* 4, 1, 21-36.
- [10] Zachmann, G., 1998. "Rapid Collision Detection by Dynamically Aligned DOP-Trees", *Proc. of IEEE Virtual Reality Annual International Symposium; VRAIS*, 90–97.
- [11] Ehmann, S. and Lin, M., 2001. Accurate and fast proximity queries between polyhedra using convex surface decomposition. *Computer Graphics Forum*. 20,500–10.
- [12] He, T., 1999. "Fast collision detection using QuOSPO trees", *Proc. of the Symposium on Interactive 3D graphics*, 55–62.
- [13] Zyda, M., Osborne, W., Monahan, J. and Pratt, D., 1993. "NPSNET: Real time vehicle collisions, explosions and terrain modifications", *The Journal of Visualization and Computer Animation*, Vol. 4, No. 1, 13-24.
- [14] García-Alonso, A., Serrano, N. and Flaquer, J., 1994. "Solving the collision detection problem", *IEEE Computer Graphics and Applications*, V14, No.3, 36-43.
- [15] Zhang, D., Yuen, M., 2000. "Collision detection for clothed human animation", *Proceedings of Pacific Graphics '00*, 328–337.
- [16] Ganovelli, F., Dingliana, J., O'Sullivan, C., 2000. "BucketTree: Improving collision detection between deformable objects", *Proceedings Spring Conference on Computer Graphics SCCG '00*.
- [17] Hubbard, P. M., 1996. "Approximating polyhedra with spheres for time-critical collision detection", *ACM Trans. Graphics*, Vol. 15, No. 3, 179-210.
- [18] Kitamura, Y., Ahuja, H. and Kishino, F, 1994. "Coarse to fine collision detection for real time applications in virtual workspace", *Intern. Conference on Artificial Reality and Tele-Existence*, 147-157.
- [19] Luque, R.G., Comba, J.L.D. and Freitas, C.M.D.S., 2005. "Broad-Phase Collision Detection Using Semi-Adjusting BSP-trees", *Proc.s of the 2005 symposium on Interactive 3D graphics and games*, 179–186.
- [20] Figueiredo, M., 2009. "Surface Collision Detection for Virtual Prototyping", *Proc. of the IADIS International Conference Computer Graphics, Visualization, Computer Vision and Image Processing 2009*, 65-72.
- [21] Teschner, M., Heidelberger, B., Mueller, M., Pomeranets, D. and Gross, M., 2003. "Optimized Spatial Hashing for Collision Detection of Deformable Objects", *Proceedings of Vision, Modeling, Visualization*, 47–54.
- [22] Eitz, M., and Lixu, G., 2007. "Hierarchical Spatial Hashing for Real-time Collision Detection", *International Conference on Shape Modeling and Applications*.
- [23] Baciú, G. and Wong, S., 2003. "Image-based Techniques in a Hybrid Collision Detector", *IEEE Trans. On Visualization and Computer Graphic*, 9, 2, 254-271.
- [24] Knott, D. and Pai, D., 2003. "ClnDeR: Collision and Interference Detection in Real-time using Graphics Hardware", *Proc. of Graphics Interface 2003*, 73-80.
- [25] Govindaraju, N., Redon, S., Lin, M. and Manocha, D., 2003. "CULLIDE: Interactive collision detection between complex models in large environments using graphics hardware". *Graphics Hardware 2003*, 25–32.
- [26] Yoon, S., Salomon, B., Lin, M. and Manocha, D., 2004. "Fast Collision Detection between Massive Models using Dynamic Simplification", *Eurographics Symposium on Geometry Processing*, 136-146.
- [27] Raabe, A., Hochgurtel, S., Anlauf J. and Zachmann, G., 2006. "Space-efficient FPGA-accelerated collision detection for

virtual prototyping”, *Proc. of Design, Automation and Test in Europe*, 206-211.

[28] Klein, J., Zachmann, G., 2004. “Point cloud collision detection”, *Computer Graphics Forum (EUROGRAPHICS)*, 567–576.

[29] Figueiredo, M., Boehm, K. and Teixeira, J., 1993. “Precise Object Interactions using Solid Modeling Techniques”, *Proc. of IFIP TC 5/WG 5.10 Conference on Modeling in Computer Graphics*, 157-176.

[30] Guttman A., 1984. “R-trees: A dynamic index structure for spatial searching”, *Proc. of the ACM SIGMOD International Conference On Management of Data*, 47-57.

[31] Trenkel, S., Weller, R. and Zachmann, G., 2007. “A Benchmarking Suite for Static Collision Detection Algorithms”, *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, Czech Republic.

[32] Kim, T., Byun, Y., Kim, Y., Moon, B., Lee, S. and Yoon, S., 2010. “HCCMeshes: Hierarchical-Culling oriented Compact Meshes”, *Proc. of EUROGRAPHICS*.

[33] Rusinkiewicz, S. and Levoy, M., 2000. “QSplat: A Multiresolution Point Rendering System for Large Meshes”, *Proc. of SIGGRAPH*, 343-352.

About the author

Mauro Figueiredo is Professor at the Universidade do Algarve, Instituto Superior de Engenharia, Faro. His contact email is mfiguei@ualg.pt.

João Oliveira is completing his post-doc at the INESC-ID research institute in Lisbon. His contact email is joao.oliveira@inesc-id.pt

Bruno Araújo is a PhD Student at the Instituto Superior Técnico from Technical University of Lisbon and he is a researcher at INESC-ID in the Visualization and Intelligent Multimodal Interfaces Group. His contact email is bruno.araujo@inesc-id.pt.

João Madeiras Pereira is Associate Professor at Technical University of Lisbon (IST), Department of Computer Science. He is also the Head of Visualization and Simulation Group at the research institute INESC-ID. His contact email is jap@inesc.pt.

Advanced Mesh Repairing

Alexander Emelyanov

State Research Center "NIITeplopribor"

Moscow, Russia

aiem@rambler.ru

Abstract

This paper is devoted to the problem of the repairing incomplete reconstructed (damaged) CAD-models. It proposes a general concept of an abstract field whose force lines approximate the missing surface of a model to be repaired. This field is called the missing surface field. The main idea of the concept is that calculations of a missing surface estimation method can be expressed via tension indices of this field. These tension indices created by various estimation methods can be summed to obtain the resulting ones. Thus, within the framework of the concept various methods of missing surface estimation can be used together to supplement each other and achieve better repairing results.

Keywords: cloud of points, surface reconstruction, mesh repairing, parallel architectures

1. INTRODUCTION

Creating CAD-models from cloud of points obtained by sampling the corresponding original objects is widely used in science, culture and industry. But because of various physical and technical reasons such cloud of points often contains regions with unsatisfactory allocation of the points or without them. In this case using even a powerful and robust surface reconstruction algorithm often leads to absence of the obtained model surface in the badly sampled regions. So, the problem of repairing incomplete reconstructed (damaged) CAD-models is very topical.

A majority of recently developed repairing methods can be related to two groups. Methods of the first one (let's call it the *rebuilding group*), in general, rebuild all a model to be repaired [TJ04, EBV05, ZJH07]. But it means that they ignore the most part of the previous work to create the model. Their cost of work has weak dependence on the degree of damage of the model that leads to inefficient processing little damaged models. Methods of the second group (let's call it the *template-warping group*) use warping of a suitable template from a database to repair a given damaged model region [ACP03, PMG05, SKR06]. They show impressive results, but are usable only for models corresponding to their template databases. Also, to provide proper fitting of a chosen template to the corresponding damaged model region such methods require manually setting a certain number of point matches between them.

Each existing repairing method is based on the corresponding method of missing surface estimation. In spite of the fact that even a heavily damaged CAD-model contains diverse kinds of information about its missing surface (the behavior of the reconstructed one, location of unused sampled points, a

supposition of symmetry, etc) such estimation method uses only the corresponding restricted parts of it. So, using this information entirely promises significantly increasing the repairing quality with comparison of existing methods. But it is clear that developing one "universal" estimation method with this property is impossible. Thus, we have to develop a concept that provides simultaneously using several different missing surface estimation methods with possibility of these methods to supplement each other to achieve a better repairing result. The presented paper is a continuation of our previous works in the repairing theme [EM04, EAK08]. Unlike of them, in this paper the required concept is proposed. It is a general concept of an abstract field whose force lines approximate the missing surface of a model to be repaired. This field is called the *missing surface field*. The main idea of the concept is that calculations of a missing surface estimation method can be expressed via tension indices of this field. These tension indices created by various estimation methods can be summed to obtain the resulting ones. It provides the mentioned mutual supplementing of the methods. But simultaneously using several estimation methods significantly increases the cost. More generally, possibilities of the existing repairing methods are restricted by high cost of powerful missing surface estimation ones. Only way to solution this cost problem is parallelization of the repairing process using modern hardware. Its abilities allow using approaches and algorithms which even in a near past were considered extremely costly. A presented repairing method based on the missing surface field concept is an example of it. The method uses our previously developed algorithms, which are supplemented by several new quite costly ones to increase the repairing quality and robustness. But, using these algorithms significantly decreases the performance. From the architecture point of view the method implementation provides the open architecture principle that allows do develop new missing surface estimation methods by remote teams and at the same time decreases the performance as well. But good parallelization potential of the concept with the modern hardware abilities allows keeping the repairing time at an acceptable level. In general, the presented method is oriented to use on multi-processor platforms that is in agree with the modern trends in computer hardware.

The paper is organized in the following way. In the next three sections the theoretical basis of our work is described. In section 2 a formalization of assumed input is made. Then there is a description of two theoretical concepts. Section 3 explains the previously developed [EM04] *concept of bridges*. This concept considers reducing a heavily damaged model to a model that is easy to repair. In section 4 the missing surface field concept is described. In section 5 there is a description of two done algorithmic implementations of the field. Section 6 is devoted to a general overview of the developed repairing method. Some results and a discussion are in sections 7, 8.

2. FORMALIZATION OF DAMAGED CAD-MODELS

Consider a CAD-model that is an unsuccessful result of work of some surface reconstruction algorithm and has the following properties: the model contains a partially reconstructed surface (\mathbf{A}), with condition that each part of this surface represents the corresponding part of the original object; the remaining part of the model surface ($\bar{\mathbf{A}}$) is missing; the model can contain unused sampled points (*free points*). Let's call such model an *incomplete CAD-model (ICADM)*. In general case, to obtain it, we need to apply a proper filtering procedure that removes all parts of \mathbf{A} which don't represent the original object surface. Just an ICADM is the assumed input in our works.

In an ICADM \mathbf{A} can be represented by one or several isolated coherent regions. Let's call such regions *islands*.

Any ICADM contains one or several closed boundaries which separate \mathbf{A} and $\bar{\mathbf{A}}$. Each boundary has connection with one and only one region of $\bar{\mathbf{A}}$. If this region is connected only with the same boundary let's call such boundary a *hole boundary* or a *complex boundary (CB)* otherwise. As a consequence, the aggregate of a hole boundary and the region of $\bar{\mathbf{A}}$ that is connected with the boundary is called a *hole*.

Let's call a hole a *trivial hole (TH)* if its region of $\bar{\mathbf{A}}$ can be reconstructed by an existing not so complicated algorithm or a *complicated hole (CH)* otherwise. A boundary that is not the boundary of a trivial hole will be called a *non-trivial boundary*.

Using the terms defined above the following classification of ICADMs is made:

class	num. of islands	CB	CH	TH
ICM3	>1	y	y/n	y/n
ICM2	1	n	y	y/n
ICM1	1	n	n	y
ICMT	1	y	y/n	y/n

Note, that an ICADM can belong to class ICMT only if the corresponding original object is not a topological equivalence of a sphere and has one or several internal "tunnels" (for example, a torus-like object) which don't exist in the ICADM. In addition, each missing "tunnel" should be represented by the corresponding pair of complex boundaries and the both boundaries should belong to the same island. Such situation rarely occurs in practice, so this class is defined outside the main class sequence and is not widely considered in this paper.

3. CONCEPT OF BRIDGES

From the contents of the previous section it follows that obtaining a correct CAD-model from an ICADM of class ICM1 is not a problem. So, in general case we need a way to reduce an ICADM of class ICM3 to an ICADM of class ICM1 via intermediate ICADMs of classes ICMT (if this class is possible) and ICM2. To make it, it is enough to reconstruct $\bar{\mathbf{A}}$ only in properly chosen strip-like regions connecting the corresponding boundaries of \mathbf{A} (F3.1).

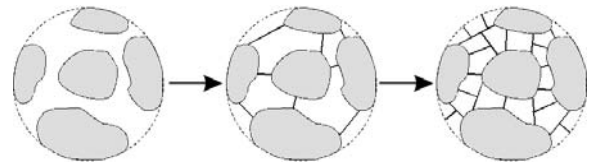


Figure F3.1

Diminishing the width of such strip region to 0 we obtain a curve connecting the two corresponding boundary points of \mathbf{A} ; at each internal point of the curve the normal vector to $\bar{\mathbf{A}}$ is defined (F3.2). Let's call such curve a *bridge*.

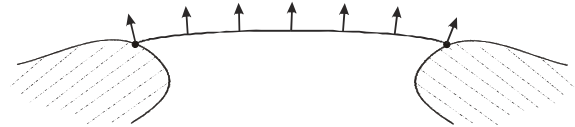


Figure F3.2

Figure F3.1 illustrates how the concept of bridges works. In the left there is a beginning ICADM of class ICM3 representing a ball-like object. Initially, using bridges we reduce it to an ICADM of class ICM2 (middle). This ICADM contains only holes, each hole boundary is formed by the corresponding bridges and fragments of island boundaries. Then using bridges again each complicated hole of the ICADM is recursively decomposed until a set of only trivial holes is obtained (right).

In this way bridges create an irregular mesh approximating $\bar{\mathbf{A}}$, inside each cell of the mesh $\bar{\mathbf{A}}$ can be reconstructed by a simple existing method (see F3.3).

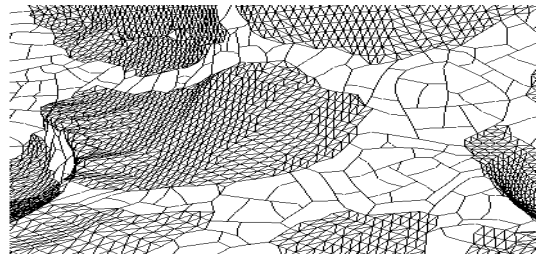


Figure F3.3

Considering that a bridge approximates the corresponding surface strip of a finite width (the width depends on a specified approximation precision) we can conclude that using the concept of bridges a major share of $\bar{\mathbf{A}}$ is reconstructed the simple method. So, using the concept significantly reduces the total repairing cost.

4. MISSING SURFACE FIELD CONCEPT

In the beginning, note that $\bar{\mathbf{A}}$ can be reconstructed only with some probability to be correct and the task of repairing a given ICADM has infinity number of acceptable solutions.

Let's suppose that in an area containing a considered ICADM there is a field of some nature. It can be considered as an instance of the missing surface field ($\bar{\mathbf{A}}F$) introduced above, if it provides obtaining the following indices at each point of the area:

- a scalar value Ψ (*potential*), such that at a specified point (X) and a point (Y) in its neighborhood $\Psi_X > \Psi_Y$ if the probability of that $\bar{\mathbf{A}}$ passes through X is greater than the same probability for Y ; non-positive values of Ψ correspond to cases when this possibility is equal to 0;
- a vector \mathbf{F} (*force*), that indicates the likeliest direction of a bridge that passes through (incidents from) the point; the length of the vector expresses the probability of the direction to be correct in the same manner like the value of Ψ expresses the corresponding probability;
- a scalar value Ω (*attraction index*) that indices the value of the attraction component of \mathbf{F} , $\Omega \leq |\mathbf{F}|$, because in general case the force consists of its attractive and repulsive components;
- a vector \mathbf{N} (*normal*) that indicates the likeliest direction of the normal to $\bar{\mathbf{A}}$ passing through the point at the point; the length of the vector expresses the probability of its direction to be correct in the same way like the length of \mathbf{F} expresses the corresponding probability.

Of course, if $\Psi=0$ (for example, at a point $\in \mathbf{A}$) then $|\mathbf{F}|=0$, $\Omega=0$ and $|\mathbf{N}|=0$ as well.

From the programming point of view the missing surface field can be represented by the following interface:

```
interface IMissingSurfaceField
{
virtual void set_point(const VECTOR& point);

virtual real get_potential() const;
virtual void get_force(VECTOR& force) const;
virtual real get_attraction_index() const;
virtual void get_normal(VECTOR& normal) const;
};
```

Each instance of $\bar{\mathbf{A}}\mathbf{F}$ is implemented on the base of the corresponding missing surface estimation method. To determine the $\bar{\mathbf{A}}\mathbf{F}$ indices this method can require the corresponding set of extra parameters that is called the *state context* of the method.

Let's call an instance of $\bar{\mathbf{A}}\mathbf{F}$ a *complete* $\bar{\mathbf{A}}\mathbf{F}$, if it provides obtaining all the $\bar{\mathbf{A}}\mathbf{F}$ indices at a point or an *incomplete* $\bar{\mathbf{A}}\mathbf{F}$ otherwise. An $\bar{\mathbf{A}}\mathbf{F}$ can be implemented as a linear composition of several (k) other $\bar{\mathbf{A}}\mathbf{F}$ s with the corresponding weights (w):

$$\bar{\mathbf{A}}\mathbf{F} = \sum_{i=1}^k w_i \bar{\mathbf{A}}\mathbf{F}_i, \text{ it means that } \Psi = \sum_{i=1}^k w_i \Psi_i, \mathbf{F} = \sum_{i=1}^k w_i \mathbf{F}_i, \\ \Omega = \sum_{i=1}^k w_i \Omega_i, \mathbf{N} = \sum_{i=1}^k w_i \mathbf{N}_i \text{ (E4.1).}$$

Let's call such $\bar{\mathbf{A}}\mathbf{F}$ a *composite* $\bar{\mathbf{A}}\mathbf{F}$. Components of a complete composite $\bar{\mathbf{A}}\mathbf{F}$ can be incomplete $\bar{\mathbf{A}}\mathbf{F}$ s, with condition that obtaining each $\bar{\mathbf{A}}\mathbf{F}$ index is provided by at least one of its components.

Thus, the concept of $\bar{\mathbf{A}}\mathbf{F}$ provides joint using various missing surface estimation methods to achieve better repairing result.

In general, the relationship between the concept of $\bar{\mathbf{A}}\mathbf{F}$ and the concept of bridges is similar like the relationship between dynamics and kinematics in physics. Using $\bar{\mathbf{A}}\mathbf{F}$ we can answer to the question: "what probability of that a correct bridge passes through a specified point and what the trajectory of this bridge?" Presence of the force index allows considering $\bar{\mathbf{A}}\mathbf{F}$ as a force field and bridges as its force lines. But there is one fine point. From the definitions of the potential and the force vector it follows that the force vector should be collinear with the potential gradient (except several degenerated cases). But if in a used instance of $\bar{\mathbf{A}}\mathbf{F}$ the force vector and the potential value are determined by different estimation methods (for example, if this instance is a composite $\bar{\mathbf{A}}\mathbf{F}$), then these vectors can be non-collinear in general. To keep adequate force line behavior in this situation, assume that a force line of $\bar{\mathbf{A}}\mathbf{F}$ is a curve such that at each point of it: all the $\bar{\mathbf{A}}\mathbf{F}$ indices are defined; the attraction factor isn't less than a specified threshold; the tangent vector to the curve is directed in accordance with the force vector and has a restricted deviation from it in the plane defined by the force and the normal vectors to be as close as it is possible to the potential gradient vector.

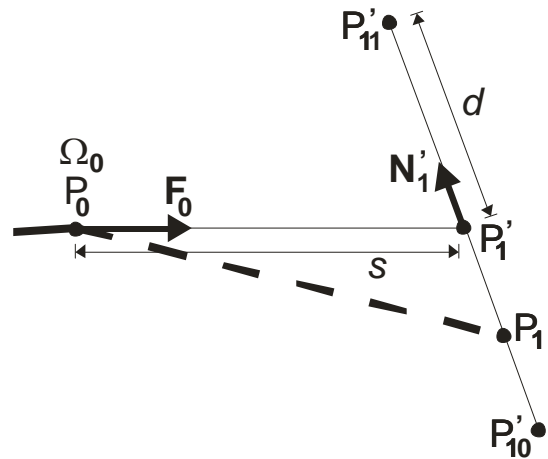


Figure F4.1

From this definition it follows that any $\bar{\mathbf{A}}\mathbf{F}$ force line starts and ends at the corresponding boundary points of \mathbf{A} . A force line can be iteratively traced from a specified boundary point until the corresponding opposite boundary point is reached. Each tracing iteration is performed in accordance with the following algorithm (F4.1).

At a specified beginning point (P_0) the attraction factor value (Ω_0) is determined. If this value is less than a specified threshold then the tracing process is terminated as unsuccessful. Otherwise the force vector (\mathbf{F}_0) is determined at the point. Using this vector and a specified tracing step (s) the first approximation of the next force line point (P_1) is obtained. At this point the acting normal vector (\mathbf{N}_1) is determined. Using this vector and a specified deviation limit (d) the segment of the next point searching (P_{10}, P_{11}) is determined. The point of the potential maximum on this segment is assumed as the next force line point (P_1).

In this way each force line can be traced independently on other ones. Moreover, the $\bar{\mathbf{A}}\mathbf{F}$ indices at a point can be determined independently on these indices at other points. It provides good parallelization abilities of implementations which use the $\bar{\mathbf{A}}\mathbf{F}$ concept.

In the end of this concept explanation let's define the *quality index* (q) of an $\bar{\mathbf{A}}\mathbf{F}$ force line:

$$q = \frac{1}{L} \int \Psi(l) dl \quad (\text{E4.2})$$

where L is the total force line length.

5. IMPLEMENTED MISSING SURFACE FIELDS

5.1 General background

Described in this section implementations of $\bar{\mathbf{A}}\mathbf{F}$ are developed in a “physical” manner, so they have several common traits adduced below.

For each field the following two kinds of objects are defined: an elementary source and an object of the field action. The both kinds of objects are represented by single points supplemented by the corresponding sets of extra features (such set can be empty). An elementary source acts in the corresponding way on an object of action.

The action of an elementary source decreases with increasing the distance between it and a considered object of action. In each of the $\bar{\mathbf{A}}\mathbf{F}$ implementations described below this property is implemented by the corresponding distance function of the following kind:

$$\lambda(r) = 1 \text{ if } r \leq r_0, \lambda(r) = 0 \text{ if } r \geq r_1,$$

$$\lambda(r) = \left(\frac{r_1 - r}{r_1 - r_0} \right)^p \text{ if } r_0 < r < r_1 \quad (\text{E5.1.1})$$

where r_0, r_1, p are positive constants, $0 < r_0 < r_1$.

\mathbf{A} has a shielding effect on $\bar{\mathbf{A}}\mathbf{F}$. This effect currently implemented in the following way: an elementary source doesn't act at a specified point if the segment between this point and the point of the source crosses \mathbf{A} .

5.2 Boundary interpolation missing surface field

To determine the $\bar{\mathbf{A}}\mathbf{F}$ indices the implementation of $\bar{\mathbf{A}}\mathbf{F}$ described here uses an interpolation of boundaries of \mathbf{A} . Because of that it is called the *boundary interpolation missing surface field* ($BI\bar{\mathbf{A}}\mathbf{F}$).

Initially, let's consider the basic geometric issues. Consider a boundary of \mathbf{A} , a point (O) on it, the normal vector to \mathbf{A} at the point (\mathbf{n}^O), and the tangent line to the boundary at the point (F5.2.1). The tangent line splits the plane defined by O and \mathbf{n}^O to the “occupied” and the “empty” half-planes. Define the unit vector ($\boldsymbol{\tau}^O$) on the tangent line such that the cross product $\mathbf{n}^O \times \boldsymbol{\tau}^O$ is in the “occupied” half-plane. Let's call such vector the *tangent vector* of a specified boundary point.

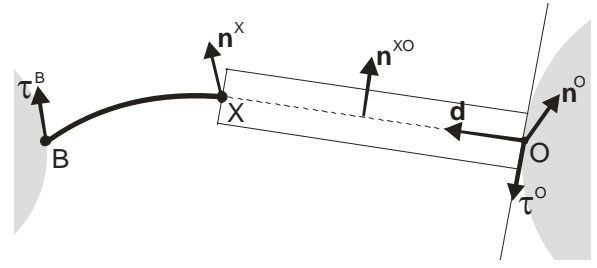


Figure F5.2.1

Consider connection of some point outside \mathbf{A} (X) with \mathbf{A} at O by an arbitrary narrow planar strip. It is obvious, that the plane of this strip should pass through X , O and the boundary tangent line at O .

From the vector cross-product properties it follows that the normal vector (\mathbf{n}^{XO}) of this strip is defined by the following equation:

$$\mathbf{n}^{XO} = \frac{\mathbf{s}^{XO}}{|\mathbf{s}^{XO}|} \quad (\text{E5.2.1})$$

$$\text{where } \mathbf{s}^{XO} = \mathbf{d}^{XO} \times \boldsymbol{\tau}^O, \mathbf{d}^{OX} = \frac{\overrightarrow{OX}}{|\overrightarrow{OX}|}.$$

Now, assuming that a normal vector at X (\mathbf{n}^X) is specified, let's define the *quality* (η^{XO}) of the considered strip connection in the following way:

$$\eta^{XO} = |\mathbf{s}^{XO}| \left(\frac{\mathbf{n}^O \mathbf{n}^{XO} + 1}{2} \right) \left(\frac{\mathbf{n}^X \mathbf{n}^{XO} + 1}{2} \right) \quad (\text{E5.2.2})$$

In this formulation the first multiplier indicates the “passability” of the strip; the second one expresses the degree of extrapolation of \mathbf{A} at O by the strip; the third expresses the degree of extrapolation of an arbitrary small surface element defined by X and \mathbf{n}^X by the strip.

Now, let's define the introduced above field. As an elementary source of the field the aggregate of a boundary point of \mathbf{A} (O), the normal (\mathbf{n}^O) and the tangent vectors ($\boldsymbol{\tau}^O$) at the point ($\{O, \mathbf{n}^O, \boldsymbol{\tau}^O\}$) is assumed. As an object of the field action let's assume the last traced point of a force line with the origin at the corresponding boundary point (B). That is the aggregate of a space point (X), a specified normal vector at it (\mathbf{n}^X) and the tangent vector at the corresponding force line origin ($\boldsymbol{\tau}^B$): $\{X, \mathbf{n}^X, \boldsymbol{\tau}^B\}$. With these assumptions define the force (\mathbf{F}^X) of action of $\{O, \mathbf{n}^O, \boldsymbol{\tau}^O\}$ on $\{X, \mathbf{n}^X, \boldsymbol{\tau}^B\}$ in the following way:

$$\mathbf{F}^X = -c \lambda^{BI} (|XO|) \eta^{XO} \mathbf{d} \text{ if } \boldsymbol{\tau}^B \boldsymbol{\tau}^O < 0$$

$$\mathbf{F}^X = 0 \text{ if } \boldsymbol{\tau}^B \boldsymbol{\tau}^O \geq 0 \quad (\text{E5.2.3})$$

where constant c expresses the length of the boundary segment represented by O (in other words it is a “charge value” of the elementary source); λ^{BI} is a distance function of the kind introduced by (E5.1.1).

It also can be written in the following matrix form:

$$\mathbf{F}^X = \mathbf{H}^X \mathbf{n}^X \quad (\text{E5.2.4})$$

where

$$H^X = \frac{1}{2} m \begin{bmatrix} d_x^{OX} n_x^{XO} & d_x^{OX} n_y^{XO} & d_x^{OX} n_z^{XO} & d_x^{OX} \\ d_y^{OX} n_x^{XO} & d_y^{OX} n_y^{XO} & d_y^{OX} n_z^{XO} & d_y^{OX} \\ d_z^{OX} n_x^{XO} & d_z^{OX} n_y^{XO} & d_z^{OX} n_z^{XO} & d_z^{OX} \end{bmatrix}$$

$$m = c |s^{XO}| \left(\frac{\mathbf{n}^O \mathbf{n}^{XO} + 1}{2} \right).$$

When the field at a point is created by a number of sources, this formulation allows obtaining the force vector for various \mathbf{n}^X without recalculation of the matrix.

The $\bar{\mathbf{A}}\mathbf{F}$ normal vector (\mathbf{N}^X), the potential (Ψ^X) and the attraction index (Ω^X) created at X by action of $\{O, \mathbf{n}^O, \boldsymbol{\tau}^O\}$ on $\{X, \mathbf{n}^X, \boldsymbol{\tau}^B\}$ are defined in the following way:

$$\mathbf{N}^X = m \mathbf{n}^{XO} \quad (\text{E5.2.5})$$

$$\Psi^X = \mathbf{N}^X \mathbf{n}^X \quad (\text{E5.2.6})$$

$$\Omega^X = |\mathbf{F}^X| \mathbf{n}^X \mathbf{n}^{XO} \quad (\text{E5.2.7})$$

The defined field is a complete $\bar{\mathbf{A}}\mathbf{F}$ because it provides obtaining all the $\bar{\mathbf{A}}\mathbf{F}$ indices at a point; vectors \mathbf{n}^X and $\boldsymbol{\tau}^B$ are considered as parameters of the state context of the field.

5.3 Point radial missing surface field

This implementation of $\bar{\mathbf{A}}\mathbf{F}$ is called the *point radial missing surface field* ($PR\bar{\mathbf{A}}\mathbf{F}$). Its elementary source is a free point of a specified ICADM. An object of the field action is just a space point. This field is incomplete, because it provides obtaining only the potential value at a specified point. This potential, created by a free point (O) at a specified space point (X) is defined by the following equation:

$$\Psi^X = c \lambda^{PR}(|XO|) \quad (\text{E5.3.1})$$

where $c \in (0,1]$ is the confidence value of the free point coordinates; λ^{PR} is a distance function of the kind introduced by (E5.1.1).

This $\bar{\mathbf{A}}\mathbf{F}$ can increase the adequateness of force line behavior of a composite $\bar{\mathbf{A}}\mathbf{F}$ with its participation, if a processed ICADM contains a sufficient number of free points.

6. THE IMPLEMENTED METHOD OVERVIEW

Using the theoretical basis described above a repairing method has been developed and implemented. It uses a composite $\bar{\mathbf{A}}\mathbf{F}$ that currently consists of $BI\bar{\mathbf{A}}\mathbf{F}$ and $PR\bar{\mathbf{A}}\mathbf{F}$. The basic steps of the method are described in the next three subsections. These steps differ from each other in determining the characteristic value of \mathbf{A} discontinuity (it used to adjust the distance function parameters of the $\bar{\mathbf{A}}\mathbf{F}$ s) and in determining the set of used elementary sources of $BI\bar{\mathbf{A}}\mathbf{F}$. The set of used elementary sources of $PR\bar{\mathbf{A}}\mathbf{F}$ for these steps is the same and consists of all free points of a processed ICADM.

In the last subsection the basic properties of the method are considered.

6.1 Connection of islands

In the input of this step an ICADM of class ICM3 is assumed. Initially, for each island a set of bridges connecting it with the other ones is obtained in the following sequence:

1. the averaged inter-island span in a neighborhood of the island is taken as the characteristic value of \mathbf{A} discontinuity;
2. the set of considered $BI\bar{\mathbf{A}}\mathbf{F}$ sources is defined as the set of all boundary points of all the other islands;
3. at all boundary points of the island the $\bar{\mathbf{A}}\mathbf{F}$ attraction index is determined, and then the points of local maximums of the index are selected;
4. from the selected points the corresponding $\bar{\mathbf{A}}\mathbf{F}$ force lines are traced, these force lines are considered as the required bridges.

After processing all the islands, the total graph of connections is obtained. In general case this graph can contain topologically conflicting bridges. Each detected conflict situation is resolved by eliminating the necessary number of bridges with the least values of the force line quality (see E4.2).

Then, in accordance with the concept of bridges, the corresponding set of holes is extracted from the graph.

6.2 Detecting and processing lost “tunnels”

If initially, or after the previous step, the processed ICADM can be referred to class ICMT and a case of this class is admitted a priori, then this step is performed. The used algorithm allows detecting and correctly processing an internal “tunnel” if it is represented by the corresponding pair of non-trivial boundaries.

At this step the averaged distance between all non-trivial boundaries is taken as the characteristic value of \mathbf{A} discontinuity. All points of these boundaries are considered as sources of $BI\bar{\mathbf{A}}\mathbf{F}$. At the same points the local $\bar{\mathbf{A}}\mathbf{F}$ attraction index is determined and then from the points of local maximums of the index the corresponding force lines are traced. When it is done, two of the boundaries are considered connected by an internal “tunnel” if two or more force lines connect them and each of these force lines has the quality higher than the quality of any other force line of these boundaries. In this case these force lines are considered as bridges. The finally obtained graph of connections is processed in the same way like in the previous step.

6.3 Decomposition of complicated holes

At this step the input consists of one coherent island with one or several complicated holes. Each of them is processed separately from other ones. Initially, the averaged size of the processed hole is taken as the characteristic value of \mathbf{A} discontinuity. Then, considering all boundary points of the hole as sources of $BI\bar{\mathbf{A}}\mathbf{F}$, the boundary point with the maximal value of the $\bar{\mathbf{A}}\mathbf{F}$ attraction index is determined. From this point the corresponding force line is traced. It splits the hole to two new ones. This algorithm is recursively applied until a set of only trivial holes is obtained.

6.4 General properties

In the software realization of the method, the declared above open architecture principle, when each $\bar{\mathbf{A}}\mathbf{F}$ instance interacts with the system kernel via the interface introduced in section 4, is implemented. It gives to the system flexibility and extensibility;

also it allows developing new $\overline{\mathbf{A}}\mathbf{F}$ implementations by remote teams of programmers.

Because the presented method works directly in regions of damage, its cost does not essentially depend on the square of \mathbf{A} and as a consequence on the number of edges, triangles and points in the input. But it has a great dependence on the square and the topology of $\overline{\mathbf{A}}$. Such dependence can't be expressed by one simple cost equation, at the same time a rough approximation can be obtained. Let's use the fact that traced $\overline{\mathbf{A}}\mathbf{F}$ force lines finally forms a mesh of trivial holes (F3.3). Assuming for simplicity that all the force lines are traced with the same step and all the trivial holes have the same square we can conclude that the number of points, at which the $\overline{\mathbf{A}}\mathbf{F}$ indices should be calculated, is proportional to the total square of $\overline{\mathbf{A}}$. Assuming that the $\overline{\mathbf{A}}\mathbf{F}$ indices calculation cost at a point is constant as well, the cost behavior can be approximated by the following expression:

$$t = k\text{Square}(\overline{\mathbf{A}}) \quad (\text{E6.4.1})$$

where k is a positive constant.

It is the same dependence, like in our previous works [EM04, EAK08]. But unlike of them the actual cost value is essentially greater, because the presented method uses the quite costly algorithms described above. For example, the used force line tracing algorithm, when at each step the $\overline{\mathbf{A}}\mathbf{F}$ potential maximum is found, is 6-8 times costlier than the previous one. Modeling the shielding property (see 5.1) and the open architecture implementation increase the cost as well.

Fortunately, the base $\overline{\mathbf{A}}\mathbf{F}$ concept has good parallelization potential that allows compensating the mentioned above cost increasing. In the current implementation the following most costly operations are parallelized: the $\overline{\mathbf{A}}\mathbf{F}$ attraction index calculation along boundaries; the force line tracing.

7. TESTS AND COMPARISON

Tests discussed in this section were performed using the following composite $\overline{\mathbf{A}}\mathbf{F}$:

$$\overline{\mathbf{A}}\mathbf{F} = 0.3\text{BI}\overline{\mathbf{A}}\mathbf{F} + 0.7\text{PRA}\overline{\mathbf{A}}\mathbf{F} \quad (\text{E7.1})$$

Figure F7.1 illustrates a benefit of using this $\overline{\mathbf{A}}\mathbf{F}$ with the advanced $\overline{\mathbf{A}}\mathbf{F}$ force line tracing. An ICADM obtained by damaging the well known "Bunny" (a) was chosen to test. The previous tracing algorithm [EAK08] provides the close-fitting reconstructed surface behavior (b). Approximately the same result is provided by the presented tracing algorithm using only $\text{BI}\overline{\mathbf{A}}\mathbf{F}$. But using this algorithm with the composite $\overline{\mathbf{A}}\mathbf{F}$ allows achieving the significantly better repairing quality (c).

Information to compare the costs of the presented method, the previous one [EAK08] and our implementation of [EBV05] (*shrinking membrane* algorithm) is added in the table below:

method	num. of proc.	time (s)
Shrink. membrane	1	120
[EAK08]	1	90
This	1	840
This	2	470
This	8	110

All the tests were performed using the same damaged "Bunny" as the input. The presented method provides the best quality and shows a comparable time when it is performed on an 8 processor hardware platform.

Ability of the method to process heavily damaged ICADMs is shown in F7.2. Among available algorithms no one can process this sample correctly except our previous [EAK08], but its result is worse in fine details like in the test shown in F7.1.

8. CONCLUSION

The done tests demonstrate that further increasing of the robustness and the repairing quality is possible. But it requires using more and more costly missing surface estimation methods. So, parallelization of computation is only way to increase the repairing quality keeping acceptable performance indices. The introduced $\overline{\mathbf{A}}\mathbf{F}$ concept provides good potential of development in the field of new missing surface estimation methods using and in the field of parallelization as well. As a particular, it seems perspective to supplement the done implementations of $\overline{\mathbf{A}}\mathbf{F}$ by an implementation that uses some database of samples and a neural network. The "physical" style of the $\overline{\mathbf{A}}\mathbf{F}$ concept corresponds to the abilities of modern GPU. Using them promises a great benefit.

The method has the proved ability to repair heavily damaged CAD-models. In general, the repairing can be performed without any manual management. At the same time, fine adjustment of some parameters (for example, the distance function power factors) allows to obtain a better result. Considering the repairing process quality as a function of these parameters, this adjustment can be made by global optimization methods using PC-clusters.

9. REFERENCES

- [ACP03] B. Allen, B. Curless, Z. Popovic. The Space of Human Body Shapes. Reconstruction and Parameterization from Range Scans. ACM Transactions on Graphics, Vol. 22, No. 3, pp. 587-594, 2003.
- [EM04] A. Emelyanov. Surface reconstruction from clouds of points. PhD Thesis. Department of Computer Science, University of West Bohemia, Plzen, 2004.
- [TJ04] Tao Ju. Robust Repair of Polygonal Models. Proceedings of ACM SIGGRAPH, ACM Transactions on Graphics, 23(3), pp. 888-895, 2004.
- [EBV05] J. Esteve, P. Brunet, A. Vinacua. Approximation of a Variable Density Cloud of Points by Shrinking a Discrete Membrane. Computer Graphics Forum, Volume 24 Issue 4, pp. 791-807, 2005.
- [PMG05] M. Pauly, N. J. Mitra, J. Giesen, M. Gross, L. Guibas. Example-based 3D Scan Completion. Symposium on Geometry Processing, pp.23-32, 2005.
- [SKR06] C. Stoll, Z. Karni, C. Rössl, H. Yamauchi, H.-P. Seidel. Template Deformation for Point Cloud Fitting. Eurographics Symposium on Point-Based Graphics, pp. 27-35, 2006.
- [ZJH07] Q-Y. Zhou, T. Ju, S-M. Hu. Topology Repair of Solid Models Using Skeletons. IEEE Transactions on Visualization and Computer Graphics, 13(4), pp. 675-685, 2007.

[EAK08] A. Emelyanov, Y. Astakhov, S. Klimenko. Estimation of Interpolation Properties of a Point. Proceedings of the 2008 International Conference on Cyberworlds. pp. 461-466, 2008.

About the author

Alexander Emelyanov is a senior scientist of state research center “NIITeplopribor”, Moscow. His contact email is aiem@rambler.ru.

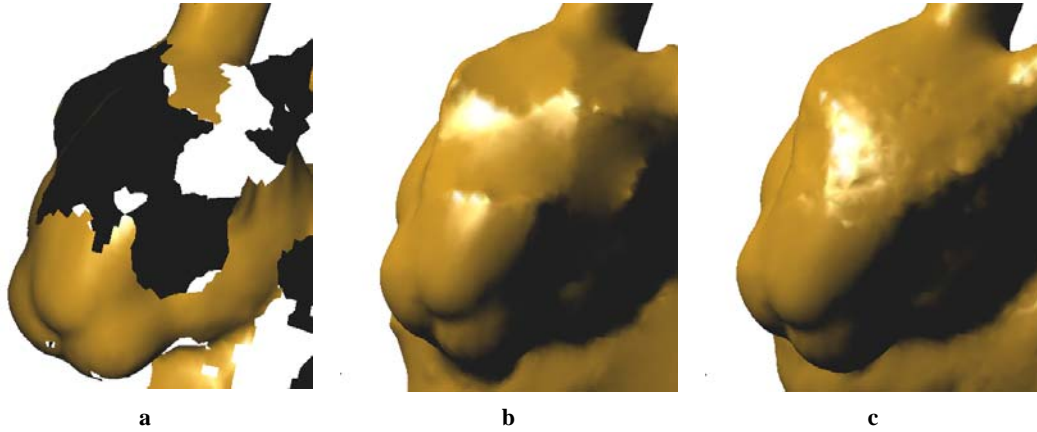


Figure F7.1: The presented method with comparison of [EAK08]

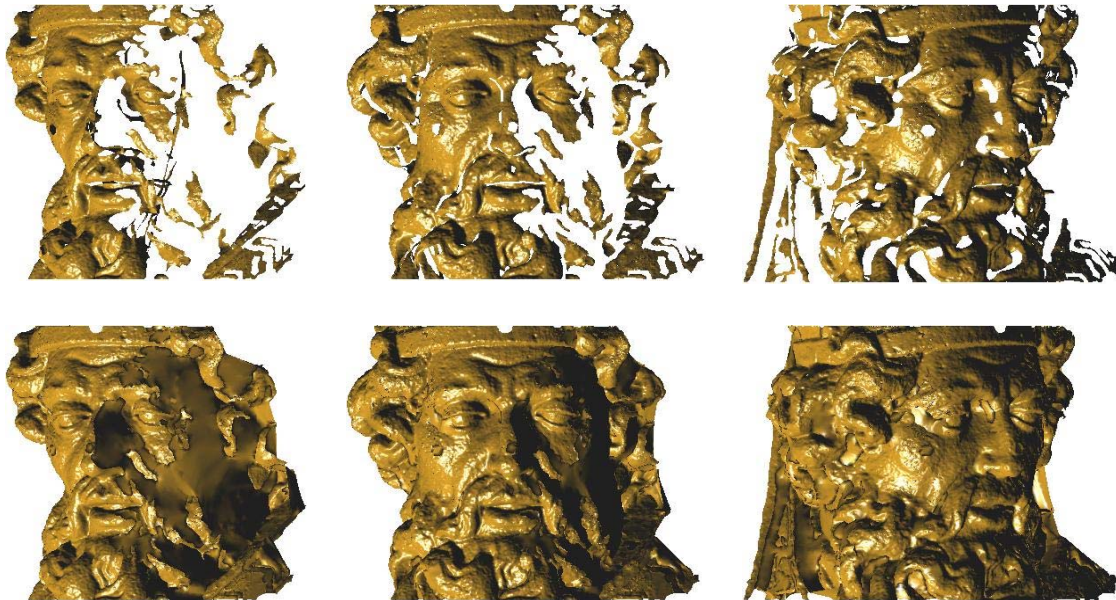


Figure F7.2: A poorly scanned sculpture fragment repairing

Low Cost Adaptive Anti-Aliasing for Real-Time Ray-Tracing

Maxim Shevtsov, Mikhail Letavin and Alexey Rukhlinskiy
 Intel Corporation
 Nizhniy Novgorod, Russia
 {maxim.y.shevtsov, mikhail.letavin, alexey.v.rukhlinskiy}@intel.com

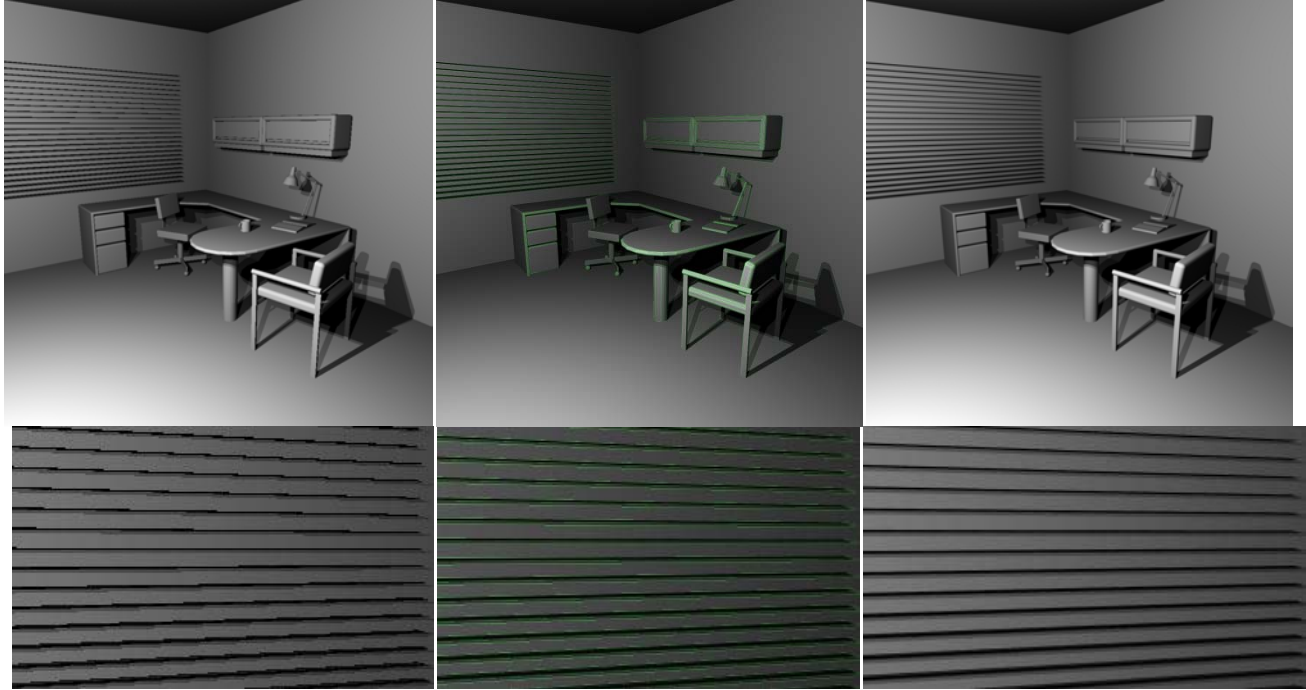


Figure 1: Evaluation of the proposed adaptive anti-aliasing scheme in the context of simple packet ray-tracing pipeline. Left is no-AA image (exactly one eye/shadow ray per pixel). In the middle additional samples are shown (notice how shadow boundary is also detected for additional sampling). Right image is anti-aliased result. By computing additional samples only where image-space color gradient is high, we significantly save computations. The resulting quality is identical to the 8X super-sampling, with only doubling rendering time in compare to no AA version.

Abstract

This paper describes simple, practical scheme for adaptive anti-aliasing, particularly suitable for packet style ray tracing. The sampling patterns are organized in a SIMD-friendly fashion. The technique explores image-space attributes to compute the gradient. Only where value of the gradient is high, additional samples are used. The final result is anti-aliasing with 8X super-sampling quality, for just 2X rendering time increase (on average).

Keywords: *Rendering, adaptive anti-aliasing, ray-tracing, SIMD, MSAA-patterns.*

1. INTRODUCTION

Raster displays use a finite number of pixels to display the scene: visible artefacts appear where pixels cannot adequately represent high-frequency data.

Rasterization and ray tracing are combating aliasing in a different manner. The two most popular ways within the rasterization context are super-sampling (SSAA) and multi-sampling anti-aliasing (MSAA). Super-sampling is brute-force approach performed by rendering the scene at a higher resolution and then down-sampling to the target resolution. Super-sampling is

expensive in terms of both performance and memory bandwidth. Today's GPUs use MSAA is an approximation to super-sampling, avoiding unnecessary shader invocations, and CSAA [1] further improves speed by decoupling coverage samples from color/z/etc samples. Hardware CSAA/MSAA modes are characterized by the pattern of the sampling grid, refer to the nice overview [2].

At the same time, both MSAA and CSAA are still brute-force methods. Performing equally for the whole render target, they lack adaptivity. The power is wasted on smooth areas, whereas some problematic pieces of the image might still lack the samples. For relatively low frequency effects an interesting alternative to HW AA is mixed resolution rendering [3] which is more adaptive. Finally, rasterization antialiasing considers quality of edges primarily. For geometry aliasing (e.g. aliasing of shadows/reflections) the existing GPU algorithms rely either on filtering or increasing overall resolution, for shadow or environment maps respectively. In contrast, for RT the true shadow/reflection rays can be traced to get the additional data at any specific frequency.

At the same time direct super-sampling methods are too expensive for real-time ray tracing. Also computing pixel coverage that is required for MSAA/CSAA would essentially mean shooting additional rays anyway, thus falling back to super-sampling.

The iterative nature of ray tracing allows for adaptive schemas, when additional rays are spawned only where it is necessary, so high-quality results are obtained without significantly increasing the storage resources and rendering time.

The major question for any adaptive scheme is the way how troublesome pixels that would need additional samples are identified. A recent scheme in [6] uses edge-detection filter that again works well for edge smoothing, but doesn't consider geometry aliasing.

For RT there is an option to find discontinuities on the per-packet basis, but smoother results are obtained, when some global (e.g. frame-wide) information is used. In the paper we focused on simple three-pass scheme. Initial pass is sparse stratified sampling, sharing as many samples as possible with adjacent pixels. Second pass is discontinuity detection. In the final pass additional samples (i.e. rays) are traced.

Researchers proposed algorithms for tracing coherent ray packets instead of single rays ([8]) using SIMD instructions. Thus a sample pattern should also be packet-friendly. The paper describes efficient grouping of the pattern rays for SIMD-aware ray tracing algorithms. We consider the 4-way SIMD (e.g. SSE) primarily.

We argue that to stay real-time one would need to consider inter-pixel pattern design. We introduce 2x2 pixels pattern. At the same time the proposed primary pattern still allows for sub-pixel accuracy.

2. RELATED WORK

Whitted was first to suggest adaptive super-sampling with recursively subdividing the pixels [9] for ray tracing. Mitchell presented effective non-uniform sampling patterns and applied contrast measure thresholds [10]. Painter and Sloan [11] presented hierarchical adaptive stochastic sampling for ray tracing that worked in progressive manner.

Cone tracing [12] is an example of the instant ray-filtering approach that overcomes the aliasing problems resulting from the point sampling approach of ray tracing. The space is probed with a finite-width cone instead of a ray. The intent is to prefilter by computing the integral of the image function within a circle on the image plane. Similar goals were pursued by different researchers through the introduction of polygonal beams [13] and finally frustum tracing with MLRTA [14]. MLRTA provides a natural measure of the geometric complexity (i.e. aliasing probability) of specific image regions. But no applications of the MLRTA for the anti-aliasing are described by the best of our knowledge, particularly for shadows/reflections, that are less advantageous for frustum tracing.

Also, the geometry complexity is not the only mechanism that contributes to unwanted high frequencies in ray-traced image: shadow edges, specular highlights, mapped textures, reflected and refracted details, etc. The only contribution that can be pre-filtered in advance is the texture aliasing (combined with ray differentials [15]). The rest require increasing the sampling rate.

We follow the previous adaptive sampling techniques in the approach of detecting problematic regions via frame-buffer *color* comparisons. This in fact, the very property that leads to the most general discontinuity detection, while also efficiently accounting for the aliasing mechanisms altogether. The only exception is texture anti-aliasing that is done locally on a surface [15], rather than in the image plane.

There are recent approaches [6] where geometry attributes (like normal) and shadow existence contribute separately to the multi-valued threshold vector. While this approach produces better quality it does increase the resource and computational pressure.

Also it can be too conservative (and expensive) for areas where some additional attributes might appear aliased thus causing additional sampling, that would be avoided if final shading and blending were performed first.

There are plenty of sampling patterns [2]. However most of them are concerned with *intra*-pixel sampling strategies. It does make sense for distribution ray tracing [4]. We consider *inter*-pixel design by sharing samples within 2 x 2 block of pixels. Since more sparse sampling may under-utilizes SIMD units, due to lowered rays coherency, we pack the pattern rays in 3 coherent groups.

3. SOLUTION

Adaptive super-sampling is a smart way of refining the rendering of the scene at those exact places where it will deliver the greatest benefit. In the paper we focused on simple three-pass scheme, Figure 2, left. We consider each pass in details below, leaving the analysis for the next section.

Initial pass is sparse pre-sampling, sharing as many samples as possible with adjacent pixels (Figure 2 right). We use FLIPTRI pattern [5] for inexpensive sampling during initial pass. FLIPTRI costs only 1.25 samples per pixel on average. It is also exhibits reasonable stratification for horizontal, vertical and diagonal strata. FLIPTRI is the most efficient filter, in terms of quality/cost for most cases [2].

We also tried scheme based on FLIPQUAD as a primary sampling pattern, refer to analysis in section 4. Both FLIPTRI and FLIPQUAD are determined for one pixel first, and the rest of the sets are then obtained by mirroring along the axis of translation. This implicitly results in interleaved sampling [16]:

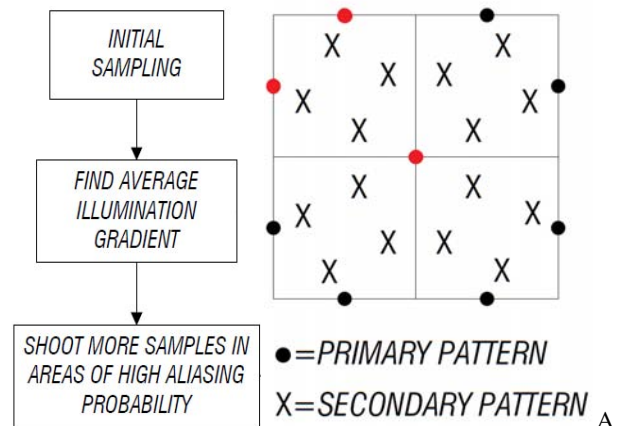


Figure 2: Simple three-pass scheme for adaptive anti-aliasing, left. An example how primary and secondary pattern rays are shared for a 2x2 pixel block. Single FLIPTRI pattern is marked with read, right.

To utilize available CPU's SIMD units completely for primary pattern, we group initial samples in SSE packets: 2 edge packets and one corner package. These three packets are efficiently shared for the 2x2 pixel block, Figure 3. Similarly are shared the samples within four (3 edge/1 corner) packets of FLIPQUAD.

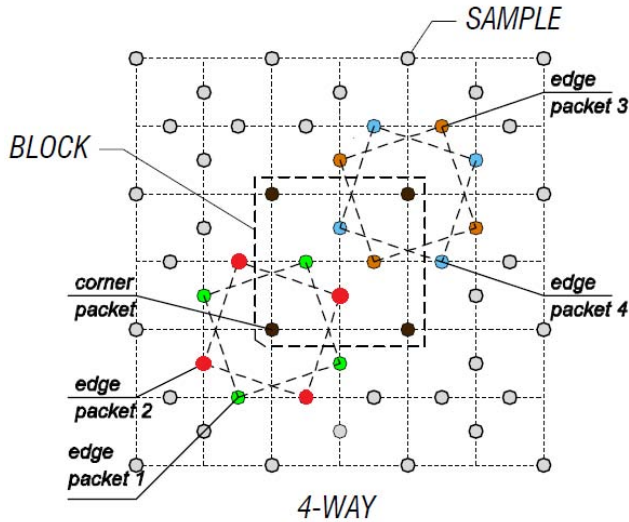


Figure 3: Packet-grouping of the rays within FLIPTRI-based primary pattern. Two edge packets are shared within 2x2 block of pixels, e.g. (edge) packets 1 with 2, or packets 3 with 4. The corner (black dots) packet is also shared, but within another 2x2 block. This irregularity breaks up symmetry somewhat, which increases the quality.

Second pass is discontinuity detection. It involves computation of gradients, performed pair-wise between samples in a primary pattern, Figure 4. The gradients can be computed either for luminance (i.e. brightness) value or separately per-color component. Finally the average gradient magnitude is computed for the frame. This value serves as a threshold in the final pass, where additional samples (i.e. rays) are generated. If super-sampling threshold value from the previous frame is used, then storing all pre-sampling results for finding average gradient can be avoided. Then, no dedicated pass is required, instead, the decision to super-sample or not can be immediately applied, once the values of initial sampling pattern are determined. This way the original three-pass scheme (Figure 2, left) can be boiled down to a single pass, while average gradient estimation (for the next frame) can be coupled with post-processing routine like tone-mapping. This approach improves cache utilization, resulting in overall performance improvements of ~5%.

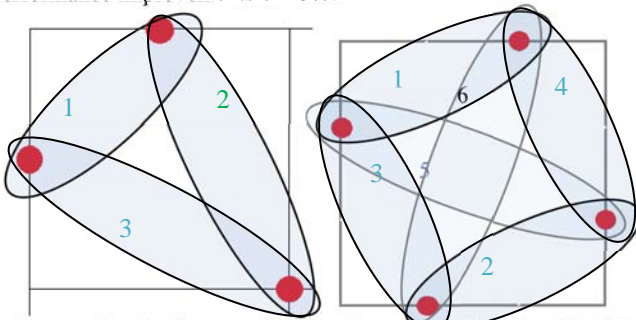


Figure 4: Gradient evaluation is performed pair-wise for samples of the primary pattern. Three gradients are computed for FLIPTRI (left), and six for FLIPQUAD (right).

For additional samples for FLIPTRI scheme we use conventional instantiation of N-rooks sampling [17], known as rotated grid super-sampling (RGSS), refer to Figure 2, right. The

final color value for pixel is computed via simple averaging with equal weights for all samples (e.g. box-filter).

4. Results and analysis

Below is a performance for the scenes with the anti-aliasing approach described in the previous section:

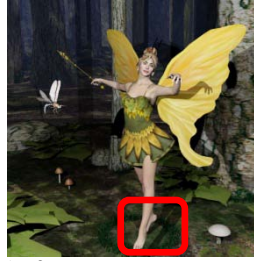

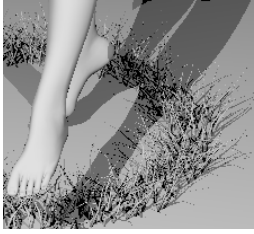
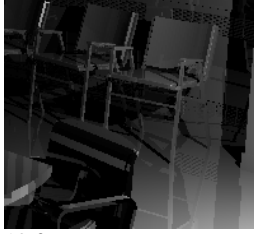
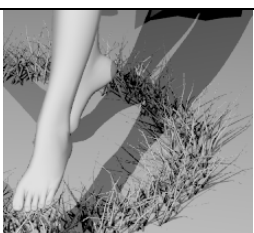
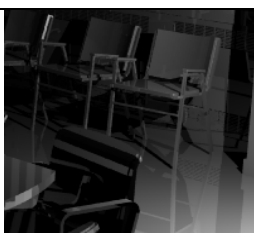
Fairy (178K triangles) 1 point light, no reflections  16 fps	Conference(282K tris) 1 point light, 1 reflection bounce  12 fps	scene lights refl.
 16 fps	 12 fps	no AA
 8 fps	 7 fps	FLIP TRI

Table 1: Performance/quality results for the proposed adaptive anti-aliasing scheme vs conventional no-AA rendering. FLIPTRI is used for primary pattern. Models are ray traced at 1024x1024 on a Intel @Core™ i7 @3.33 GHz machine with 4 Gb RAM. For the Fairy scene close-up the material colors are turned off to better demonstrate the effect.

The baseline time to frame is only 1.25X of the no-AA version (due to inexpensive primary pattern). The rest is contributed by additional sampling that costs from 20% to 30% of the frame time, depending on the scene. The final result is anti-aliasing with 8X super-sampling quality, for just 2X rendering time (on average).

The FLIPQUAD primary pattern is more expensive while produces better quality, refer to Figure 5. In compare to other sampling patterns costing 2 samples per pixel (e.g. Quincunx), it's behaviour is clearly preferable [2], [18]. It is also able to find and fix more discontinuities than FLIPTRI. Coupled with more additional samples, the FLIPQUAD might be recommended as a good higher-quality preset:

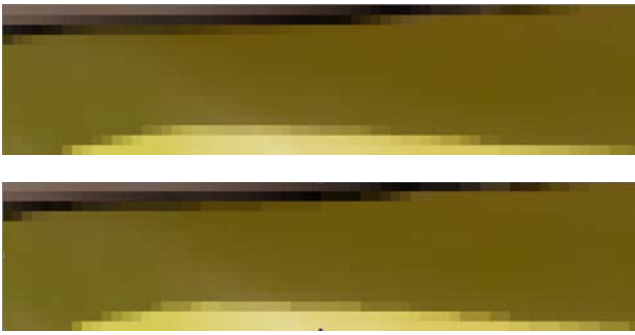


Figure 5: Quality comparison for 2 anti-aliasing presets: FLIPQUAD with 16 additional per-pixel samples (upper image) and FLIPTRI with 4 samples shared for 2x2 pixels block (lower image). The performance difference is not very high (just 2X) due to better ray coherence of the FLIPQUAD-based preset.

Note on texture anti-aliasing. For scenes with textures exhibiting large variations, discontinuity detection in the image space might generate unnecessary sampling. Since texture anti-aliasing is performed locally on a surface, it makes sense to estimate gradients (section 3) separately from textures, in spirit of [7]. However we found that using local average texture intensity for gradient estimation works fine, while avoiding many false positives, see Figure 6.



Figure 6: When the texture exhibits high variation, only its local average value is considered for gradient estimation. This helps to avoid unnecessary super-sampling. Still other sources of aliasing are detected correctly (e.g. reflection boundaries: at the right).

5. FUTURE WORK

We consider using MLRTA as a topic for future research. This would allow quickly skipping areas that don't exhibit geometry aliasing. While this is obvious for primary rays, the research is required for secondary rays.

Currently we use box-shaped reconstruction filter and equal weights for all samples. Increasing the size of the reconstruction filter from 1x1 to 2x2 (or 3x3, which is advantageously symmetric) pixels enables the sampling pattern to more accurately approximate a wider reference filter [17]. This neither increase resource consumption, nor complexity of the filtering algorithm.

6. REFERENCES

- [1] NVIDIA Corporation Rendering. "CSAA (Coverage Sampling Antialiasing)" (2007). <http://developer.nvidia.com/object/coverage-sampled-aa.html>
- [2] Hasselgren J., Akenine-Moller T., Laine S.: "A Family of Inexpensive Sampling Schemes", Computer Graphics Forum, v. 24 (2005).
- [3] Shopf J. "Mixed Resolution Rendering", GDC 2009 slides.

- [4] Boulos S., Edwards D., Lacewell J D., Kniss J., Kautz J., Shirley P., and Wald I. "Interactive Distribution Ray Tracing" Tech. rep., University of Utah, 2006
- [5] Akenine-Moller T.: "An Extremely Inexpensive Multisampling Scheme". Tech. rep., Chalmers University of Technology, 2003.
- [6] Iourcha K., Yang J. C., Pomianowski A. "A Directionally Adaptive Edge Anti-Aliasing Filter", Proceedings of High Performance Graphics, 2009.
- [7] Bongjun Jin, Insung Ihm and Byungjoon Chang. "Selective and Adaptive Super-sampling for Real-Time Ray Tracing", Proceedings of High Performance Graphics, 2009.
- [8] Wald I., Benthin C., Wagner M., and Slusallek P., "Interactive Rendering with Coherent Ray Tracing". Proceedings of Eurographics 2001.
- [9] Whitted T., "An Improved Illumination Model for Shaded Display", Comm. ACM, 23(6), 1980.
- [10] Mitchell D.P., "Generating Antialiased Images at Low Sampling Densities", Computer Graphics, 21(4), 1987.
- [11] Painter J., and Kenneth S., "Antialiased Ray Tracing by Adaptive Progressive Refinement", Computer Graphics, 23(3), 1989.
- [12] Amanatides, J. "Ray Tracing with Cones", Computer Graphics, 18(3), 1984.
- [13] Heckbert P., Pat Hanrahan P., "Beam Tracing Polygonal Objects", Computer Graphics, 18(3), 1984.
- [14] Reshetov A., Soupikov A.. and Hurley J. "Multi-level ray tracing algorithm". Proceedings of ACM SIGGRAPH (2005).
- [15] Keller A., Heidrich W. "Interleaved sampling". Proceedings of Eurographics Workshop on Rendering (2001).
- [16] Igehy H. "Tracing ray differentials". Proceedings of SIGGRAPH(1999).
- [17] Shirley P. "Discrepancy as a quality measure for sampling distributions". Proceedings of Eurographics 1991.
- [18] Laine S., Aila, T. "A Weighted Error Metric and Optimization Method for Antialiasing Patterns", Computer Graphics Forum, v.25 (1), 2006.

About the authors

Maxim Shevtsov is a Research Scientist in Nizhniy Novgorod Laboratory of Intel Corporation. He received MS degree in CS from Novosibirsk State University in 2003. His contact email is maxim.y.shevtsov@intel.com

Mikhail Letavin is a Software Engineer in Nizhniy Novgorod Laboratory of Intel Corporation. He received MS degree from Nizhniy Novgorod State Technical University in 1998. His contact email is mikhail.letavin@intel.com

Alexey Rukhlinskiy is a Graphics Software Engineer in Nizhniy Novgorod Laboratory of Intel Corporation. He received MS degree in CS from Novosibirsk State University in 2002. His contact email is alexey.v.rukhlinskiy@intel.com

Biased global illumination via Irradiance Caching and Adaptive Path Tracing on GPUs

Vladimir Frolov^(1,2), Alexander Kharlamov^(1,2) and Alexey Ignatenko⁽¹⁾

(1)The department of Computational mathematics and cybernetics, Lomonosov Moscow State University, Moscow, Russia.

{vfrolov, ignatenko}@graphics.cs.msu.ru

(2) NVIDIA, akharlamov@nvidia.com

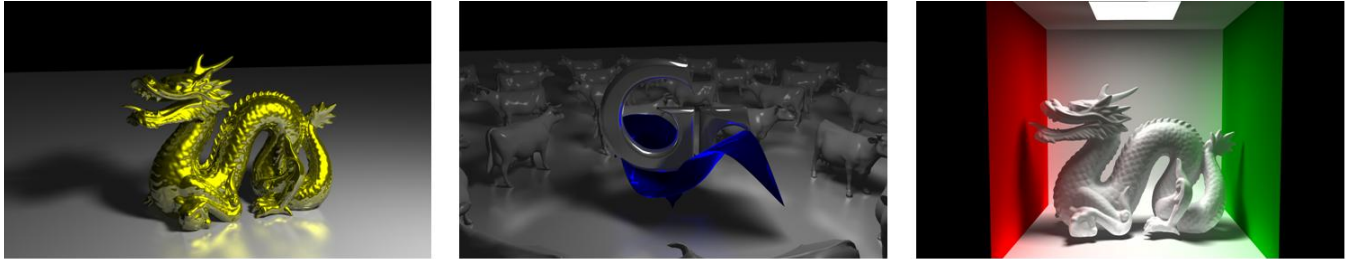


Figure 1. The presented hybrid approach uses irradiance caching to approximate smooth indirect lighting and path tracing for fuzzy effects such as soft shadows or glossy reflections. All screenshots were rendered at 1920x1200 resolution on a GTX 260 under 3 minutes.

1. ABSTRACT

This work presents an approach for biased photorealistic rendering on GPUs. The key idea is to combine irradiance caching with coherent adaptive path-tracing to maximize performance of the SIMD style execution.

Key words: *Global illumination, photorealistic rendering, GPU computing.*

2. INTRODUCTION

For the last decade Graphics Processor Units (GPUs) have made a great advance and became fully programmable processors. Since pixel shader 2.0 appeared, ray-tracing community has seen several successful implementations that used GPUs to perform ray-tracing. With the introduction of CUDA programming model this area of research experiences further growth. As more APIs are becoming available one can expect to see an increasing interest in this area. We have started our research at the point when CUDA C was the only option, however since programming model is shared between CUDA C / OpenCL and DirectX Compute APIs, we can generalize our results, and make some HW independent conclusions.

Several unbiased GPU photorealistic renderers are available now (IRay, Octane, Arion). However, we find that unbiased solutions have two significant drawbacks. Unbiased approaches usually perform computations in a brute-force manner. This means, that algorithmic complexity is higher, and in practice, unbiased approaches may use up to an order of magnitude more rays than alternative algorithms (such as irradiance cache). The second problem comes from a highly irregular nature of ray-tracing itself. Although each ray can be processed in parallel, the workload and data access pattern per each ray can be very different. This can lead to inefficient resource utilization. Although algorithmically more efficient biased approaches are more difficult to implement on GPU because of their complex nature and unbalanced work distribution. Our paper presents the research that we have performed on GPU efficiency for ray-tracing. We present a global illumination pipeline that uses irradiance cache with path tracing

to quickly compute smooth indirect illumination, and soft shadows / glossy reflections.

3. RELATED WORK

3.1 GPU ray-tracing

Purcell et al. in [1] proposed to implement ray tracing pipeline in a set of fragment programs. Uniform grid has been used as an acceleration data structure due to simplicity. Data streaming was arranged so that a ray was generated within one kernel (this was implemented using a fragment program executed over a full screen quad), second kernel would perform grid traversal. If the ray hits a voxel with triangles, it is passed to a ray-triangle intersection kernel. If the intersection is not found, it is passed back to the grid traversal kernel. To manage the state of a ray (traversal, intersection, and shading) stencil test was set up respectively. The simplicity of this approach is appealing even today. It allows easier debugging, along with a more focused performance bottleneck analysis.

Foley et al. in [2] suggest two alternative approaches ray tracing kd trees. Since GPUs don't natively support stack, the authors suggest implementing one of the following techniques:

- 1) Modify kd tree nodes to support a reference to the parent node. This reference is used whenever a ray needs to backtrack to the parent node and to process a different subtree.
- 2) Traverse a kd tree until a non empty leaf is found. However if a ray doesn't intersect any triangles within this leaf, ray's origin is modified to skip the same leaf. The kd tree traversal is restarted and the whole process is repeated until intersection is found or the ray exits the scene.

Horn et al. in [3] suggest a modification of the restart algorithm. The idea is to keep a short stack in registers and resorting to restarts in fewer cases.

For bounding volume hierarchies Thrane et al. in [4] have shown the stackless traversal for efficient GPU implementations. Each leaf stores an escape index to the corresponding node as shown in Figure 2.

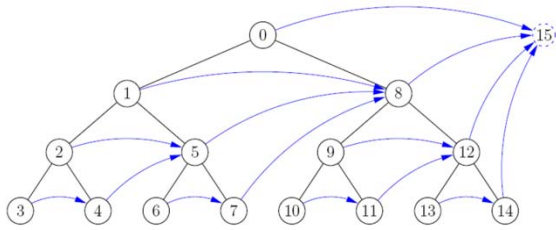


Figure 2 BVH structure for stackless traversal.

3.2 GPU Global Illumination

Wang in [5] presents an efficient approach for the global illumination using photon mapping on GPU. The key aspect of this work is to use irradiance cache with photon mapping and final gathering to quickly compute smooth indirect illumination. Direct lighting is computed using simple ray tracing and supports hard shadows from point light sources.

In [5] irradiance cache points positions were determined from the geometry discontinuities. Quad tree was used for adaptive subdivision to determine irradiance cache points positions. To evaluate the discontinuities, geometry metrics (screen space discontinuity in normals or positions) have been introduced. If the Quad tree corner point's difference exceeds certain threshold, then the subdivision of this node is required, otherwise no further subdivision is required. The similar approach was used in [6], but without final gathering. Direct illumination was computed with ray tracing and indirect - with photon mapping. This algorithm works well for caustics but produces noisy results for indirect lighting.

In [7] photon mapping was used to compute the full light equation. The drawback of this approach is strong low-frequency noise and dark edges. The low-frequency noise is a general problem of photon mapping. It can be removed with a final gathering step or filtering in object space. Dark edges appear on the borders of geometry because light is gathered only from the half of the disc but the result value divides on the area of the full disc.

Both filtering and final gathering introduce additional bias. Also we consider that final gathering has difficulties when two surfaces lie close to each other and there is a lack of photons in the scene. To eliminate the coming artifacts, the more complex and slow secondary final gathering should be used (as described in [8]). So, approaches from [5] and [6] may be a good choice for an interactive rendering but not for a photo realistic image synthesis.

McGuire and Luebke used in [9] the combination of the rasterization for direct lighting and CPU-based photon tracing with GPU-based photon splatting for indirect illumination.

4. RAY TRACING BOTTLENECK ANALYSIS

A naïve implementation of a ray tracer kernel will most likely yield poor results. To get the maximum out of any architecture a deep analysis of underlying HW is required. A good starting point with a focus on ray-tracing would be [10].

The naïve kernel that traverses spatial subdivision structure, does ray-primitive intersection and shading, shows the following signs of illness:

- 1) High register count.
- 2) Visual profiler shows 90% bottleneck in “instruction” unit.
- 3) Extreme amounts of local memory spilling
- 4) Divergent branching counter spikes.

These four issues in fact are tightly interleaved. They are causing a shifting bottleneck from instruction throughput to being memory bound.

- 1) Register count directly affects HW occupancy. Occupancy is the ratio between threads running on hardware to maximum possible threads amount. Occupancy can serve as a performance metric as well as a bottleneck indicator.
- 2) HW resources such as registers or shared memory are limited. Since all threads run in parallel, HW scheduler has to make sure there are enough resources for the launched threads.
- 3) Register count indicates the amount of registers that compiler allocates per thread.
- 4) Given an X registers per streaming multiprocessor and Y registers allocated by the compiler, the total amount of threads that can coexist on the streaming multiprocessor is X / Y . In particular NVIDIA Tesla 10 architecture has $16 * 1024$ 32bit registers per Streaming Multiprocessor. Thus a register count of 32 will leave room for no more than 512 threads.

With high register count exploding it is expected to observe poor HW utilization but this is not the final problem. GPUs rely on large scale threads parallelism to cover memory access latency. Poor occupancy can turn around and become a memory bottleneck.

Compiler will try and lower the register count by pushing and popping data into local memory. However local memory has the same latency as global memory. Analyzing CUDA PTX code one would local store and local load instructions happening repeatedly within a loop. This increases bandwidth pressure, and in addition to poor latency hiding can make application memory bound.

And finally divergent branching spikes are a sign of... divergent branches. When a block of threads executes it can diverge in two ways:

- 1) Different warps follow different code paths. This is perfectly fine, because it means no additional overhead except for condition evaluation.
- 2) Different threads within a warp follow different code paths. This for example can happen as soon as a single ray from a warp finds a non empty leaf and starts ray-triangle intersection. In this case HW will generate additional warp that will execute the code path. Partly threads will be masked out. The warps will be merged together as soon as the code paths merge back together.

4.1 Proposed ray tracing pipeline

Removing the bottlenecks can be tricky so we decided to implement a simple divide and conquer strategy. We split the ray tracing pipeline into the following stages (as shown in Figure 3):

- 1) **Ray generation kernel.** This can be a kernel that generates eye rays or secondary rays. Rays are packed into a linear list; direction and origin are stored in a structure of arrays fashion.
- 2) **Tree traversal kernel.** At this stage all rays are traced through a kd tree using stack in local memory. The idea is that the tree depth is usually defined beforehand during tree construction. This allows us to conservatively estimate stack size to be no more than tree depth. That stack stores an index

and t_{far} for kd tree. The output of this kernel is a list of non-empty leaf indices. This list serves as input to the next stage. To avoid constant switches between kernels and extra overhead, we traverse the tree until a number of suitable leafs is found. They are all written to a pre-allocated buffer. The stack in local memory is lost after traversal kernel completes, so in case we can't afford to allocate enough memory to keep enough leaf nodes in it, we can resume tree traversal using "restart" logic.

- 3) Ray-primitive intersection kernel accepts a list of rays as input along with a list of leaf candidates per each ray. If intersection was not found within leaf boundaries, then it sends the modified ray back to tree traversal stage. If the intersection point was found, then the ray is passed along.
- 4) Shadow kernel generates shadow rays and checks light visibility.
- 5) At shading stage we compute direct illumination with shadows.
- 6) The goal of the material kernel is to generate secondary rays. Typically it would generate reflection and refraction rays. These rays are sent to traversal stage.
- 7) Store result stage performs final light equation integration.

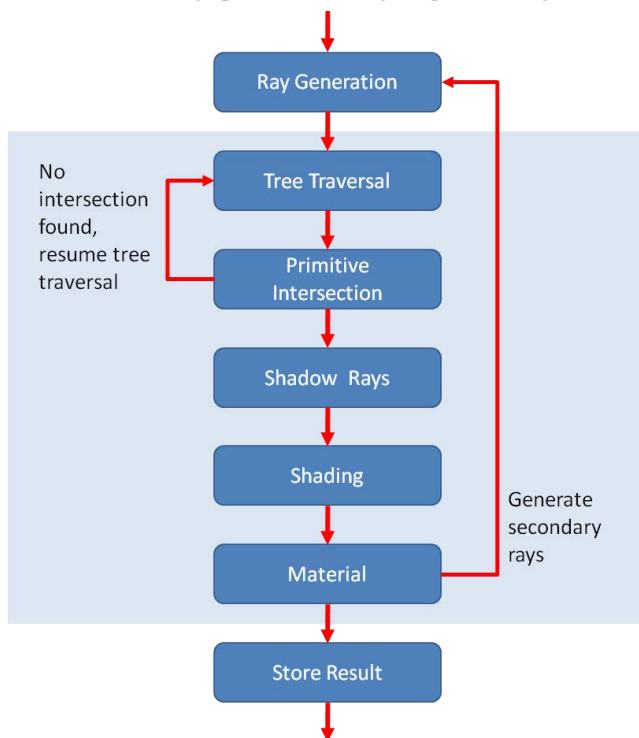


Figure 3 Ray tracing pipeline

This separation provides the following advantages:

- 1) Complete register usage comprehension. While kd tree traversal can fit into 16 registers and achieve perfect occupancy, ray-primitive intersection consumes 32 registers per thread. On Tesla 10 architecture we can't do better than with $\frac{1}{2}$ occupancy. Shading turned out to be the most register hungry kernel due to the shading model peculiarities.
- 2) By reducing register pressure we have removed a significant portion of local stores and loads.

- 3) We have removed a significant portion of divergent threads. In fact since traversal / intersection and shading are all different kernels, the only divergence on a warp level is due to different time spent within the kernel loops.

There is still one remaining problem: varying workload per thread. Now, since we have different kernels, they serialize and intersection will not start until traversal is fully complete. This may well turn into waiting for a single thread that has the longest route through the tree. To reduce that further we implement a technique called persistent threads: each thread processes several rays instead of one. We divide the screen into blocks as shown in Figure 4 yet we launch a number of thread blocks that GPU can process in parallel. In this case each thread block has a fixed number of blocks to process. This is similar to ray pool described by Aila in [10], however the number of rays per each thread is fixed and we avoid using atomics.



Figure 4. Screen divided into blocks. Blocks of the same color are processed by the same thread block

Finally, after achieving a good performance of 50 Ms rays/sec on average we have combined the existing traversal – intersection kernels back, leaving just the shading outside.

We can now pass leaf nodes between two stages through a short list in shared memory. Each thread has a few private leaf IDs stored in shared memory (it's indexable and essentially free to use instead of registers).

Combining traversal and intersection into an uber-kernel with simple persistent threads management provides us with additional benefit of lesser kernel launch overhead, easier thread management. The uber-kernel is in general slightly (5-10%) faster than its separated analogue, and doesn't have any memory overhead. However both solutions are just variations of the same software load-balancing idea.

5. SUGGESTED APPROACH

For the fast global illumination solution a combination of distributed ray tracing and irradiance caching is commonly used. We suggest a similar idea: for fast and smooth indirect lighting we use irradiance caching technique as described in [11] and we use path tracing for other effects, such as soft shadows, glossy reflections and refractions, depth of field and motion blur [8]. The main motivation behind this step is to use simple iterative algorithm and avoid complex recursive nature of distributed ray

tracing. On the other hand due to highly divergent nature of path tracing we avoid using it for full light equation evaluation and consider it only for special effects.

However, this also increases the problem with the work distribution. The black circles in Figure 5 show simple regions that require several iterations for light integration to converge. The red squares show complex regions with soft shadows and reflections. While 10-20 iterations are enough for most pixels to converge to light equation solution, some areas of pixels require 100-1000 iterations to eliminate noise. This problem is solved on the CPU by processing each pixel until sufficient quality is achieved. On the GPU, however, this presents a challenge due to of the unpredictable workload.

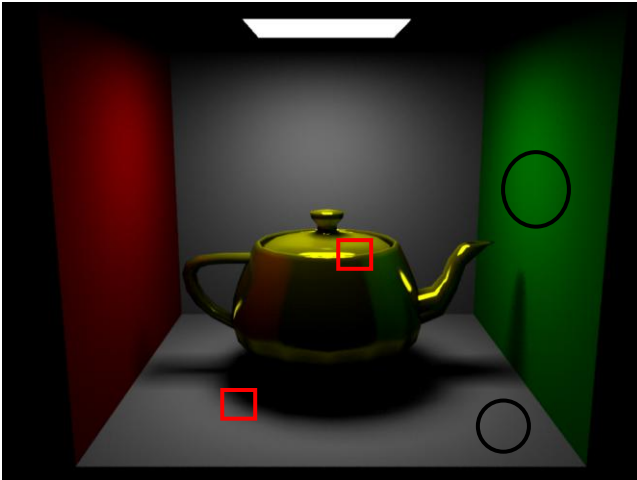


Figure 5. Teapot inside the Cornell box, direct lighting only.

5.1 Adaptive path tracing

For adaptive path tracing we split our screen into tiles as shown in Figure 6. Within the tiles we use Z-curve indexing for all ‘per-ray’ data (ray position, direction, and intersection info *etc*). This removes large address gaps for all pixels within a tile and enables an important bandwidth-saving optimization on NVIDIA hardware (coalesced memory reads and writes).

We define **TMAX** to be a number of tiles that we can process in parallel. **TMAX** depends on the amount of memory that we are prepared to allocate.

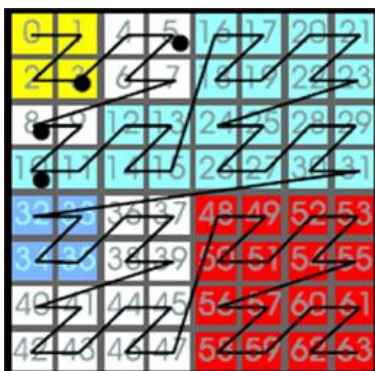


Figure 6. Z-Curve used to indexing pixels inside the tile.

We mark all tiles as active in the beginning of the rendering and add all tiles to the “active-tiles” list. In the example pseudo code below we assume tile size is 16x16:

```

var rays_per_pixel : integer;

procedure Adaptive_Path_Tracing is
  active_list: list of Tile;
  active_array: array (0..TMAX-1) of Tile;
  sz,i: inreger range 0..TMAX;
  tile : Tile;
begin

  subdivide screen to tiles;
  add all tiles to the active_list;
  rays_per_pixel := 1;

  while not active_list.empty():

    sz := min(active_list.size(), TMAX);
    active_array[0..sz] := active_list[0..sz];

    Process_Tiles_On_GPU(active_array, \
                          sz, rays_per_pixel);

    for i in 0..TMAX-1:
      tile := active_array[i];
      if not tile.finished():
        active_list.push_back(tile);
    end for

    if active_list.size() < TMAX * 0.5:
      rays_per_pixel *= 2;

  end while;

```

end Adaptive_Path_Tracing;

During the rendering process, some tiles finish earlier than the other. They are discarded from **active_array** and from **active_list** and replaced by new tiles from **active_list** if the last is not empty. When the number of active tiles is less than **TMAX/2**, we double the number of rays per pixel.

```

procedure Process_Tiles_On_GPU (
  active_array array (0..TMAX-1) of Tile,
  sz : Integer,
  rays_per_pixel : Integer
) is
  tile_size : Integer;
  rays_num : Integer range 0..TMAX-1;
begin

  tile_size := 16*16;
  rays_num := tile_size*rays_per_pixel*sz;
  assert (rays_num <= TMAX);

```

```

on the GPU:
  generate initial rays from the eye \
    according to the rays_per_pixel;

```

```

trace exactly rays_num rays (paths in fact);
sample result according to the rays_per_pixel;

```

```

end Process Tiles On GPU;

```

For large resolutions, like 1920x1200 our approach allows a good balance between memory consumption, performance and GPU workload.

Each tile is represented by a Tile structure. This structure is passed back and forth between CPU and GPU after each iteration.

```

type Tile is record
  index      : integer;
  max_diff   : float;
  counter    : integer;
end record;

```

The ‘index’ field is an offset to a group of 256 rays in a GPU memory. It is used when fetching rays and sampling the resulting color.

To evaluate when a tile has converged we use the following approach: each ray accumulates partial sum of lighting integral into sum_{odd} and sum_{even} for all odd and all even passes of the path tracing (normalized e.g. divided by total ray count). After each iteration we compute max_diff value – it represents the maximum difference (among all rays in a tile) between these partial sums as shown in pseudo code below:

```

for i from 0 to 255 do:
  diff(i) := ||sumodd - sumeven||c
max_diff := max of all diff(i);

```

Since we are using quasi Monte Carlo integration, we expect that integral should converge at some point. Though, there is no good estimate for the number of iterations, however, sum_{odd} and sum_{even} should converge to the same value sum . This leads us to the conclusion that as soon as $\| \text{sum}_{\text{odd}} - \text{sum}_{\text{even}} \|_c < \varepsilon$, where ε is a certain threshold that represents error, than we can stop our integration process for this pixel. When the $\text{max_diff} < \delta$ we can stop integration process for all pixels in the tile and discard that tile from the `active_list`.

Finally `counter` represents the number of passes that have been completed already.

5.2 Irradiance cache

Our irradiance cache implementation is very similar to the Wang's implementation in [5]. For each pixel we compute a surface position and normal. We do that on the GPU. Next, we construct a quad tree in screen space as in [5]. We used an initial size of 32 pixels both in horizontal and vertical directions. Each 32x32 quad is subdivided with a quad tree and geometry discontinuity is computed between quad tree nodes. When the discontinuity is less than a threshold, we do not perform further quad tree subdivision. The chosen pixels correspond to the irradiance cache points in object space. At each point we generate a set of rays to sample hemisphere and compute indirect illumination. To have more coherent groups of rays we subdivide hemisphere into sectors and generate $32 \times k$ rays for each sector where $k \geq 1$. We do that on CPU in tangent space. On GPU we transform directions from

tangent to object space to get correct hemisphere sampling. Next, we construct a multiple-reference octree as described in [11]. We implement interpolation algorithm, quality metric and stackless octree look-up as described in [12]. It seems that stackless approach should be efficient on GPUs. However we find that multiple-reference octree is not the best solution.

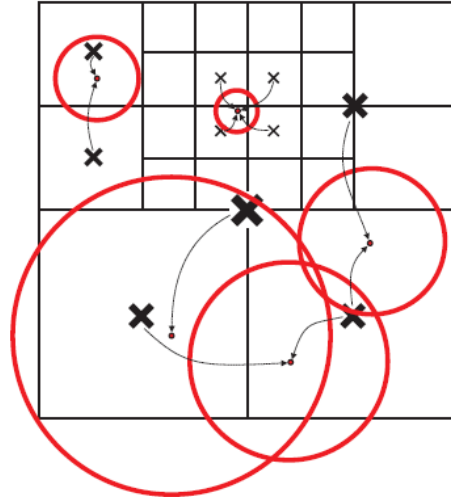


Figure 7. Multiple reference octree as described in [11].

The key advantage of the multiple reference octree is a stackless ‘root-to-leaf’ look-up algorithm. To find all points in the given sphere we can traverse tree from the root to a leaf and there is no need in stack or recursion. But the price for such simplicity is multiple references: each point can be referenced from multiple octree nodes. During octree construction or point insertion, we need to add each irradiance cache point P_i to all octree nodes that intersect with sphere centered at P_i , with radius equal to the search radius. The number of references in this approach can be a 5x-10x times larger than the number of points. On GPU this leads to dependent texture fetches and cache trashing. In our implementation octree look-up costs as much as a ray-tracing part. We suppose that kd-tree Wang’s approach from [5] will be more efficient than the multiple-reference octree from [11], this is one of the future research strategies.

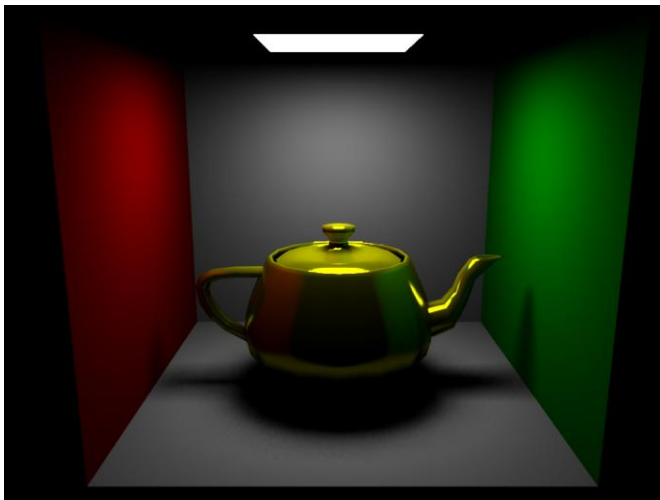


Figure 8. V-Ray; Core 2 Quad, 6600; 62 sec in 1024x768

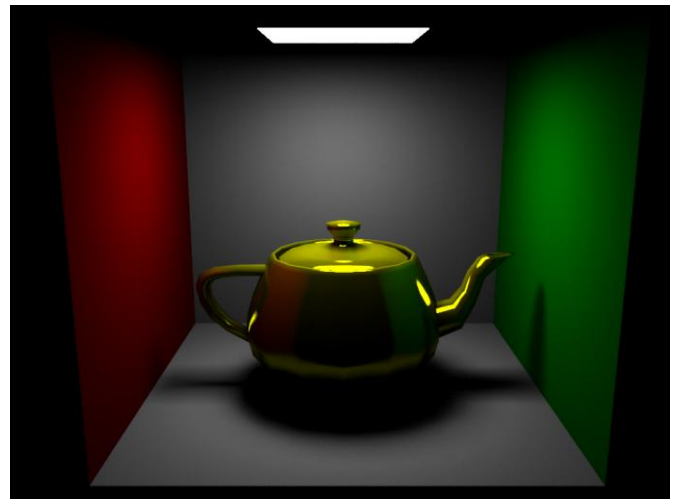


Figure 10. Our Implementation; GTX260; 15 sec in 1024x768



Figure 9. V-Ray; Core 2 Quad; 6600; 153 sec in 1024x768



Figure 11. Our implementation; GTX260; 31 sec in 1024x768

5.3 Complete solution

We have experimented with separate kernel architecture. Our motivation not to use uber-kernels in this case was:

- 1) Uber-kernels are bound by their most heavy part. For example, if we have a complex shading code, it can affect on the ray tracing performance and downgrade it;
- 2) It is possible that several pixels require thousands of rays and all these rays for each pixel will be traced in series. We suggest a solution to trace them in parallel with different threads;
- 3) Last but not least, for the complex code, like ray-tracing with different shading techniques and materials, separate kernel architecture is much more convenient than uber-kernel, especially for profiling and debugging reasons.

At each bounce of path tracing we compute direct illumination by tracing shadow rays towards each light and look-up indirect illumination from the irradiance cache. To reduce octree look-up cost we trace several shadow rays for each shadow sample. This solution allows us to do less look-ups on the regions with complex soft shadows.

6. RESULTS

Our ray tracing implementation runs with 30-50M rays per second on the 'Conference Room' scene and GTX260 GPU. It corresponds to the other works related to GPU ray tracing: [10], [13]. We use a SAH kd tree to accelerate ray-triangle intersection. We compared our renderer with V-Ray on the simple scene both with direct and indirect illumination. Our implementation shows good performance scaling for large geometry and higher resolutions (Fig 8-11).

V-Ray is a commercial renderer and we don't know exactly how it works, so it is hard to make a precise per-pixel image comparison. On the middle-level hardware our implementation performs up to 4x times faster than V-Ray. As we did not pursue the aim to make a per-pixel comparison, Figures 9 and 11 are slightly different but our original image contains less visible noise.

For the Dragon model irradiance cache construction takes 4 seconds. For the simple scenes, like a teapot in Cornell Box, it takes less than 0.5 sec.

All our Demos, videos, comparisons and screenshots can be found at <http://ray-tracing.com> (English) and <http://ray-tracing.ru> (Russian).

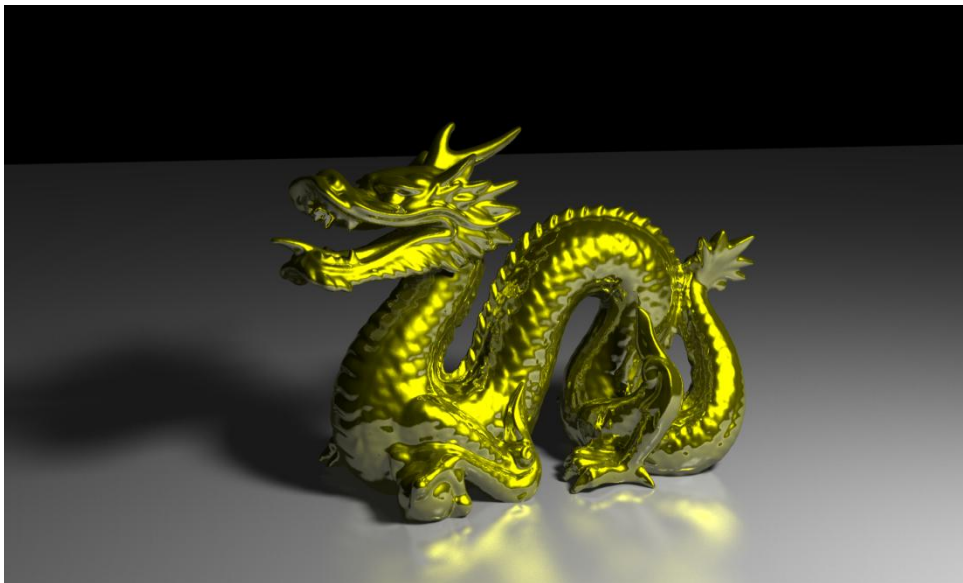


Figure 12. 1920x1200. GTX260; 144 sec.

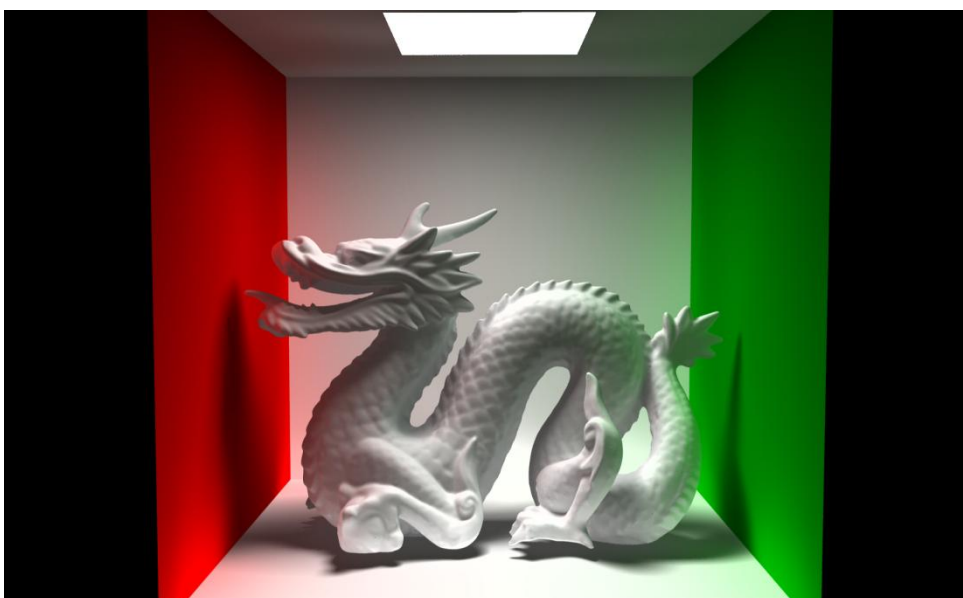


Figure 13. 1920x1200. GTX260; 181 sec

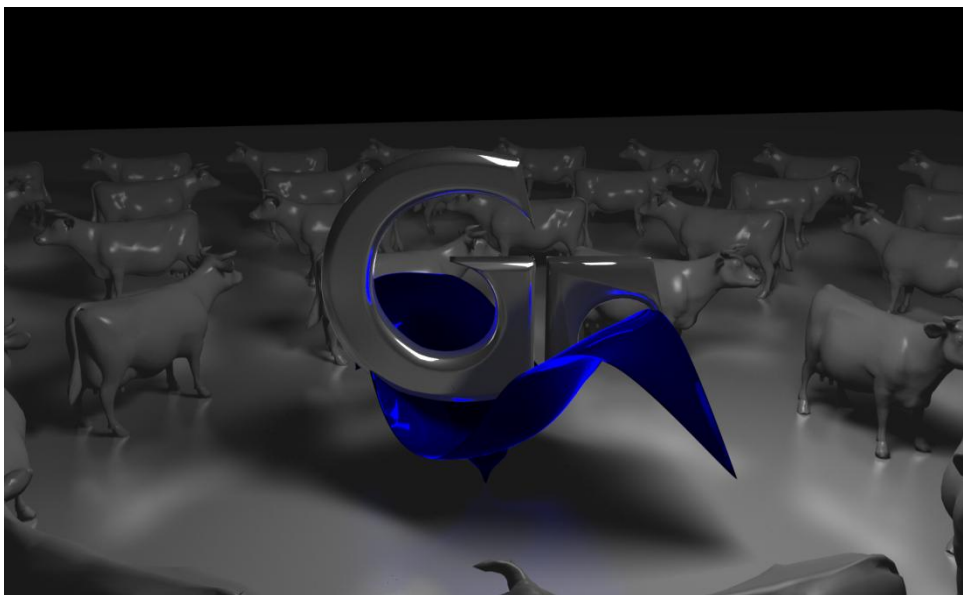


Figure 14. 1920x1200. GTX260; 159 sec

7. LITERATURE

- [1] Purcell, T. J., Buck, I., Mark, W. R., and Hanrahan, P. 2005. Ray tracing on programmable graphics hardware. In *ACM SIGGRAPH 2005 Courses* (Los Angeles, California, July 31 - August 04, 2005). J. Fujii, Ed. SIGGRAPH '05. ACM, New York, NY, 268. DOI=<http://doi.acm.org/10.1145/1198555.1198798>
- [2] Foley, T. and Sugerma, J. 2005. KD-tree acceleration structures for a GPU raytracer. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware* (Los Angeles, California, July 30 - 31, 2005). HWWS '05. ACM, New York, NY, 15-22. DOI=<http://doi.acm.org/10.1145/1071866.1071869>
- [3] Horn, D. R., Sugerma, J., Houston, M., and Hanrahan, P. 2007. Interactive k-d tree GPU raytracing. In *Proceedings of the 2007 Symposium on interactive 3D Graphics and Games* (Seattle, Washington, April 30 - May 02, 2007). I3D '07. ACM, New York, NY, 167-174. DOI=<http://doi.acm.org/10.1145/1230100.1230129>
- [4] THRANE N., SIMONSEN L. O.: A Comparison of Acceleration Structures for GPU Assisted Ray Tracing. Master's thesis, University of Aarhus, 2005. 2, 3
- [5] Wang R., Zhou K., Pan, M., and Bao, H. 2009. *An efficient GPU-based approach for interactive global illumination*. *ACM Trans. Graph.* 28, 3 (Jul. 2009), 1-8.
- [6] Fabianovski B., Dingliana J. *Interactive Global Photon Mapping*. In Proceedings of the EUROGRAPHICS conference, 2009. p. 1151-1159.
- [7] Frolov V., Ignatenko A. *Interactive GPU Ray Tracing and Photon Mapping*. In: GraphiCon'2009.; 2009. p. 255-262. (In Russian).
- [8] Jensen, H. W., Suykens F., Christensen Per H., Kato T. A *Practical Guide to Global Illumination using Photon Mapping*. SIGGRAPH 2002 Course Note #43. ACM, July 2002. (San Antonio, USA, July 21-26).
- [9] McGuire, M. and Luebke, D. 2009. Hardware-accelerated global illumination by image space photon mapping. In *Proceedings of the Conference on High Performance Graphics 2009* (New Orleans, Louisiana, August 01 - 03, 2009). S. N. Spencer, D. McAllister, M. Pharr, and I. Wald, Eds. HPG '09. ACM, New York, NY, 77-89. DOI=<http://doi.acm.org/10.1145/1572769.1572783>
- [10] 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.
- [11] Křivánek, J., Gautron, 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. DOI=<http://doi.acm.org/10.1145/1401132.1401213>
- [12] Pharr, M. and Humphreys, G. 2004 *Physically Based Rendering: from Theory to Implementation*. Morgan Kaufmann Publishers Inc.
- [13] Garanzha K., Loop C. *Fast Ray Sorting and Breadth-First Packet Traversal for GPU Ray Tracing*. In Proceedings of the EUROGRAPHICS conference, vol. 29 (2010), Number 2.

8. ABOUT THE AUTHORS

Vladimir Frolov graduated with MS degree from Computational Mathematics and Cybernetics department of Lomonosov Moscow State University. Vladimir works at NVIDIA as an intern. His area of research is photorealistic rendering, fluid simulation and gpu programming. His contact email is vfrolov@graphics.cs.msu.ru.

Alexander Kharlamov is a PhD student at Computational Mathematics and Cybernetics department of Lomonosov Moscow State University. Alexander works at NVIDIA as a developer technology engineer. His area of research is photorealistic rendering, physical simulation and gpu programming. His contact email is alharlamov@nvidia.com.

Alexey Ignatenko is a PhD researcher at Computational and Cybernetics department of Moscow State University. His contact e-mail is ignatenko@graphics.cs.msu.ru.

Modeling BRDF by a Probability Distribution

Aydın Öztürk¹, Murat Kurt² and Ahmet Bilgili²

¹Department of Computer Engineering, Yasar University, Izmir, Turkey
aydin.ozturk@yasar.edu.tr

²International Computer Institute, Ege University, Izmir, Turkey
murat.kurt@ege.edu.tr, ahmetbilgili@gmail.com

Abstract

In this work we investigate reciprocity and energy conservation properties of a BRDF model which is based on a probability distribution. Representational ability of underlined model is demonstrated using a measured BRDF data set. We show that our model satisfies both reciprocity and energy conservation properties of BRDF.

Keywords: BRDF model, Rendering, Copula distribution.

1. INTRODUCTION

A complete representation of surface reflection behavior through a mathematical model can be achieved by considering a number of variables including angle of light, wavelength, polarization and position etc. both for incoming and for outgoing lights. A common approach is to use a simplified model by considering the directional and spectral properties of underlying reflection only. A function of incident and reflected angles is called as the Bidirectional Reflectance Distribution Function (BRDF) defined as

$$\rho(\vec{\omega}_i, \vec{\omega}_o) = \frac{dL_o(\vec{\omega}_o)}{L_i(\vec{\omega}_i) \cos \theta_i d\vec{\omega}_i}, \quad (1)$$

where L_i and L_o are the incident and reflected radiance, respectively, $(\vec{\omega}_i, \vec{\omega}_o) = \{(\theta_i, \phi_i), (\theta_o, \phi_o)\}$ are the corresponding incoming and outgoing vectors, and $d\vec{\omega}_i$ is the differential solid angle in the ω_i direction.

A good BRDF model should obey both reciprocity and energy conserving principles. Generally, it is difficult to build a BRDF model that satisfies these two principles. However, some of the BRDFs have managed to produce visually acceptable images even though they violate energy conservation or reciprocity or both [1].

An interesting approach for modeling the surface reflectance would be to treat the BRDF in the context of probability theory. After a convenient normalization, the BRDF can be viewed as a four-dimensional (4D) probability density function of elevation and azimuth angles of incoming and outgoing vectors. A major problem of this approach is that it is difficult to construct a multivariate probability distribution that provides an adequate approximation for a given BRDF. The most well known multivariate distributions have the same marginal distributions. For example, the marginal distributions of multivariate normal distribution are all univariate normal. Generally, BRDF samples do not exhibit such a property. For example, the elevation angles have extremely skewed distributions but the azimuth angles have U-shaped symmetric distributions for most isotropic materials [2]. A straightforward solution to the problem of modeling the BRDF through probability distributions could be transforming the underlying variables into four independent variables and expressing the BRDF as a product of univariate probability density functions (pdfs) of these new random variables. An algorithm has been proposed by Lawrence et al. [3] who have factorized the 4D BRDF in the directions of incident and outgoing directions and have expressed the BRDF as a conditional distribution

for a given outgoing direction vector. Thus, for a fixed outgoing direction, the problem is reduced to a two-dimensional (2D) case. Furthermore, the 2D pdf also is factored to obtain a product of two univariate pdfs. However, this method requires storing a large data set for the factorization procedures. In a recent work, Öztürk et al. [2] proposed to model BRDF by employing copula distributions. Using Archimedean family of distributions for modeling BRDFs, it is shown empirically that the BRDF can be represented successfully by the underlying distribution.

In this work we explain the rationale behind representing the BRDF by copula distributions. We also show that this model satisfies both reciprocity and energy conservation principles.

2. PREVIOUS WORK

Various models have been proposed for approximating the BRDF. These models can be classified in two main groups: analytical models and data-driven models.

The Phong model [4] is one of the oldest and perhaps the most well known analytical BRDF model. Later, using the halfway vector, the Blinn-Phong model [5] has been introduced as an improved version of the Phong model. Ward [6] introduced a BRDF model for describing the isotropic and anisotropic materials. This model is based on Gaussian distribution. Duer [7] proposed a variation of the Ward model. This model has different normalization factor from the Ward model and this improves the fitting results [8]. The BRDF model, developed by Lafortune et al. [9], can be considered as an improved and generalized Phong model. It can represent non-Lambertian diffuse reflections, retro reflections and Fresnel effects. Ashikhmin and Shirley [10] introduced a model for describing anisotropic materials with an improved Phong model. Edwards et al. [1] has modeled the BRDF with a probability distribution. Other empirical models include the models by Lewis [11] and by Westlund and Meyer [12].

While aforementioned analytical BRDF models are phenomenological, there exist physically-based analytical BRDF models. These physically-based analytical BRDF models [13], [14], [15], [16], [17] are based on microfacet theory and can represent effects such as Fresnel reflection and rough microgeometry. Models that can represent anisotropic materials have been developed by Kajiya [18], Poulin and Fournier [19], Ward [6], Lafortune et al. [9], Ashikhmin and Shirley [10], Duer [7] and Edwards et al. [1].

Other class of analytical BRDF models consists of linear combination of some set of basis functions. For describing the BRDF with linear models, Spherical Harmonics were used by Westin et al. [20]. Similarly, Zernike polynomials were used by Koenderink et al. [21] to model the BRDF. Linear models based on Wavelets [22], [23] and PCA [24] have also been used for modeling the BRDF. Öztürk et al. [25] presented an approach based on principal component transformations of some explanatory variables for approximating both isotropic and anisotropic reflectance for diffuse and glossy surfaces. These models require a large number of coefficients to represent the BRDF. Therefore they are inefficient in terms of stor-

age space and computation complexity.

Data-driven models generally depend on large data sets. Matusik et al. [24] acquired dense measurements of the BRDF and used these measurements directly in rendering process. Since this representation requires large storage space, accurate and compact representations of measured BRDFs have been investigated [26], [27], [28], [3], [29], [30]. In these models, BRDF data are decomposed using various factorization methods.

3. COPULA DISTRIBUTIONS

Copula is a multivariate cumulative distribution function of the uniform random variables on the interval $[0, 1]$. They provide a simple and general structure for modeling multivariate distributions through univariate marginal distributions. More specifically if $F_1(x_1)$ and $F_2(x_2)$ are the continuous marginal cumulative distributions then the joint cumulative distribution function of X_1 and X_2 can be constructed in the following form

$$H(x_1, x_2) = C\{F_1(x_1), F_2(x_2)\} = C(u_1, u_2), \quad (2)$$

where C is the copula function, $u_1 = F_1(x_1)$ and $u_2 = F_2(x_2)$ [31]. It is well known that $F_1(x_1)$ and $F_2(x_2)$ are uniformly distributed on the interval $[0, 1]$ and if these cumulative distributions are continuous then the corresponding copula function is unique [31]. Based on this definition, it is clear that the marginal distributions can be determined independently after an appropriate copula model is chosen to represent dependency between the marginal distributions.

An important family of copulas, called Archimedean copulas, has a simple form with some desirable properties. These copula distributions mostly have closed forms and simple solutions. One particularly simple form of a p -dimensional Archimedean copula is

$$C(u_1, u_2, \dots, u_p) = \varphi^{-1}\{\varphi(u_1) + \varphi(u_2) + \dots + \varphi(u_p)\}, \quad (3)$$

where the function φ is known as *generator* and φ^{-1} is the inverse generating function.

Any function satisfying the following properties

- $\varphi(1) = 0$
- $\lim_{t \rightarrow 0} \varphi(t) = \infty$
- $\varphi'(t) < 0$

is defined as a generator. For example, the Frank distribution is obtained from the generating function

$$\varphi(t) = \ln \left\{ \frac{\exp(\alpha t) - 1}{\exp(\alpha) - 1} \right\}, \alpha \neq 0, \quad (4)$$

where α is the parameter of the distribution. This generator has the inverse form as

$$\varphi^{-1}(s) = \frac{1}{\alpha} \ln \{1 + \exp(s)(\exp(\alpha) - 1)\}, \quad (5)$$

and the derivation of corresponding copula is straightforward:

$$\begin{aligned} H(x_1, x_2, \dots, x_p) &= C(u_1, u_2, \dots, u_p) \\ &= \varphi^{-1} \left\{ \ln \left\{ \frac{\exp(\alpha u_1) - 1}{\exp(\alpha) - 1} \right\} + \dots + \ln \left\{ \frac{\exp(\alpha u_p) - 1}{\exp(\alpha) - 1} \right\} \right\} \\ &= \frac{1}{\alpha} \ln \left\{ 1 + \frac{(\exp(\alpha u_1) - 1) \cdots (\exp(\alpha u_p) - 1)}{\exp(\alpha) - 1} \right\}. \end{aligned} \quad (6)$$

Finally the joint pdf of X_1, X_2, \dots, X_p is obtained by taking the p th partial derivative of these variables giving

$$f(x_1, x_2, \dots, x_p) = \frac{\partial^p}{\partial x_1 \partial x_2 \cdots \partial x_p} C(u_1, u_2, \dots, u_p). \quad (7)$$

Thus, the pdfs of 2D and three-dimensional (3D) Frank distribution can be found as

$$\begin{aligned} f(x_1, x_2) &= \frac{\alpha g_1 (1 + g_{u_1 + u_2})}{(g_1 + g_{u_1} g_{u_2})^2} \prod_{i=1}^2 f_i(x_i), \\ f(x_1, x_2, x_3) &= \frac{\alpha^2 g_1^2 (1 + g_{u_1 + u_2 + u_3})(g_1^2 - g_{u_1} g_{u_2} g_{u_3})}{(g_1^2 + g_{u_1} g_{u_2} g_{u_3})^3} \prod_{i=1}^3 f_i(x_i), \end{aligned} \quad (8)$$

where $g_t = \exp(\alpha t) - 1$ and f_1, f_2 and f_3 are the marginal pdfs. It is seen from the equations above, the dependency between the variables are defined through the Frank copula with one unknown parameter α .

4. MODELING THE BRDF BY PROBABILITY DISTRIBUTIONS

Suppose for simplicity that the measured BRDF data is obtained by sampling, uniformly along each azimuth and elevation angles of incident and outgoing directions. Let $(\theta_i^r, \phi_i^s, \theta_o^t, \phi_o^u)$ denote a sample point at which the corresponding measured BRDFs (\tilde{b}_{rstu}) are obtained. Up to a constant, we can view that the measured BRDF values are the densities of a continuous joint probability function of $\theta_i, \phi_i, \theta_o$ and ϕ_o . In other words we assume that the measured BRDFs are proportional to the normalized frequencies of corresponding classes (bins). The normalizing coefficient can be determined in such a way that the volume of corresponding joint pdf is equal to 1. When the measured BRDF data is obtained by a uniform spacing with length 1 degree in the intervals $0 \leq \theta_i, \theta_o < 90^\circ$ and $0 \leq \phi_i, \phi_o < 180^\circ$ then the normalized BRDF values can be obtained as

$$b_{rstu} = \frac{\tilde{b}_{rstu}}{b_{\dots}}, \quad (9)$$

where $r, t = 0, 1, \dots, 89$; $s, u = 0, 1, \dots, 179$ and

$$b_{\dots} = \sum_{r=0}^{89} \sum_{s=0}^{179} \sum_{t=0}^{89} \sum_{u=0}^{179} \tilde{b}_{rstu}. \quad (10)$$

Based on this definition, the empirical marginal distributions can be determined easily. For example, the empirical marginal pdf for θ_i is obtained by summing over the other three variables as

$$f_{\theta_i}^r = b_{r\dots} = \sum_{s=0}^{179} \sum_{t=0}^{89} \sum_{u=0}^{179} b_{rstu}, \quad (11)$$

where $r = 0, 1, \dots, 89$. Similarly, the empirical joint pdf of θ_i and ϕ_i is determined as

$$f_{\theta_i, \phi_i}^{r,s} = b_{rs\dots}, \quad (12)$$

where $r = 0, 1, \dots, 89$ and $s = 0, 1, \dots, 179$. Marginal pdfs of $\theta_i, \phi_i, \theta_o$ and ϕ_o are shown in Figure 1 for an isotropic material. It is seen that the marginal pdfs of θ_i and θ_o both are skewed to the right and they are almost the same. This similarity property also is observed for the U-shaped marginal pdfs of ϕ_i and ϕ_o . The joint pdf of θ_i and ϕ_i is illustrated in Figure 2 for the same material.

Modeling BRDFs by probability distributions using the standard coordinate system whose coordinates are defined by $\theta_i, \phi_i, \theta_o$ and

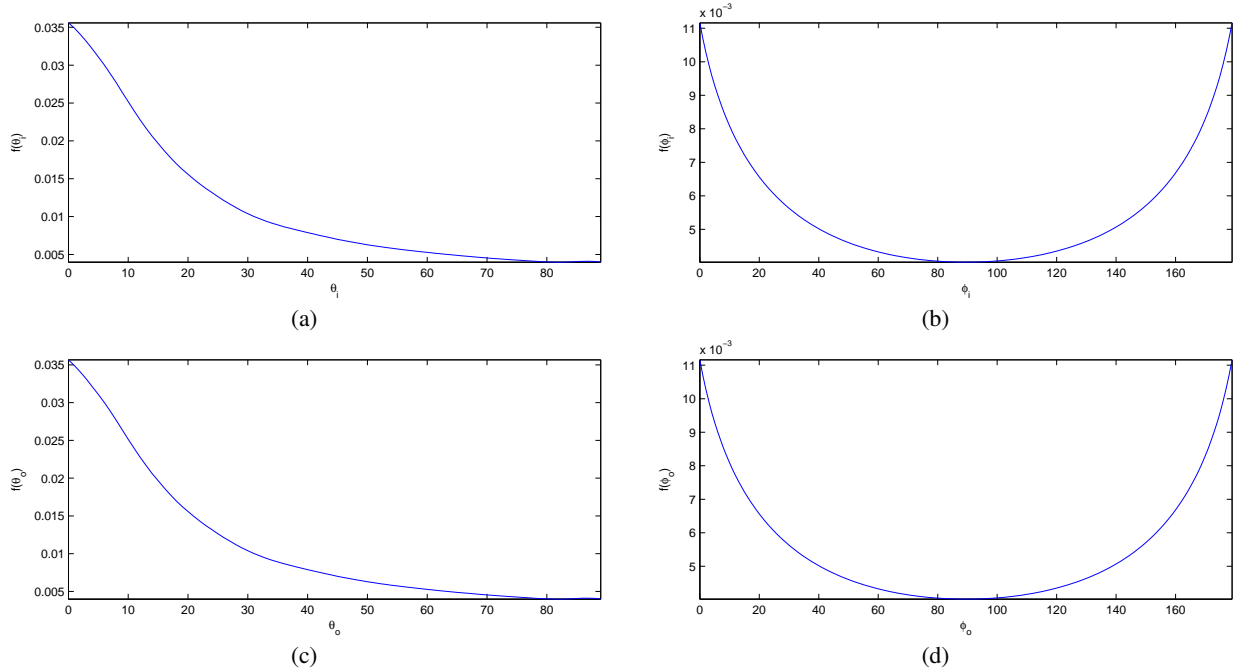


Figure 1: Empirical marginal distributions of elevation and azimuth angles for incident and outgoing directions (blue-metallic-paint).

ϕ_o provide a considerable flexibility. For a given outgoing direction, the distribution of $\vec{\omega}_i = (\theta_i, \phi_i)$ can be any bivariate pdf. However, such a representation usually does not produce a visually plausible BRDF [1]. Based on the measured BRDF data, Ngan et al. [8] have shown that the halfway vector representation of BRDF yields more visually plausible result than that of the standard representation. In this work we proceed to use the halfway vector representation of Rusinkiewicz [32] for isotropic materials. This representation depends on the halfway vector $\vec{\omega}_h = (\theta_h, \phi_h)$ and the difference vector $\vec{\omega}_d = (\theta_d, \phi_d)$. It is well known that isotropic BRDFs are independent of the angle ϕ_h and thus the corresponding pdf can be expressed as a function of three variables namely θ_h , θ_d and ϕ_d .

For an isotropic material the corresponding 3D BRDF copula model can be written as

$$\rho(\theta_h, \theta_d, \phi_d) = Kc(u_1, u_2, u_3; \alpha)f_1(\theta_h)f_2(\theta_d)f_3(\phi_d), \quad (13)$$

where $u_1 = F_1(\theta_h)$, $u_2 = F_2(\theta_d)$, $u_3 = F_3(\phi_d)$, c is 3D copula pdf, α is the copula parameter and K is the scaling coefficient obtained in a similar way for this 3D case as in Equation (10). The distribution functions F_1 , F_2 and F_3 are the cumulative marginal distribution functions of θ_h , θ_d and ϕ_d , respectively. These marginal distributions are estimated directly from the measured BRDF data.

5. ESTIMATION

Since we treated the normalized BRDF measurements as the observed probability densities corresponding to a bin in a 4D histogram, we can assume that the number of observations (frequencies) falling into these bins for this 3D case can be expressed in a similar way as

$$n_{rst} = nb_{rst}, \quad (14)$$

where $n = n_{\dots}$ represents the total number of frequencies which is unknown for our case. Any well known statistical estimation

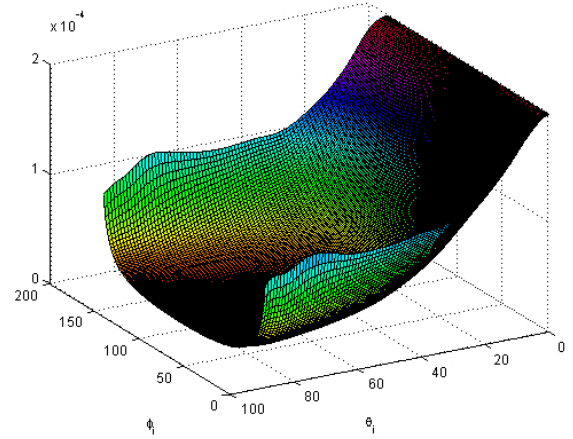


Figure 2: The joint pdf of θ_i and ϕ_i for blue-metallic-paint.

method can be adopted for estimating the unknown parameters of hypothesized distribution. For example, the maximum likelihood technique can be used straightforwardly for this purpose.

Based on the data matrix whose rows are organized as follows

$$\{b_{rst}, \theta_h^r, \theta_d^s, \phi_d^t\}, \quad (15)$$

the corresponding log-likelihood function for our special case can be written as

$$l(\lambda) = n \sum_r \sum_s \sum_t b_{rst} \log \{f(\lambda; \theta_h^r, \theta_d^s, \phi_d^t)\}, \quad (16)$$

where f is the joint pdf and λ is the vector of unknown parameters. The Archimedean copula distribution used in this work has a single

parameter α . It is interesting to note that the maximization of above objective function with respect to the parameters does not depend on the unknown sample size n . Thus the estimates of parameter can be obtained as a function of normalized measured BRDFs.

One difficulty in modeling BRDF by a multivariate probability distribution is the number of unknown parameters which is prohibitively large for a practical application. This problem could be overcome if the corresponding probability model can be constructed as a function of marginal distributions of underlying variables. In such case the estimation process can be performed in two stages: first the parameters of marginal distributions can be determined separately and then they are substituted in the objective function to estimate remaining parameters [33]. In this sense the copula distributions provide a unified approach both for modeling and estimating the BRDF.

Considering some practical difficulties that are encountered in the tails of distributions, we employed the least squares technique instead of the maximum likelihood technique for estimating the dependency parameter of copula pdf. The least squares estimation procedure is explained in detail in [2]. The objective function for obtaining the non-linear least squares estimate of α is defined as

$$S(\alpha) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^r \{b_{ijk} - c(F_1^{(i)}, F_2^{(j)}, F_3^{(k)}; \alpha) f_1^{(i)} f_2^{(j)} f_3^{(k)}\}^2, \quad (17)$$

where $f_1^{(i)} = b_{i..}$, $f_2^{(j)} = b_{.j.}$, $f_3^{(k)} = b_{..k}$ and F_1, F_2 and F_3 are cumulative marginal distributions.

We have observed that the marginal distributions of θ_h for specular materials are extremely skewed [2]. Our empirical results showed that copula distributions do not provide satisfactory approximations for these cases. We overcome this difficulty by dividing data along θ_d into subsamples and fitting the BRDF model to each of these subsamples. In this work, we obtained 6 subsamples by dividing θ_d into 6 non overlapping intervals each with a length of $90^\circ/6 = 15^\circ$.

6. PROPERTIES OF THE MODEL

A physically plausible BRDF model should obey the reciprocity and the energy conservation properties. Our BRDF model lends itself to satisfy both reciprocity and energy conservation properties.

Reciprocity: The reciprocity property of BRDF is expressed by the following equation

$$\rho(\vec{\omega}_i, \vec{\omega}_o) = \rho(\vec{\omega}_o, \vec{\omega}_i). \quad (18)$$

That is the BRDF is unchanged when the order of $\vec{\omega}_i$ and $\vec{\omega}_o$ is changed. Our model depends on the halfway vector representation. To enforce the reciprocity condition we use the identity

$$\phi_d = \phi_o + \pi. \quad (19)$$

Under this enforcement, our model satisfies the reciprocity principle.

Energy conservation: Energy conservation property of a BRDF model dictates that for every outgoing light direction

$$\int_{\Omega} \rho(\vec{\omega}_i, \vec{\omega}_o) (\vec{\omega}_i \cdot \mathbf{n}) d\vec{\omega}_i \leq 1. \quad (20)$$

Our BRDF model depends on a multivariate probability distribution function but it is scaled with a coefficient K . In this sense, our model may not be considered as an energy conserving model. However, one can modify this coefficient to account for absorption of different wavelengths of light as suggested by Edwards et al. [1].

7. IMPORTANCE SAMPLING

In this section we describe a sampling technique that can be used for our BRDF representation. Our main goal is to evaluate the following integral for a given outgoing direction:

$$L_o(\vec{\omega}_o) = \int_{\Omega} L_i(\vec{\omega}_i) \rho(\vec{\omega}_i, \vec{\omega}_o) \cos \theta_i d\vec{\omega}_i, \quad (21)$$

where Ω stands for unit hemisphere above the surface. We need to develop an efficient sampling procedure since direct or numerical evaluation of the underlying integral is difficult. A common approach to tackle this problem is to use variance reduction techniques. This technique exploits the fact that an estimator $(1/n) \sum_{i=1}^n h(X_i)/f(X_i)$ where f is the pdf of X and n is the sample size, is an unbiased estimator of the integral $\int h(x)dx$ and if the function f is similar to the function h then this statistic converges to the integral more quickly.

At a given pixel position and an outgoing direction, the incident illumination integral is given by Equation (21) where $\rho(\vec{\omega}_i, \vec{\omega}_o)$ is the BRDF based on our representation. Since we assume that the outgoing direction is known, we can use the conditional pdf of $\vec{\omega}_i$ given that $\vec{\omega}_o$ to estimate the integral value as

$$L_o(\theta_o, \phi_o) = \frac{1}{n} \sum_{i=1}^n L_i(\theta_i, \phi_i) \frac{\rho(\theta_i, \phi_i, \theta_o, \phi_o) \cos \theta_i \sin \theta_i}{f(\theta_i, \phi_i | \theta_o, \phi_o)}, \quad (22)$$

where the conditional pdf in the denominator can be expressed in terms of the BRDF and the joint marginal density function h of θ_o and ϕ_o as

$$f(\theta_i, \phi_i | \theta_o, \phi_o) = \frac{\rho(\theta_i, \phi_i, \theta_o, \phi_o)}{h(\theta_o, \phi_o)}. \quad (23)$$

The marginal distribution of θ_o and ϕ_o can be estimated from the measured BRDF data. Substituting this conditional pdf in Equation (22), we get the following simplified expression

$$L_o(\theta_o, \phi_o) = \frac{1}{n} h(\theta_o, \phi_o) \sum_{i=1}^n L_i(\theta_i, \phi_i) \cos \theta_i \sin \theta_i. \quad (24)$$

It is interesting to note that the resulting Monte Carlo estimator of the illumination integral does not contain the BRDF and it can be computed easily for each incident vector. However, the corresponding incident vector has to be generated from the conditional distribution of θ_i and ϕ_i in Equation (23). It can be shown that the joint cumulative conditional distribution function of θ_i and ϕ_i is [34]

$$F(\theta_i, \phi_i | \theta_o, \phi_o) = \frac{\varphi^{-(2)}\{\varphi(u_1) + \varphi(u_2) + \varphi(u_3) + \varphi(u_4)\}}{\varphi^{-(2)}\{\varphi(u_1) + \varphi(u_2)\}}, \quad (25)$$

where $u_1 = F_1(\theta_o)$, $u_2 = F_2(\phi_o)$, $u_3 = F_3(\theta_i)$, $u_4 = F_4(\phi_i)$ and $\varphi^{-(2)}$ stands for the second derivative of the inverse generating function. An algorithm for generating random numbers from Frank distribution is provided by Genest and Rivest. However, generating random variates from the conditional bivariate Frank distribution using Equation (25) is not straightforward. For this purpose we generate two standard uniform random variates ξ_1 and ξ_2 first and then generate θ_i as the solution of

$$\xi_1 = F_{\tilde{\theta}_i}^-(\tilde{\theta}_i | \theta_o, \phi_o), \quad (26)$$

and ϕ_i as the solution of

$$\xi_2 = F_{\tilde{\phi}_i}^-(\tilde{\phi}_i | \theta_o, \phi_o, \tilde{\theta}_i). \quad (27)$$

These nonlinear equations can be solved using numerical techniques. Our work is being continued along the line of finding approximate closed form solutions for these equations.

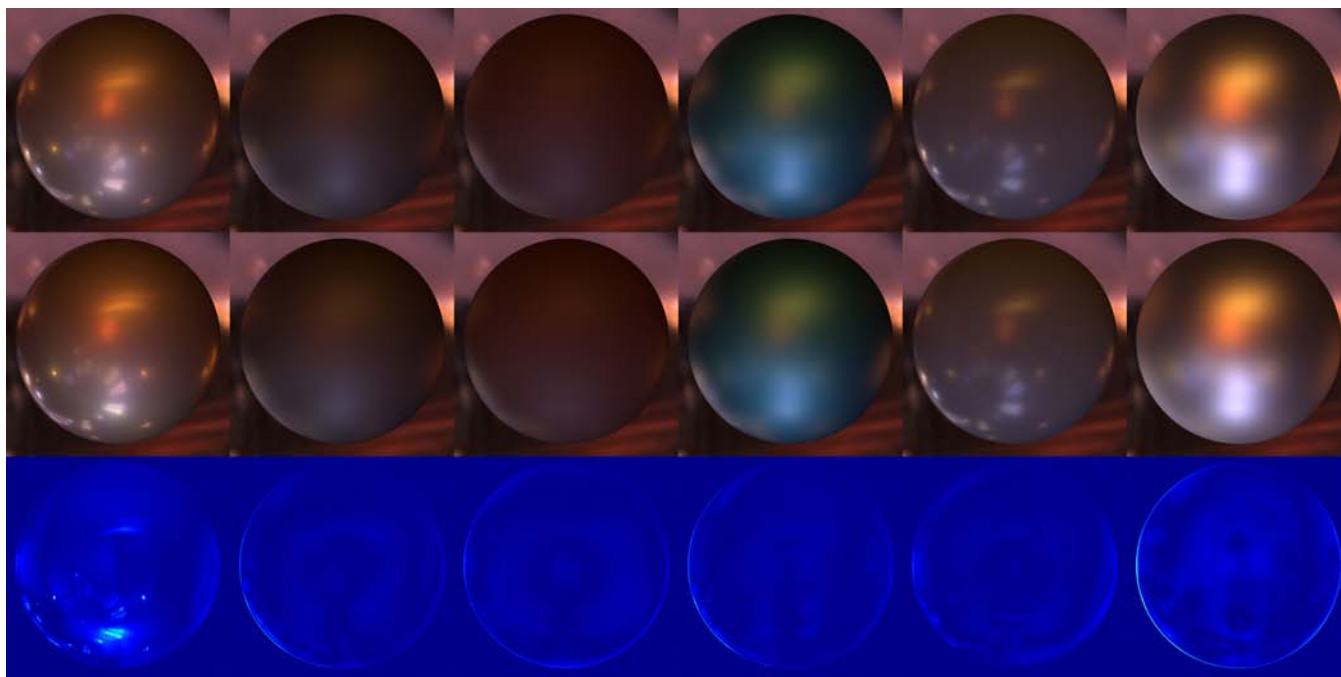


Figure 3: Various spheres were rendered with our Frank copula model using different materials. Columns left to right: alum-bronze, black-oxidized-steel, dark-specular-fabric, green-metallic-paint, pvc and silver-metallic-paint. Rows top to bottom: Reference images were rendered using measured data; images were rendered using our Frank copula model and color-coded difference images (Color-coded differences are scaled by a factor of five to improve the visibility of differences between the real and approximated images).

8. RESULTS

In this work we used the measured BRDF data of Matusik et al. [24] for isotropic materials. This data is sampled using the halfway representation because this representation is more suitable both for specular and for diffuse materials. All materials in this data set are sampled uniformly with a resolution $90 \times 90 \times 180$ for $(\theta_h, \theta_d, \phi_d)$, respectively. We have obtained empirical marginal distributions of θ_h, θ_d and ϕ_d and estimated the unknown parameters of Archimedean copula pdfs using this data set. To illustrate the quality of approximation of proposed model we fitted 3D Frank copula model to 6 different materials which include diffuse, glossy and specular materials. The results are presented in Figure 3. The spheres shown on the last row of figure are color-coded difference images [35] to improve the visibility of differences between the real and approximated images. Based on these materials the Frank copula model has produced satisfactory approximations for the real images.

Our model satisfies the two important properties of BRDF namely the reciprocity and energy conservation. The reciprocity property of our model is illustrated in Figure 4. In this figure we interchanged the incoming and outgoing direction vectors and rendered the Princeton scene [3], [1]. It is clearly seen in the color-coded difference image that the reciprocity property of our BRDF model is satisfied for this special case.

The energy conservation property of the model is demonstrated by using the procedure of Neumann et al. [36]. We evaluated the integral of Equation (20) for each outgoing directions and for three different isotropic materials. The results are shown in Figure 5. In these calculations no modification is made on the coefficient K . It is seen that our model satisfies the energy conserving property for each of the three materials. This property of our model has been verified by considering various isotropic materials.

9. CONCLUSION

In this paper we considered the problem of representing BRDF through a multivariate probability distribution and discussed some related problems. Archimedean family of copula distributions can be used to model the BRDF as a function of the corresponding cumulative marginal distributions. This class of distributions not only provides a simple functional form for modeling the BRDF but also lends itself to satisfy the reciprocity and energy conservation properties of an ideal physically-based BRDF.

Modeling the BRDF by a probability distribution has another advantage that is the Monte Carlo estimator of the illumination integral can be simplified and hence can be evaluated without using the estimated BRDF function. In this case, however, the incoming vector should be sampled using an appropriate simulation technique.

ACKNOWLEDGEMENT

This work was supported by the Scientific and Technical Council of Turkey (Project No: 108E007).

10. REFERENCES

- [1] Dave Edwards, Solomon Boulos, Jared Johnson, Peter Shirley, Michael Ashikhmin, Michael Stark, and Chris Wyman, "The halfway vector disk for brdf modeling," *ACM Transactions on Graphics*, vol. 25, no. 1, pp. 1–18, Jan. 2006.
- [2] Aydın Öztürk, Murat Kurt, and Ahmet Bilgili, "A copula-based brdf model," *Accepted in Computer Graphics Forum*, 2010, doi:10.1111/j.1467-8659.2010.01649.x.
- [3] Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi, "Efficient BRDF importance sampling using a



Figure 4: Left scene was rendered using Frank copula model. Middle scene was rendered using the same Frank copula model with incoming and outgoing vectors interchanged. Right scene shows a color-coded difference between left and middle images.

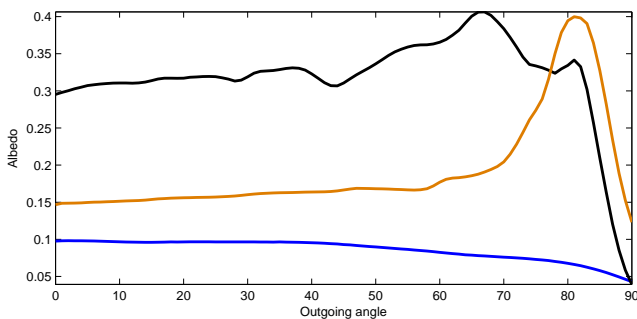


Figure 5: Albedo functions of Frank copula model for three measured materials: blue-metallic-paint (blue), yellow-matte-plastic (orange) and nickel (black).

factored representation,” *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 496–505, 2004, (Proc. SIGGRAPH ’04).

- [4] Bui Tuong Phong, “Illumination for computer generated pictures,” *Communications of the ACM*, vol. 18, no. 6, pp. 311–317, June 1975.
- [5] James F. Blinn, “Models of light reflection for computer synthesized pictures,” *Computer Graphics*, vol. 11, no. 2, pp. 192–198, 1977, (Proc. SIGGRAPH ’77).
- [6] Gregory J. Ward, “Measuring and modeling anisotropic reflection,” *Computer Graphics*, vol. 26, no. 2, pp. 265–272, 1992, (Proc. SIGGRAPH ’92).
- [7] Arne Duer, “On the Ward model for global illumination,” Unpublished material, 2005.
- [8] Addy Ngan, Frédo Durand, and Wojciech Matusik, “Experimental analysis of brdf models,” in *Proc. of Eurographics Symposium on Rendering*, Kavita Bala and Philip Dutré, Eds., Konstanz, Germany, 2005, pp. 117–126, Eurographics Association.
- [9] Eric P. Lafortune, Sing-Choong Foo, Kenneth E. Torrance, and Donald P. Greenberg, “Non-linear approximation of reflectance functions,” in *Proc. SIGGRAPH ’97*, New York, NY, USA, 1997, pp. 117–126, ACM Press/Addison-Wesley Publishing Co.
- [10] Michael Ashikhmin and Peter Shirley, “An anisotropic Phong BRDF model,” *Journal of Graphics Tools*, vol. 5, no. 2, pp. 25–32, 2000.
- [11] Robert R. Lewis, “Making shaders more physically plausible,” *Computer Graphics Forum*, vol. 13, no. 2, pp. 109–120, 1994.
- [12] Harold B. Westlund and Gary W. Meyer, “Applying appearance standards to light reflection models,” in *Proc. SIGGRAPH ’01*, New York, NY, USA, 2001, pp. 501–510, ACM.
- [13] K. E. Torrance and E. M. Sparrow, “Theory for off-specular reflection from roughened surfaces,” *Journal of the Optical Society of America*, vol. 57, no. 9, pp. 1105–1114, Sept. 1967.
- [14] Robert L. Cook and Kenneth E. Torrance, “A reflectance model for computer graphics,” *Computer Graphics*, vol. 15, no. 3, pp. 307–316, 1981, (Proc. SIGGRAPH ’81).
- [15] Michael Oren and Shree K. Nayar, “Generalization of lambert’s reflectance model,” in *Proc. SIGGRAPH ’94*, New York, NY, USA, 1994, pp. 239–246, ACM.
- [16] Xiao D. He, Kenneth E. Torrance, François X. Sillion, and Donald P. Greenberg, “A comprehensive physical model for light reflection,” *Computer Graphics*, vol. 25, no. 4, pp. 175–186, 1991, (Proc. SIGGRAPH ’91).
- [17] Michael Ashikhmin, Simon Premože, and Peter Shirley, “A microfacet-based brdf generator,” in *Proc. SIGGRAPH ’00*, New York, NY, USA, 2000, pp. 65–74, ACM Press/Addison-Wesley Publishing Co.
- [18] James T. Kajiya, “Anisotropic reflection models,” *Computer Graphics*, vol. 19, no. 3, pp. 15–21, 1985, (Proc. SIGGRAPH ’85).
- [19] Pierre Poulin and Alain Fournier, “A model for anisotropic reflection,” *Computer Graphics*, vol. 24, no. 4, pp. 273–282, 1990, (Proc. SIGGRAPH ’90).

- [20] Stephen H. Westin, James R. Arvo, and Kenneth E. Torrance, "Predicting reflectance functions from complex surfaces," *Computer Graphics*, vol. 26, no. 2, pp. 255–264, 1992, (Proc. SIGGRAPH '92).
- [21] Jan J. Koenderink, Andrea J. van Doorn, and Marigo Stavridi, "Bidirectional reflection distribution function expressed in terms of surface scattering modes," in *ECCV '96: Proc. of the 4th European Conference on Computer Vision-Volume II*, Bernard F. Buxton and Roberto Cipolla, Eds., London, UK, 1996, vol. 1065 of *Lecture Notes in Computer Science*, pp. 28–39, Springer-Verlag.
- [22] Peter Schröder and Wim Sweldens, "Spherical wavelets: efficiently representing functions on the sphere," in *Proc. SIGGRAPH '95*, New York, NY, USA, 1995, pp. 161–172, ACM.
- [23] Paul Lalonde and Alain Fournier, "A wavelet representation of reflectance functions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, no. 4, pp. 329–336, 1997.
- [24] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan, "A data-driven reflectance model," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 759–769, July 2003, (Proc. SIGGRAPH '03).
- [25] Aydin Ozturk, Murat Kurt, Ahmet Bilgili, and Cengiz Gungor, "Linear approximation of bidirectional reflectance distribution functions," *Computers & Graphics*, vol. 32, no. 2, pp. 149–158, Apr. 2008.
- [26] Alain Fournier, "Separating reflection functions for linear radiosity," in *Proc. of Eurographics Workshop on Rendering*, Pat Hanrahan and Werner Purgathofer, Eds., Dublin, Ireland, 1995, pp. 296–305, Springer.
- [27] Jan Kautz and Michael D. McCool, "Interactive rendering with arbitrary brdfs using separable approximations," in *Proc. of Eurographics Workshop on Rendering*, Dani Lischinski and Greg Ward Larson, Eds., Granada, Spain, 1999, pp. 247–260, Eurographics Association.
- [28] Michael D. McCool, Jason Ang, and Anis Ahmad, "Homomorphic factorization of brdfs for high-performance rendering," in *Proc. SIGGRAPH '01*, New York, NY, USA, 2001, pp. 171–178, ACM.
- [29] Rui Wang, John Tran, and David Luebke, "All-frequency relighting of non-diffuse objects using separable brdf approximation," in *Proc. of Eurographics Symposium on Rendering*, Alexander Keller and Henrik Wann Jensen, Eds., Norrköping, Sweden, 2004, pp. 345–354, Eurographics Association.
- [30] Xin Sun, Kun Zhou, Yanyun Chen, Stephen Lin, Jiaoying Shi, and Baining Guo, "Interactive relighting with dynamic brdfs," *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 27:1–27:10, 2007, (Proc. SIGGRAPH '07).
- [31] Christian Genest and Anne-Catherine Favre, "Everything you always wanted to know about copula modeling but were afraid to ask," *Journal of Hydrologic Engineering*, vol. 12, no. 4, pp. 347–368, 2007.
- [32] Szymon M. Rusinkiewicz, "A new change of variables for efficient brdf representation," in *Proc. of Eurographics Workshop on Rendering*, George Drettakis and Nelson L. Max, Eds., Vienna, Austria, 1998, pp. 11–22, Springer.
- [33] Harry Joe, *Multivariate Models and Dependence Concepts*, Chapman & Hall/CRC, Boca Raton ; London ; New York, 1997.
- [34] Edward W. Frees and Emiliano A. Valdez, "Understanding relationships using copulas," *North American Actuarial Journal*, vol. 2, no. 1, pp. 1–25, 2007.
- [35] Miloš Hašan, Jaroslav Křivánek, Bruce Walter, and Kavita Bala, "Virtual spherical lights for many-light rendering of glossy scenes," *ACM Transactions on Graphics*, vol. 28, no. 5, pp. 143:1–143:6, 2009, (Proc. SIGGRAPH Asia '09).
- [36] László Neumann, Attila Neumann, and László Szirmay-Kalos, "Compact metallic reflectance models," *Computer Graphics Forum*, vol. 18, no. 3, pp. 161–172, 1999, (Proc. Eurographics '99).

ABOUT THE AUTHOR

Aydın Öztürk is a professor at Yasar University, Department of Computer Engineering. His contact email is aydin.ozturk@yasar.edu.tr.

Murat Kurt is a Ph.D. student at Ege University, International Computer Institute. His contact email is murat.kurt@ege.edu.tr.

Ahmet Bilgili is a Ph.D. student at Ege University, International Computer Institute. His contact email is ahmetbilgili@gmail.com.

Pulse of the city: Visualizing Urban Dynamics of Special Events

Andrea Vaccari

Massachusetts Institute of
Technology

77 Massachusetts Avenue
Cambridge, MA 02139 USA
++ 1-617-324-5560

avaccari@mit.edu

Mauro Martino

Massachusetts Institute of
Technology

77 Massachusetts Avenue
Cambridge, MA 02139 USA
++ 1-617-324-5560

mmartino@mit.edu

Francisca Rojas

Massachusetts Institute of
Technology

77 Massachusetts Avenue
Cambridge, MA 02139 USA
++ 1-617-324-5560

fmr@mit.edu

Carlo Ratti

Massachusetts Institute of
Technology

77 Massachusetts Avenue
Cambridge, MA 02139 USA
++ 1-617-324-5560

ratti@mit.edu

ABSTRACT

How does a city perform during a special event? How can we inform local authorities and urban dwellers of public happenings? Until today it has been difficult to monitor urban dynamics in real time. Traditional methods, like head counts, surveys, and aerial inspection, are costly and limited. In this paper we present our twofold approach to the processing and visualization of real time information gathered from the telecommunication infrastructures that percolate modern cities.

This paper presents visualizations for a popular audience that examine the social psychogeography of the cityscape. Our work is a step towards understanding the functions of the city through digitally-enabled urban functions such as the pervasive telecommunications network. Based on an analysis of data from a telecom network, we illustrate how special events influence the normal rhythms of the city, and how crowds move and respond to large-scale public events.

This paper also discusses the potential of realizing an interface aimed at informing urban dwellers in real time of the dynamics that occur in the very place they find themselves. We contend that this type of public intervention could enable local authorities, service providers, businesses, and citizens themselves to function and behave more efficiently, making the places they inhabit more livable and sustainable.

This paper presents the two approaches, compares them to similar recent projects, and discusses the design choices that drove our implementation and the insights that we gained by their implementation.

Categories and Subject Descriptors

H.5.1 [Multimedia Information Systems]: Animations; [J.4 Social

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GraphiCon'2010, September 20–24, 2010, St.Petersburg, Russia.
Copyright 2010 ACM 1-58113-000-0/00/0010...\$10.00.

and Behavioral Sciences]: Sociology.

General Terms

Design, Human Factors.

Keywords

Urban Dynamics, Urban Informatics, Urban Interaction Design, Visualization for the Masses.

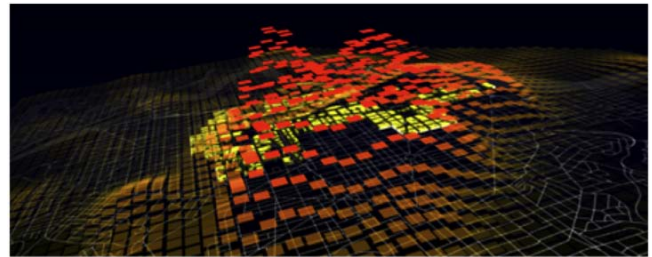


Fig. 1. *Obama / One People* [15] visualizes cell phone call activity during the Inauguration of President Barack Obama as a way to understand the urban dynamics of special events.

1. INTRODUCTION

Over the past decade there has been an explosion in the deployment of pervasive systems like cell phone networks and user-generated content aggregators on the Internet that produce massive amounts of data as a by-product of their interaction with users. This data is related to the actions of people and thereby to the overall dynamics of cities, how they function and evolve over time. Electronic logs of cell phone calls, subway rides, GPS-enabled buses, and geotagged photographs are all digital footprints [6] that today allow researchers to better understand how people flow through urban space, and could ultimately help those who manage and live in urban areas to configure more liveable, sustainable, and efficient cities [2].

At the same time, these pervasive systems present the possibility of extracting and inserting real-time information about social dynamics into the built environment. The possibility of establishing this dynamic feedback loop of information about how a city functions has the potential to influence many aspects of

urban management by assisting local authorities, service providers, enterprises, and even citizens themselves to make more accurate decisions and thereby create a more efficient urban environment. In order for this to happen, we contend that it is necessary to first educate the public in understanding how individual choices build up to form emerging urban processes that affect the city and its inhabitants as a whole.

Obama / One People [15] employs digital footprints from a major telecom operator's cell phone network to examine how people occupy urban space during a major special event. The project consists of two visualizations, *The City* and *The World*, which are aimed at a popular audience and represent the event from both a local and an international perspective. We chose President Barack Obama's inauguration day in Washington D.C. as a case to develop the special event visualizations and we partnered with a major telecom operator to use cell phone data as proxy of social activity. We designed and implemented the *Obama / One People* visualizations to provide aesthetically compelling answers to questions like: How does a city perform during a special event or, analogously, a sudden emergency? Where do people congregate and how does this vary over time?

It should be noted that there are many projects that focus on specific aspects of urban pervasive systems and the study of digital traces like cell phone calls, text messages and geotagged photographs [6, 13, 7] and the implementation of new mobile services [12, 1, 9] and interfaces [8, 14, 5]. Our approach in *Obama / One People* is slightly different. The goal of our visualizations is to translate the invisible digital traces of cell phones and relate these to urban space so that people can grasp the large-scale dynamics that occur within a cityscape. The ultimate intent of this work is to envision a future where urban actors can process existing information in real time directly from the city's digital infrastructure and make more informed decisions about their social behavior within the city.

The data presented in *Obama / One People* consists of hourly counts of cell phone calls per each base transceiver station serviced by a major telecom operator in the Washington, D.C. metro area along with the location where those phones are registered, by state for American phones and by country for international phones. The data set provided covered the period from January 10, 2009 to January 28, 2009, out of which we chose to represent the three days before and after the event, from the 17 to the 23. To ensure the complete privacy of the telecom provider's customers, the data is in accordance with the provider's privacy policies and we performed our analyses in compliance with the 2002 Directive of the European Parliament and Council on Privacy. Our use of aggregate data implies that we cannot identify individual callers, and we hope that our project might stimulate a dialogue on the responsible access to digitally-generated urban data and on how it can provide value-added services to local and regional communities.

This paper first reviews the relevant design works in the field of urban informatics, then discusses the design requirements we sought to achieve in *Obama / One People* and describes the visualizations themselves and their results, reviews the architecture designed to implement the project, and finally considers the limitations and practical applications of this type of work. All in all, we believe *Obama / One People* offers the following contributions to the field:

- It presents two visualizations that use existing digital

footprints to create an aesthetically compelling representation of the temporal and spatial aspects of a public event in a major urban area.

- It proposes how such visualizations can be accessed through digital interfaces to support real-time decision-making in the urban environment.

2. RELATED WORKS

Urban space and cartographic space are inseparable: from the grid to the aerial perspective to contemporary digital mapping technologies, the experience of urban space is always an important component in people's mental map of the city. The map has become a popular interface in illustrating data sets drawn from our increasingly digitally enabled urban infrastructures: from GPS [18, 17, 4] to cell phone networks [13, 11] and other objects equipped with radio frequency ID tags [4, 3]. The purposes behind these types of visualizations vary: from purely artistic endeavors to traffic monitoring to the uncovering of emerging patterns of urban activity.

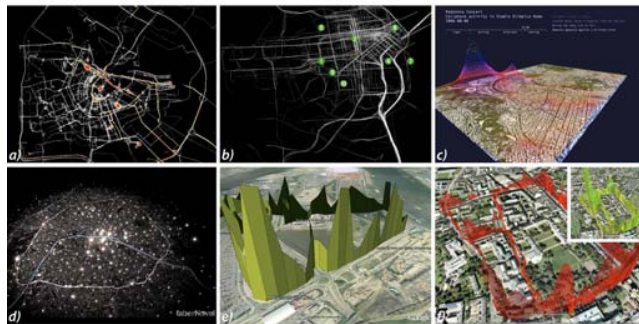


Fig. 2. Examples of visualizations of urban dynamics.

There seem to be four distinctive elements in the visualization of urban digital footprints:

Infrastructure and scalability. The type of data being used in the visualizations are generated by digitally-operated urban infrastructures in the course of the everyday functions of the city (RFID, WiFi hotspots, GPS, cell phones).

Flows and contexts. The data are displayed dynamically over time and in geographic space in order to represent flows of activity (i.e. pedestrians and cars). Some visualizations attempt to represent activity in almost real-time, often with a slight time lag for processing, such as a 15-minute delay.

2D and 3D perspective views. Depending on the objective of the visualization, some perspectives are in 2D, such as in representing traffic flows, and others are in 3D to present concepts like the volume of activity in urban space.

Levels of data aggregation. The scale at which the data is presented varies in detail and aggregation, such as in the case of sensitive information about individuals.

The following is a discussion of precedent projects that exemplify the most notable approaches to mapping urban dynamics and that offer starting points for illustrating the city-wide special event for *Obama / One People*.

Amsterdam Real Time (2003) by the Waag Society and Esther Polak [18] (Figure 1.a) focused on visualizing the connection

between the structure of the city and the movements of Amsterdam's inhabitants.

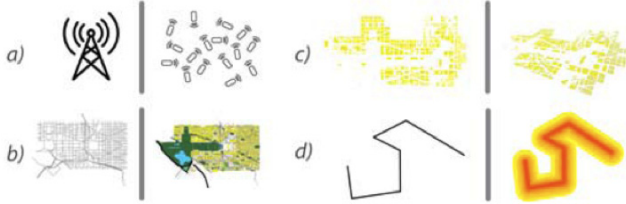


Fig. 3. Distinctive elements in the visualization of urban digital footprints.

By examining people's mobility through GPS, the project uncovered a map of the city defined solely by people's activity in space. Only GPS trajectories were recorded in the visualization and this urban flux revealed Amsterdam's urban form only by capturing the digital traces of people's movements (Figure 2.a). Similar to Cabspotting in San Francisco [17] (Figure 1.b) this visualization employs the flow information to structure the geographic space of the city, highlighting thoroughfares of movement but leaving out buildings and public spaces. Because of a lack of geographic references, it is difficult for those not familiar with the map of Amsterdam to understand the connection between the form of the built environment and the flows of people. Nevertheless (Figure 2.b), using GPS systems to visualize urban dynamics is a powerful way to detail (Figure 2.d) the movements of large numbers of people. Though as in all projects that use GPS, extending this kind of mapping to include larger number of individuals involves a high cost and a major organizational effort.

As mentioned previously, Cabspotting is similar to Amsterdam Real Time [18] in visualizing movement within the city and in its design approach. Directed by Scott Snibbe [17], Cabspotting uses GPS technology Figure 2.a in San Francisco's taxicabs to trace their movement in real time through the city. The patterns traced by each cab create a dynamic sense of traffic flows through the city's arteries, revealing where traffic is moving quickly and where it is congested.

Real Time Rome [13] (Figure 1.c) and UrbanMobs [11] (Figure 1.d) introduce a 3D perspective view (Figure 2.c) and provide a sense of the collective emotions of a city. As the MIT SENSEable City Lab's contribution to the 2006 Venice Biennale, Real Time Rome took aggregated data from cell phones and mapped these calls onto the geography of the city during two special events over the summer of 2006: the World Cup finals match between Italy and France, and a Madonna concert. The visualizations show peaks in the volume of calls during stirring moments (such as Italy scoring a goal during the World Cup match) in a sense revealing the emotional signature of the city as well as where people are congregating. Similarly, Orange Labs and Faber

Novel developed UrbanMobs as a tool to showcase popular emotion cartography through the analysis and visualization of citywide cellular network traffic activity. Both examples signal a shift in both the aesthetic qualities of visualizing dynamic urban data and in the methodology of positioning cell phones within urban space according to the location of cell phone towers that service those calls. This methodology may lose the detail of GPS data but in tapping into the network infrastructure of cell phones, it can harness vast amounts of data representing large swaths of the city, thus increasing the scale of representation.

Measuring and mapping people's emotional responses to the built

environment itself is the goal of BioMapping by Christian Nold [4] (Figure 1.e). This is a community-mapping project that utilizes a GSR (Galvanic Skin Response) device to measure how people react psychologically to different areas of the city. Over 1,500 people have contributed to this exercise in various cities to produce these communal emotion maps. Presented in a 3D view, the visualizations use Google Earth to relate people's emotional arousal as measured by the GSR device to the geographic locations they traverse. BioMapping clearly presents the urban context of people's reactions to the built environment, and the GSR sensors and GPS scale the visualizations down to show individuals' responses; however the number of people involved limits the extent to which these maps may present a collective psychogeography of the city.

Another project that visualizes data gathered directly from consenting individuals is CamMobSense in Cambridge, UK [3] (Figure 1.f). Using sensors mounted on pedestrians and cyclists, CamMobSense monitors pollution in the city and relays the collected data to a website in real time. It presents a low-cost, distributed sensor model where people themselves collect data that can be then visualized to better understand the environmental impacts of urban functions such as transportation, and in turn this can help those who manage urban regions make more informed decisions.

Building upon these precedents, *Obama / One People* utilizes aggregated data from cell phones and overlays the call dynamics onto an aerial photograph of Washington D.C., which highlights the event spaces of the Presidential Inauguration such as the Mall and Pennsylvania Avenue. We present the visualization in 3D to highlight the peaks and troughs in cell phone activity before, during and after the special event itself, totalling a week's time of data. While the aggregated data does not reveal details of individual movements, it does reveal collective dynamics over the course of a week in large swaths of the Nation's Capital. We have thus created a scalable system that is able to capture the call activity of large numbers of cell phones resulting in a visualization that highlights the communications flows of a crowd and how these relate to geographic locations within the city.

3. DESIGN REQUIREMENTS

In light of the above discussion, we identified the following specific design requirements to guide the design of *Obama / One People*. The visualizations should create a "digital skin" of urban spaces, with a simple and clear visual language system. They should convey the social dynamics of a crowd, which are real phenomena, using informational data. The user should perceive the former, not the latter. They should establish a strong relationship between the crowd and the urban landscape. For this reason, we considered the spatial representation of the information as a given, reflecting the geographic nature of the data.

The visualizations should be interactive but simple, allowing the user to explore the evolution of the event in space and time. However, they are not meant to be visual analytics tools, so they should not provide quantitative details on the data. To achieve this, there should be a good balance between aesthetics and functionality. Inspired by Norman's work [10], we should use color, saturation, and luminosity to improve the aesthetic and emotional impact of the project.

Finally, from a technical perspective, *Obama / One People* should ideally run on a real-time stream of data. However, due

restrictions on the time required to collect the data from the telecom provider's network it was not possible for represent the event as it occurred.

4. OBAMA | ONE PEOPLE

On January 20, 2009, Washington D.C. hosted the largest crowd in its history for a very special event: the Inauguration of President Barack Obama. This crowd presented great challenges to both federal and municipal authorities in terms of management, transportation, security, and emergency response. Where would people congregate? How would they get there? When would they arrive and depart? From where were they coming? The authorities even had difficulty in estimating how many people would attend the event. Early predictions called for over 3 million people, and it was only after the week of the Inauguration that the City was able to release an official estimate of 1.7 million people. The 2009 Presidential Inauguration offers perhaps one of the most extreme cases of the value of gathering data about large crowds in real time so that decision makers can understand and manage special events as they occur.

In considering the volume of mobile phone calls at any given moment as indicators of a crowd's geographic distribution over time, our access to aggregated cell phone data from a large telecom operator presented us with the opportunity to explore the possibilities of using such data to understand the dynamics of large-scale special events such as the Presidential Inauguration. The *Obama | One People* visualizations sought to explore such questions as: Who was in Washington, D.C. for President Obama's Inauguration Day? When did they arrive, where did they go, and how long did they stay? The nature of interpersonal communications also allows us to interpret the resulting visualizations of call data as revealing the psychogeography of the city during a special event. Guy Debord, a French theorist, defined psychogeography in 1955 as the "the study of the precise laws and specific effects of the geographical environment, consciously organized or not, on the emotions and behavior of individuals." [TODO Guy Debord. 1955. Introduction to a Critique of Urban Geography (essay)]

Thus, another question *Obama | One People* explores is: when and where did the crowd in Washington D.C. sense the need to share thoughts, information, and feeling with others who were not co-present? Not only does *Obama | One People* map the invisible flows of communications over the geography of the city, it also begins to gauge the collective emotional pulse of the city at a large scale and over time. Below we discuss the two visualizations produced to present our analyses of aggregated phone calls in Washington D.C. during the week of President Barack Obama's Inauguration: The City and The World.

4.1 The City

The City illustrates two pieces of information about the Presidential Inauguration: 1) where the crowd came from and 2) the emotional flow of the massive event in Washington, D.C.

The City summarizes this information by relating call activity to the geography of Washington D.C. We overlay the map of the Nation's Capital with a 3D color-coded animated surface of square tiles, with one tile representing a geographic area of 150 x 150 meters. Each tile rises and turns red as call activity increases and likewise drops and turns yellow as call activity decreases. On the left side of the screen, a dynamic bar chart breaks down the call activity by showing the normalized contributions of calls

from the U.S.'s fifty states and 138 foreign countries grouped by continent. The timeline at the bottom illustrates the weeklong trend of call activity in Washington, D.C., which follows the 3D square tiles that rise and fall on top of the map of the city.

In visualizing the volume of calls served by a large telecom network in Washington DC, it is possible to see peaks of call activity as the crowd anticipates Obama's oath, a drop in call activity as the crowd listens to Obama's address, and peaks again as the crowd celebrates the inauguration of the new President. Through their cell phones, those present at the historic event share their impressions with friends and family in vast numbers: on the morning of January 20th, call activity is two to three times stronger than usual, and it rises to five times the normal levels after 2 pm as President Obama takes his oath and people begin to celebrate.

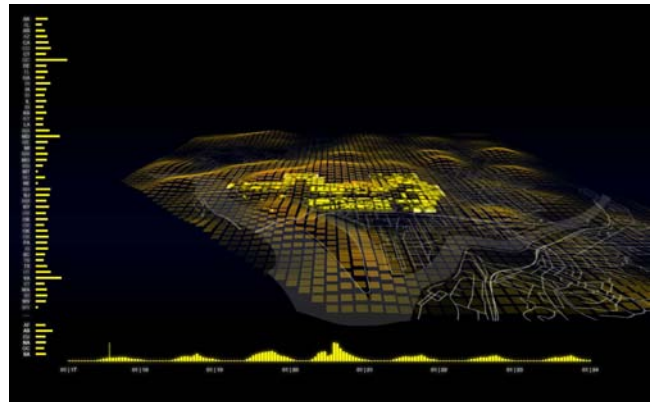


Fig. 4 The City visualizes call activity in the week of the inauguration. Superimposed on the map of Washington, D.C. is a 3-D color-coded animated surface of square tiles (1 tile represents an area of 150 x 150 meters). Each tile raises and turns red as call activity increases and likewise drops and turns yellow as activity decreases. On the left, a bar chart breaks down the call activity by showing the normalized contributions of calls from the 50 states and 138 foreign countries grouped by continent. The timeline at the bottom illustrates the overall trends of call activity in the federal areas of Washington, D.C., which are represented with 3-D yellow models on the map at the center of the screen.

4.2 The World

The World reveals the international nature of Inauguration Day. By using the area codes and country codes of the cell phones present in Washington D.C. during the week of the Inauguration, this visualization traces the trajectories of people travelling from all over the U.S. and the world. We interpret the variations in call activity as flows of people arriving in Washington, D.C. and then departing the capital to go back home. The visualization employs a world map to link the city of Washington to US states and countries abroad. Packets of information representing 100 calls for US states and 10 calls for foreign countries move to and from Washington depending on whether call activity increased or decreased in relation to the previous hour. The timeline on the bottom of the screen shows the overall trend of call activity in Washington, D.C., relating the flows of people with events in the city.

This visualization shows that people from almost every corner of the world and almost all fifty states attended the Inauguration of

President Barack Obama. The aggregated call data indicates that, at the least, there were people present from 138 countries, totaling over half of all the countries in the world. This is likely a low estimate since the data represents only one domestic telecom carrier. Among the foreign countries represented, the main international callers are from Canada, Great Britain, France, and Puerto Rico, which register a five-fold increase in call activity. In the U.S., the top calling states are also the country's most populous: California, Florida, New York, and Texas. Notably, Georgia also figures in the list of top five callers on Inauguration Day, even though it ranks ninth in US population.

4.3 Implementation Choices

In the previous section we described the design requirements of the two visualizations. Here we explain the thinking that motivated our implementation choices with respect to the those requirements.

First, we chose to represent the data from two perspectives. The City shows the evolution of the event within the context of Washington D.C. and allows users to see the build up in anticipation of the Inauguration, and how the crowd faded out after its conclusion. The World represents the international nature of the event and aims to convey the idea of a global pilgrimage to the American capital by displaying a map of the world and showing how people from near and far gathered in one city for one reason. In both visualizations, one aim is to represent the social dynamics that enveloped the city before, during, and after the President's Inauguration. Another aim is to help users understand how participation in a public event, like other individual choices in the context of the urban environment, builds up to create a collective social dynamic that affects the city as a whole. For this second reason, we purposefully deemphasized the quantitative nature of the data by using the visual metaphors of waves and packets.



Fig. 5. The World visualizes call traffic to and from the capital. Variations in call activity are represented here as flows of people coming to Washington, D.C. and then leaving the capital to go back to their home states and countries. The world map links Washington, D.C. to capitals abroad. Packets move to and from Washington depending on whether call activity increased or decreased in relation to the previous hour. The timeline on the bottom of the screen shows the overall trends of call activity in the federal areas of Washington, D.C., allowing to associate the flows of people with the events highlighted in The City.

In formulating how the data representation would occur, we considered more abstract visualizations of the flow of phone calls but instead decided to use maps given the inherently geographic nature of the data and the message we aim to convey about the influence of special events on urban space. Moreover, we employ map details to assist users in orienting themselves within the context of Washington D.C. without overloading the visualization with unnecessary details. For this reason, The City visualization highlights only the major roads of Washington D.C. and the Potomac River, while The World simplifies the representation of flows by showing only connections between Washington D.C., US state capitals and the countries around the world.

While we simplified the representation Washington's geography, we decided to represent call activity in The City with some redundancy along different features of the visualization: we used color, opacity, and position of the tiles to represent the intensity of activity within each cell. The combination of these three features makes for a strong representation. Color varies from unsaturated yellow to saturated red to represent the volume of activity (a feature sensitive to color blind users). The use of opacity hides those tiles where activity is low and highlights those where it is strong, thus guiding the user through the evolution of the event. Finally, the third dimension recreates the wave metaphor mentioned above and further helps to convey the idea of the city as a living system.

In The World visualization we represented the origin of the callers' phones only using packets of information. In this case, employing multiple features as in The City would have focused attention on those connections with the strongest activity, whereas in this case we sought to highlight the overall exchange of inflows and outflows happening in Washington D.C. For this, identical packets are a strong solution as their movements alone create denser or thinner flows emerging from Washington D.C.

The timelines placed at the bottom of the screen unify the two visualizations and highlight the weeklong trend of activity in the areas of the Washington D.C. where the Inaugural events took place. This common element is intended to help the user associate the information represented in one visualization with that being represented in the other. The two visualizations also share a minimalist interaction system that allow the user to navigate the 3D space with the mouse, using left and right clicks to rotate and span, and the mouse wheel to zoom in and out. Keyboard arrows can be used to pause and play the animation, and to move backward and forward in time.

4.4 Results

The visualizations of aggregated call activity during the Presidential Inauguration we created for *Obama | One People* have provided insights into the dynamics of a remarkable special event in Washington D.C. in terms of the composition of the crowd and its emotional behavior in space and over time.

By examining the relative increase in call activity during the week of the Inauguration as compared to a normal week of call activity in the Nation's Capital, it is possible to get a sense of who travelled to Washington to attend this special event. The states with the strongest increase in calls were the southern states of Alabama, Georgia, Kentucky and Tennessee, with calls up to twelve times the normal levels. These are states that played a prominent role in the Civil Rights movement and notably are also red states whose voting population went for the Republican

candidate, John McCain. Other states with a ten-fold increase in call activity were Illinois, Barack Obama's home state, and Michigan, Ohio and Indiana, swing states which went blue, voting for President Obama. These results illustrate that participation in the event from each U.S. state did not depend on its political leaning or geographical proximity to Washington, D.C.

As the visualizations show, call activity in the days before and after January 20 reveal that the Inauguration was a multi-day event as traffic increased markedly throughout the week. On Sunday, January 18th call activity began to increase and, further analysis reveals that on Monday it was already two to three times what it would normally be on a Monday in January in Washington D.C. On the day of the Inauguration, call activity reached unprecedented levels throughout the day and the telecom provider's network served six times more SMS than normal levels. In the morning, call activity was two to three times stronger than normal levels as the crowd anticipated Barack Obama's oath, then dropped as the crowd listened to President Obama's inaugural address and peaked to five times the normal levels after 2 pm as President Obama took the oath and people celebrated.

Geographically, The City visualization reveal how the Inaugural visitors spread beyond the confines of the Federal areas of Washington, D.C. - such as the Mall and Pennsylvania Avenue - into the neighborhoods of the city. The hotspots of activity were clustered in the Northwest neighborhoods of the city, around Downtown, Adams Morgan and U Street. The historic inauguration of President Obama offered people from all over the United States and the world the occasion for a truly urban celebration. While these results are not surprising in themselves, our analyses allow for the first time to quantify and compare the presence of people in different areas of the city and to reveal their dynamic movements and, to a certain extent, their emotional reactions through time.

In post-project discussions with the telecom provider we learned that while the origin of the crowd is not a key factor driving capacity planning decisions of network operators, the remaining questions are relevant. The visualization capabilities presented in this paper made it possible to understand that the Inauguration was a multi-day event that began two days before January 20th and which did not end immediately, stretching visitors' presence in the Nation's Capital through the middle of the week. The telecom operator appreciated the visualizations' simplicity and their ability to summarize vast amounts of data, both temporal and geographical. *Obama | One People* also convinced network engineers of the benefits of collecting, analyzing and visualizing digital footprints of telecommunications flows. This in turn led to a new partnership with the radio access network engineers to get access to more data and to get the support of their management for further analysis.

Future user studies should include those who manage municipal services during special events who often must make educated guesses about crowd capacity in the city's public spaces. Moreover, it would be valuable to gauge how the information transmitted through these visualizations could alter the behavioral decisions of the participants of special events themselves. Would they decide to arrive at the event earlier or later? Would they choose to reorganize themselves in the public spaces of the as they learn the real time dynamics of the crowd? Ultimately, what these visualizations aim to do is to create a feedback loop that provides real time information about urban dynamics back to

people so that they can adjust their decision-making as they go about their daily activities.

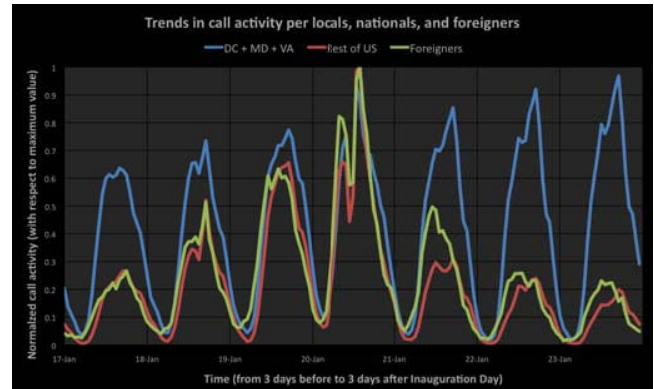


Fig. 6. Trends of normalized call activity before, during, and after the event.

5. SYSTEM IMPLEMENTATION

We developed *Obama | One People* on top of an agile architecture that supports the operations of data gathering, storage, and analysis. These three operations exist in three separate layers and we implemented the system in the Python programming language, using the open-source PostgreSQL database system and the PostGIS geospatial extension. Figure X details the workflow between the layers, which we describe below.

5.1 Data Gathering

This layer provides standard tools for uploading urban data. It consists of a secure FTP server and a server daemon. The server allows the telecom operator to upload CSV files containing data on the network activity in the last 60 minutes. The daemon detects when a new file has been loaded, then parses it and loads the data in the Data Storage layer.

We chose not to implement more advanced data streaming technologies to simplify the interaction with the telecom operator. This solution allowed us to obtain data in different formats and resolutions and to rapidly restructure it in a form that is compatible with the data storage layer. However, this solution did not allow us to visualize network activity in real time. Nevertheless, we were able to access the data and complete the visualizations within a couple of weeks of the Inaugural event.

5.2 Data Storage

This layer stores the network activity data into a geospatial-enabled database, which also stores information about the location of the base transceiver stations and on the geographical structure of the spatial grid used in The City visualization. Each data point has a time reference and a geographic reference to the country or state where the phones making calls are registered, which allows inter-country normalization and continent-scale aggregation of the data.

Since in this dataset the activity generated by U.S. phones is on average two orders of magnitude higher than that generated by foreign phones, we chose to normalize all time series in the range [0; 1]. To do this, we divided the activity by the maximum reached by each state or country within the time span of our data set. In most cases, the maximum was right after President Obama's Inaugural speech (see Figure 6).

5.3 Data Analysis

This layer accesses data from the Data Storage layer and performs geospatial operations that redistribute the punctual information on network activity over the two-dimensional Washington, D.C. area. The Data Analysis layer is also in charge of performing pre-processing operations on the data, like correcting missing values and normalization. The geospatial operation adopted to redistribute the network activity visualized in The City was based on a discrete grid of 150x150 meter cells that cover the Washington, D.C. area. The activity in each cell is estimated by summing the contributions of all the base transceiver stations weighted according to the distance between the cell and the station. The weighting factor is defined by the exponential decay function $w = \exp(-\text{dist}(\text{cell}, \text{bts})^2 / 0.2)$ which causes the weighting factor to become insignificant at the reasonable distance of 1 km from the station. The operation chosen to estimate the flows visualized in The World was based on the gradient of the network activity generated by each US state or country. Under the assumption that the average number of calls per person does not change radically over the course of the special event, we estimated a larger presence of people from a given state or country when the related call activity increased from one hour to the next. To account for normal daily fluctuations, we subtracted the average variations and represented only unusual fluctuations.

The main objective of our analyses was to summarize the data collected by the cell phone network into two easily understandable metaphors of social dynamics and urban communications flows. We consciously avoided using more invasive analyses, for example using radio wave propagation models to redistribute the data, in favor of an approach that focuses the viewers' attention on the high-level understanding about urban processes that this data can provide.

6. LIMITATIONS

Up to this point we have not able to validate the efficacy of the visualizations as the project is still confidential and cannot be shown outside our research group (we have specific permission for this publication). In this section we describe the major insights and limitations of our approach, which will be addressed in a future user study. Thus far, we have been able to collect informal feedback from the research scientists of the telecom operator, which supports our design and implementation choices. Their positive feedback confirms that the *Obama | One People* visualizations offer a new perspective on the nature and dynamics of a special event such as Inauguration Day. While the main results are not surprising - it is to be expected that call activity would increase markedly on the day of the event - they nevertheless reinforce assumptions about how people may behave during large-scale urban events. As service providers, the telecom operator valued the insight that there was a slow build-up to the actual day of the Inauguration and likewise a slow retreat of the crowd from the city over the course of several days. This may present a future scenario where the provider deploys extra telecom capacity for a special event a few days before the event and leaves it up and running for longer than expected.

Moreover, while most of the operator's feedback praised the simplicity of the interface and the visual system, some would have liked to see more analytical tools included in the visualization so that users could themselves examine the data. In the future, therefore, we intend to create an interface that allows the experts to study more in detail the data while maintaining a simple

interface for basic users. On the technical side, we limited our visualizations to cell phone activity recognizing that there are other forms of digital footprints. Works that used vehicle traffic data, for example, have been discussed in the section on Related Works. While it is not clear which datum or combination of data is the best proxy for estimating social activity, our visualizations could be immediately used with traffic data. Again, however, the focus of our project is not on quantitative precision but rather on the qualitative narrative.

Finally the architecture implemented is quite simple. While this allows designers to easily adapt to changes in format and resolution of the data, it does not allow the gathering, storage, and analysis of data in real time. This is a major drawback that is due to the slow process of data collection on the network side by the telecom operator. In the future, however, as information will flow more freely, a more advanced system architecture that fully supports real time data streams will be necessary. In that scenario, the simple geospatial procedures on the data that we perform will not clog the system.

7. DISCUSSION

Obama | One People is part of a long term vision centered on the idea that diffused social consciousness regarding urban dynamics can improve the efficiency of cities by enabling local authorities, service providers, businesses, and citizens to make better informed decision in their urban landscape. This vision was presented in *WikiCity* [16] (see Figure 7) that considered another public event, the *Notte Bianca* in Rome, Italy, an all-night festival that hosts artistic performances at various locations throughout the city. In that occasion, we developed a public interface that featured a satellite image of Rome with four types of information overlaid on top of it: real-time cell phone activity, the real-time location of public transport buses, starting and ongoing event tags at their corresponding location, and live news feeds from journalists at key locations. While the scope of this paper is not to present *WikiCity*, we believe it is important to discuss its underlying concept to contextualize *Obama | One People*.

Through *WikiCity* we envision the participatory provisioning of urban information by local authorities, service providers, businesses, and citizens, and the enabling of semantic processes for the development of an evolving hierarchy of analysis tools and information visualizations. These mechanisms, coupled with ubiquitous interfaces, are expected to enact the active use of digital technologies for real time location-based decision making in the city. In a similar fashion, they are intended to support the active collection of public feedback on the condition of the city and its services. The mapping of urban information is not limited to providing a real time representation of the city. Instead it becomes an instrument for citizens and local authorities to base their actions and decisions in a better informed manner. In this way the real time information changes the city context as well as is changed by that altered context accordingly. The potential in this scenario is to address a wide variety of problems that interest the multifaceted aspects of managing the city as a complex system. The results of *WikiCity* were mixed. If on one side it proved the technical feasibility of our vision, on the other it failed in creating a user base and the visualization was not adopted for subsequent major public events. While there are many possible reasons behind this, we believe that one of the major issues was that our interface tried to convey too much information to a public that was almost completely unaware of the significance of that

information. We see *Obama | One People* as an intermediate step in dramatically illustrating and popularizing the idea of the city as a living system that responds collectively to the actions and decisions of individuals.

8. CONCLUSION

This paper discussed a visualization project that considered how crowds behave during large-scale public events in urban areas. *Obama | One People* took as its case the Inauguration of President Barack Obama and overlaid telecommunications data over the geography of Washington D.C. The two resulting visualizations – *The City* and *The World* – showed from a local and an international perspective that the Inauguration was a multi-day event and that it drew people from all over the US and the world. The potential of visualizations such as *Obama | One People* are great for those who manage and act within urban space. They are a first step in revealing collective patterns of social dynamics, and data such as telephone call activity can offer a insights into how people inhabit the city in different times and locations, and how is the urban environment can be stressed in the case of public events or sudden emergencies.

Having the results of these analyses in real-time, urban planners would be able to promptly detect and correct phenomena that reduce the liveability and sustainability of the city: instead of planning urban interventions and waiting months to evaluate their impact, they could switch to a more reactive, real-time management of the city. Moreover, these kinds of visualizations, if accessed through an appropriate interface, can also be of value to urban dwellers themselves. The ultimate goal of this work is to build urban interfaces where people can deposit and extract information on the functions of the city in real time and thus work together in building more efficient, intelligent and sustainable cities.



Fig. 7. *WikiCity* [16] was publicly displayed during a major event in Rome, Italy and presented real-time cell phone activity, the real-time location of public transport buses, starting and ongoing event tags at their corresponding location, and live news feeds from journalists at key locations.

9. REFERENCES

- [1] M. Arikawa, S. Konomi, and K. Ohnishi. NAVITIME: Supporting pedestrian navigation in the real world. *IEEE Pervasive Computing Magazine*, 2007.

- [2] F. Calabrese, K. Kloeckl, and C. Ratti. *Handbook of Research on Urban Informatics*, chapter *WikiCity: Real-Time Location-Sensitive Tools for the City*. IGI Global, 2008.
- [3] Centre for Scientific Computing, University of Cambridge in collaboration with Imperial College London, Leeds University, Newcastle University and Southampton University. *CamMobSens*, <http://www.escience.cam.ac.uk/mobiledata>, 20 February 2010.
- [4] Christian Nold. *Bio Mapping*, <http://biomapping.net/index.htm>, 15 February 2010.
- [5] S. Eisenman, N. Lane, E. Miluzzo, R. Peterson, G. Ahn, and A. Campbell. *Metrosense project - people-centric sensing at scalAe*. *CM Conference on Embedded Networked Sensor System*, 2s006.
- [6] F. Girardin, F. Calabrese, F. Dal Fiore, C. Ratti, and J. Blat. *Digital footprinting: Uncovering tourists with user-generated content*. *IEEE Pervasive Computing Magazine*, 7(4):36–43, 2008.
- [7] M. Gonzalez, C. Hidalgo, and A. Barabasi. *Understanding individual human mobility patterns*. *Nature Magazine*, 2008.
- [8] J. Liu and F. Zhao. *Towards semantic services for sensor-rich information systems*. *2nd IEEE/CreateNet International Workshop on Broadband Advanced Sensor Networks*, 2005.
- [9] R. Murty, A. Gosain, M. Tierney, A. Brody, A. Fahad, J. Bers, and M. Welsh. *Citysense - an urban-scale wireless networking testbed*. *IEEE Conference on Technologies for Homeland Security*, 2007.
- [10] D. Norman. *Emotional Design: Why we love (or hate) everyday things*. Basic Books, 2004.
- [11] Orange Labs and Faber Nove. *UrbanMobs*, <http://www.fabernovel.com>, Revised in 22 April 2010.
- [12] A. Pawling, T. Schoenharl, P. Yan, and G. Madey. *Wiper: An emergency response system*. *6th International Conference on Information Systems for Crisis Response and Management*, 2008.
- [13] J. Reades, F. Calabrese, A. Sevtsuk, and C. Ratti. *Cellular census: Explorations in urban data collection*. *IEEE Pervasive Computing Magazine*, 6(3):30–38, 2007.
- [14] A. Santanche, S. Nath, J. Liu, B. Priyantha, and F. Zhao. *Senseweb - browsing the physical world in real time*. Microsoft Research, 2006.
- [15] Senseable City Lab, *Obama | One People*, <http://senseable.mit.edu/obama>, Revised in 6 May 2010.
- [16] Senseable City Lab, *WikiCity*, <http://senseable.mit.edu/wikicity>, Revised in 6 May 2010.
- [17] Stamen Design, Produced by Peter Richards and Susan Schwartzenberg, directed by Scott Snibbe. *Cabspotting*, <http://cabspotting.org>, Revised in 20 May 2010.
- [18] Waag Society together with Esther Polak. *Amsterdam RealTime*, <http://senseable.mit.edu/wikicity/>, Revised in 15 April 2010.

An Efficient Slice-Based Water Simulation Approach for Breaking Waves

Chung-Nan Lee, Jyun-Ming Wu, Yan-Zheng Huang, Da-Jing Zhang-Jian
Department of Computer Science and Engineering National Sun Yat-sen University

ABSTRACT

The simulation of breaking wave has a computationally intensive application. To reduce the computation this paper presents a slice-based water simulation method for ocean breaking waves on natural simulation by generating the 2D simulations and then integrating these 2D simulation results into a 3D shape. We first simulate a 2D wave by a 2D Navier-Stokes (NS) solver to obtain the varying of ocean. Then, we propose a new reconstruction free surface method via Volume of Fluid (VOF) to further reduce the computation time and achieve better efficiency by using a linear interpolation with noise function to construct a complete 3D ocean simulation from these 2D simulations.

Keywords: *Slice Method, Volume of Fluid, Breaking Wave*

1. INTRODUCTION

Physics-based natural simulation of computer animation becomes popular in recent years. The 3D graphics hardware has been widely applied to various fields with the rapid progress of multimedia system and virtual reality. In natural simulation, it was difficult to render the ocean on the PC, due to the restriction of computing power in the past. Fortunately, after the first graphics processing unit introduced by NVIDIA in 1999 that integrating transform, lighting, triangle setup/clipping, and rendering engines is capable of processing a minimum of 10 million polygons per second, as a result, the natural simulation and many other 3D graphic applications become feasible. Speed and realism are two mainly considerations in breaking wave simulation. For the game purpose the speed is mainly concerned. For the physical research, "realism" is very important due to users hope to get precise image or results.

For ocean simulation one can use Navier-Stokes equations (NSE) to describe the motion of incompressible fluid. NSE is viscous Newtonian fluid, and can calculate the fluid motion accurately and simulate the effect effectively. The breaking wave needs much more calculation to simulate the fluid situation.

In addition, most of existing studies about real-time water simulation cannot simulate breaking wave. The breaking wave simulation spends a lot of time because it needs complex physical calculation. In order to simulate the breaking waves in real time, this paper proposes a slice-based fluid simulation, and combines linear interpolation with noise function to construct a complete 3D fluid simulation. The goal of this paper is to get a fast breaking wave simulation without loss of too much realism.

2. Related Work

Because the focus of computer animation is perceived by the human eye that can be fooled the visual on clever in the movies or games even if it is incorrect. Therefore, they often focus on real-time simulation on current research. There are two types of fluid frameworks Eulerian and Lagrangian for the ocean simulation. In [2] a hybrid method, called particle level set method that combines Eulerian with Lagrangian framework is proposed.

It is very important to get the free surface on the ocean simulation. According to the definition of Kinetics, the stress of the tangent direction sum is equal to zero. Assuming that the particle system based on the Lagrangian framework has enough number of particles, the particle will form automatically free surface. It needs to obtain the free surface on Eulerian framework, with the complicated calculation of boundary. Currently based on the grid method, there are some methods such as MAC grid, level set, particle level set, VOF, front tracking, ghost fluid, and height field.

2.1 Height field

Ocean wave is a sine wave alike. Breaking waves are traditionally modeled using sinusoidal and trochoidal functions [16]. It gets height of the surface by the height field function. The height function method doesn't consider the eddy varying under the water, and the simulation is fast and has good result in quality. Height field has better effect when sea is calm, but the result is worse for windy weather. There is one problem for height field [8], when simulating the shallow water, due to it cannot generate the breaking wave. Tessendorf [6] used statistical methods by the oceanographer observation and fast Fourier Transform (FFT) to obtain the results of breaking wave. Jensen and Goliias [7] introduced a new real-time level-of-detail deepwater animation scheme, which uses many different proven water models. Johanson [1] used Perlin noise to simulate the details of ocean, and presented an alternative technique called "projected grid". For the shallow water waves on a height-field, the wave equation in [17] can be used to describe the fluid flow by the 2D NSE and wave effects. In addition, a noise-based animation of detailed fluid structures further improves the realistic appearance.

2.2 Breaking wave

When the weather is stormy or windy, the wave will overturn and break. The breaking wave is very important for reality. The breaking wave includes the spilling breaking, plunging breaking and surging breaking.

If we can simulate the different kinds of breaking waves, the simulation results will become more real. The height field can get the real result and fast simulation, but the disadvantage is that it cannot simulate breaking waves. Existing methods for breaking wave use NSE to calculate accurately the varying of sea to implement breaking waves. Unfortunately, the NSE is a set of highly non-linear partial differential equations (PDE).

There are many methods for discrete the PDE in the literature (e.g. finite difference [9], finite volume [18] and finite element [14]). Stam [5] proposed the semi-Lagrangian approach to solve the advection of NSE and simulate the varying of the 3D fluid. Foster and Fedkiw [10] presented a general method for modeling and animating liquids, and extended the object interaction mechanism to provide control over the liquid's 3D motion. Mihalef et al. [20] developed a novel fluid animation control approach to control breaking waves, and proposed the slice method to define the

shape of a breaking wave at a desired moment by the library evolution of breaking waves. Irving et al. [4] proposed an approach which optimizes the fluid simulations with tall and thin cells, and also reduces the computational complexity for large fluid volumes. Thürey et al. [12] presented a new method for enhancing shallow water simulations by the effect of overturning waves that make it possible to simulate scenes in real-time.

The wave simulation including the splashes and foam has an important role in the visual representation of the breaking waves. The particle system often specifies the splashes and foam movement in 3D environment. Takahashi et al. [19] proposed the control of the generation, vanishing and transition rule of splashes and foam. Based on the particle model Thürey et al. [11] presented a shallow water to achieve real-time by using a smoothed particle hydrodynamics simulation. Wang et al. [13] used moving particle semi-implicit (MPS) method, and using fractional Brownian motion (fBm) 2D simulation to expand into 3D representation by giving motion variation.

Due to simulating the effect of breaking waves needs a large amount of calculation and spends considerable time. Therefore, this paper proposes a slice-based method to speed up the simulation of breaking wave by using a new reconstructing algorithm keeps the simulation of volume fraction of fluid in VOF in 2D. Then it is expanded to 3D representation by combining linear interpolation with noise function. The proposed algorithm is effective in reducing the computation time.

3. THE PROPOSED METHOD

We proposed a slice-based method to get 2D simulation results first and then combined them using a linear interpolation with noise function to generate the new 2D simulation slices from previous two 2D simulations. Because the linear interpolation will cause unnatural result, the noise function is used to make the motion and behavior look more natural. For the 2D simulation we use NSE as the physical equation and solve free surface via VOF. In addition, about the reconstruction of volume fraction in VOF [15] one can improve by a moving interface reconstructing algorithm, and use the project grid method in reducing unnecessary computation in implementing the level of detail (LOD).

The system architecture for the proposed ocean simulation is given in Figure 1. It consists four parts including the grid generation, wave simulation, particle simulation, and render respectively. And the flowchart is given in Figure 2.

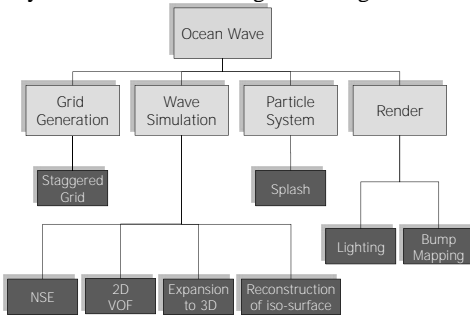


Figure 1 : System architecture

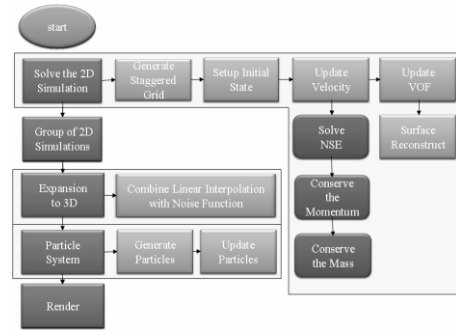


Figure 2 : The flowchart of the proposed method

3.1 Grid generation

We use finite difference schemes that can be described as non-staggered or on staggered grids. In the same positions all the scalar variables for the non-staggered grid are stored, and the staggered grid the scalar variables (e.g. pressure and density) are stored in the cell centers of the control volumes, whereas the velocity or momentum variables are located at the cell faces. Using a staggered grid is a feasible way to avoid odd-even decoupling between the velocity and pressure. Odd-even decoupling is a discretization error that can occur on collocated grids and can lead to checkerboard patterns in the solutions.

3.2 Wave simulation

The main part we concern for the ocean simulation is the variation of water wave. We can use 2D NSE to calculate the fluid varying, and solve the problem of free surface using VOF. However, by combining the linear interpolation with noise function one can generate the new 2D simulation results between any two existing 2D simulations. Finally, we use an isosurface reconstruction approach to obtain the 3D surface.

3.2.1 Fluid equation

The incompressible NS equation is used for the ocean simulation.

The NSE includes three types of forces, they are body forces (F_g), pressure forces (F_p) and viscous forces (F_v). The body forces act on the entire water element. We assume only the gravity, $F_g = \rho G$, where ρ is density and G is the gravitational force (9.81m/s^2). The pressure forces act inwards and normal to the water surface, so the pressure forces are defined as the negative ($F_p = -\nabla \cdot p$). The viscous forces are $F_v = \mu \nabla^2 V$ according to the definition of Newtonian fluid, where $\mu = 1/\rho \nu L$, ρ is density, V is velocity and L is dimension. Newton's second law is used ($F = mA$) to describe the motion. According to the $F = F_g + F_p + F_v = \rho A$, we can get the expansion as $\rho A = \rho G - \nabla \cdot p + \mu \nabla^2 V$. Assume that density is uniform, the acceleration is $A = G - \nabla p / \rho + \mu \nabla^2 V / \rho$. The varying velocity over time is $\partial V / \partial t + (\nabla \cdot V)V = G - (\nabla \cdot p) / \rho + (\mu \nabla^2 V) / \rho$.

This equation conserves the momentum. In addition, the equation $\nabla \cdot V = 0$ describes a conservation of mass. The last two equations together are referred to as the NSE. The NSE contains four terms: force, advect, diffuse, and pressure. The steps of the NSE solver are illustrated in Figure 3. We can solve the equation one by one.

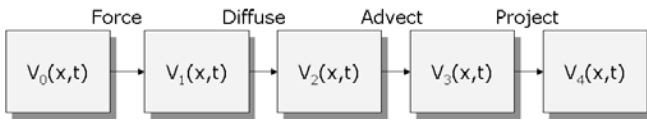


Figure 3 : The steps of the NSE solver

First, the simplest term is the external force f . These forces may be either local forces (wind blow) or body forces (gravity). Here we assume that the force does not vary considerably during the time step. So, we can get the equation as $V_1(t+\Delta t)=V_0(t)+f \cdot \Delta t$. The next step is the diffuse term. It is for the effect of viscosity and is equivalent to a diffusion equation. Viscosity is measure of how resistive a fluid is to flow. We can get the equation by the finite difference method as $V_2(t)=V_1(t)+(\Delta t \cdot \nu \nabla^2 V)/\rho$. The third step solves for the effect of advection (or convection) of the fluid on itself. The advection term is non-linear and cannot be approximated by the finite difference method. Here, we use the method of characteristics to solve partial differential equations. All the fluid particles are moved by the velocity of the fluid itself at each time step. It is necessary to calculate the velocity at a point x at the new time $t+\Delta t$. So, we backtrace the point x through the velocity field U_2 over a time Δt . This defines a path $p(x,s)$ corresponding to a partial streamline of the velocity field. The new velocity at the point x is set to the velocity at its previous location a time Δt ago. The equation is $V_3(t)=V_2(p(x,-\Delta t))$. The final step is project term which is used to solve the pressure term. According to Helmholtz-Hodge decomposition theorem, any vector can be decomposed into a set of basis vector components. So, we can get the equation $w=u+\nabla p$. The new velocity field w with nonzero divergence to solve the NSE that involves three computations to update the velocity at each time step : force, diffusion, advection application. The velocity field u is free divergence velocity, and p is the pressure field. If we apply the divergence operator to both sides, then we obtain $\nabla w=\nabla \cdot (u+\nabla p)$. Because the velocity field u is free divergence ($\nabla \cdot u=0$), the equation can be simplified to $\nabla \cdot w=\nabla^2 p$. The equation is a Poisson equation for the pressure of the fluid. We can solve the equation for p , and then use w and p to compute the new divergence-free field u . So, the velocity equation of the final step is $V_4(t)=V_3(t)-\nabla p/\rho$.

3.2.2 Free surface

The free surface is the most influential factor for reality. So the classification of the free surface grid is used to distinguish the location of free surface. It is called full cell when the grid contains only liquid. The grid is located at the top of the free surface and the grid that contains no liquid is called empty cell. In addition, the grid that contains liquid and gas, and where at least one neighboring grid is empty is called the surface cell. We should determine whether the grid belongs to the above cells before the surface reconstruction. If the grid is a full cell, one can calculate the velocity according to the internal fluid field. On the other hand, it does not need to calculate when the grid is an empty cell. Besides, it should get the average density when the grid is surface cell, then calculate density by internal fluid field equation, $\rho=\rho_1+(\rho_2-\rho_1)f$, where ρ_1 and ρ_2 are the gas and liquid density respectively, and f is the fraction of fluid in the grid. We can get the free surface through the above description but the premise is that the grid type has already been determined. Therefore we have to determine the grid type using VOF, which defines an additional variable in the computing grid called

fraction-of fluid function, $f(x,z,t)$. The value of f is between zero and one. The value “0” expresses an empty grid, and we defined “1” as a full grid and “0~1” as a surface grid. Because the free surface moves over time, the value f will change with the free surface. The VOF gets the free surface through the definition of f for each grid, and the f in each cell is transported by the advection equation as $f_t+u \cdot \nabla f=0$.

We reconstruct the free surface according to a moving interface reconstructing algorithm [15], but the reconstructing algorithm is only implemented on the closed boundary grid. Hence, we modify the algorithm for the ocean simulation (non-closed boundary). The grids whose value is between zero and one are first located. The intersection of the surface of the boundary line and the grid is called spot. The surface grids shown in Figure 4 have three types. Each type includes four patterns according to the relationship of the location of the surface grid and the adjacent grid.

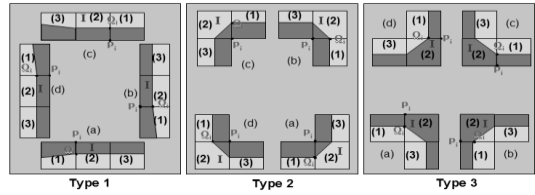


Figure 4 : Three types for the free surface grid

The grid (1)(2)(3) is the order index for surface grid as shown in Figure 4. The grid I is the current boundary grid. Type 1 is characterized by “linear” that the current grid and the adjacent grid is a straight line parallel to the x axis or y axis. Type 2 is characterized by “convex” that the current grid and the adjacent grid are not straight but convex. Type 3 is characterized by “concave” that the current grid and the adjacent grid are not straight but concave. We define four different cases according to the different types. In order to obtain the spot, we define Q_i as the boundary spot. P is the grid node that locates at the adjacency of Q_i . The variable t_i is the reference value and is defined as follows.

$$t_i = \begin{cases} |PQ_i|/\Delta x & , \text{if } PQ_i \text{ is parallel to the } x \text{ axis} \\ |PQ_i|/\Delta y & , \text{if } PQ_i \text{ is parallel to the } y \text{ axis} \end{cases} \quad (1)$$

where Δx and Δy are the length of the grid x axis and y axis respectively. We record the related information of each Q_i , including the number of spot, type and the coordinate of $P_i(X_{P_i}, Y_{P_i})$. Assuming the volume fraction of fluid of the I surface grid is f_i , we can build volume fraction of fluid as listed in Table 1 according to the type of the surface grid. After obtaining t_i one can calculate the coordinate of spot (X_{Q_i}, Y_{Q_i}) according to the spot Q_i as shown in Table 2.

Table 1 : Volume fraction of fluid

The i grid type	Keeping volume fraction of fluid
Type 1 : Linear	$\frac{1}{2}(t_i+t_{i+1})=f_i$
Type 2 : Convex	$\frac{1}{2}t_i t_{i+1}=f_i$
Type 3 : Concave	$1-\frac{1}{2}(1-t_i)(1-t_{i+1})=f_i$

Table 2 : Formula of spot Q

The pattern of the I type	Spot Q	
	X_{Q_i}	Y_{Q_i}
(a)	X_{P_i}	$Y_{P_i} - t_i$
(b)	$X_{P_i} + t_i$	Y_{P_i}
(c)	X_{P_i}	$Y_{P_i} + t_i$
(d)	$X_{P_i} - t_i$	Y_{P_i}

Assuming the boundary grid, \dot{i} , is the volume fraction of fluid f_i . If the boundary grid is closed and the boundary is composed of the boundary grids, we can obtain N equations. By solving these equations one can obtain $t_i (i = 1, 2, \dots, N)$ and the coordinate of Q_i by $t_{N+1} = t_1$. When the simulation has non-closed boundary, the equation would be $t_{N+1} \neq t_1$. Hence, we should obtain the spot of start grid before the operation. Here, we divide the start grid into three types and seven cases. And we can obtain the t_i as shown in Figure 5 by the equations of the different cases. We divide the start grid into three types. Type 1 is "leftward", and the adjacent grid is located on the left start grid. Type 2 is located below the start grid, called "downward". Type 3 is located above the start grid, called "upward". The different types have different cases. According to the different cases one can calculate t_i as shown in Table 3.

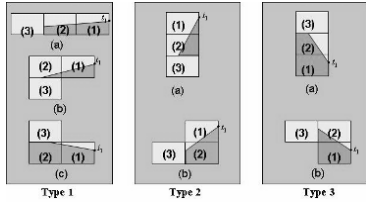


Figure 5 : The status of the start grid

 Table 3 : Formula of t_1

Type 1	
(a)	$t_1 = (3f_1 - f_2) / 2$
(b)	$t_1 = 2(f_1 + f_2) \pm 2\sqrt{f_2^2 + f_1 f_2}$
(c)	$t_1 = (2f_1 + 2f_2 - 3) + 2\sqrt{f_2^2 + f_1 f_2 - f_1 - 3f_2 + 2}$
Type 2	
(a)	$t_1 = (f_1 + \sqrt{f_1^2 + f_1 f_2}) / f_2$
(b)	$t_1 = 2f_1 \pm 2\sqrt{f_1 - f_1 f_2}$
Type 3	
(a)	$t_1 = ((f_1 - f_2) \pm \sqrt{f_1^2 + f_1 f_2 - 3f_1 - f_2 + 2}) / (1 - f_2)$
(b)	$t_1 = (2f_1 - 1) \pm 2\sqrt{f_2 - f_1 f_2}$

3.2.3 Expansion to 3D

The slice method can generate the different 2D simulations, one can further integrate them to get a 3D shape. We can use the level of detail (LOD) scheme because LOD involves decreasing the complexity of a 3D object representation as it moves away from the viewer or according other metrics such as object importance, eye-space speed or position. Based on the LOD each patch will be determined by the distance to the viewer, one can use the larger grid on the 2D simulation when grid is farther away from the camera.

In contrast to the slice method that needs to process many slices, the proposed method only process some key slices. We apply linear interpolation with noise function to generate the new 2D

simulations between two key slices. The formula is shown as follows.

$$P_{i+j} = P_{i,j} + \frac{(P_{k,j} - P_{i,j})}{k-i} t, 1 \leq j \leq n \quad (2)$$

$$P'_{i,j} = P_{i,j-1} + (P_{i,j} - P_{i,j-1}) \text{Noise}(j) \quad (3)$$

where $P_{i,j}$ is the i -th of 2D simulation result (outline), and the j is the order of the boundary. $P_{i,j}$ and $P_{k,j}$ are the two 2D simulations by VOF, and they need to interpolate 2D new slice P_{i+j} between the i -th and the k -th simulation. By Eq.(2) which is the linear interpolation one can obtain the value of the interpolation. But the quality is insufficient, when only use the value of the interpolation to generate the 2D simulation, so a noise function is integrated into Eq.(3). The noise function makes the simulation more real.

The reconstructed surface may have aliasing and not smooth. Therefore, we should add some techniques to make the surface more smooth. Here, we use the Chaikin's algorithm to smooth the reconstructed surface.

3.3 Particle simulation

The particle system is used to simulate certain fuzzy phenomena on computer graphics. The particle status of splashes and foam is very important for the ocean simulation. In this paper, we only render the splash. We generate the particles according to the value of curvature, and the new splash particle has its lifespan. Then we update position over time. And the splash particles are rendered in white. In this way, we can effectively visualize splashes.

3.4 Render

In this step, we use the lighting and texture operations for the ocean simulation. The texture technique that we used is bump mapping. Bump mapping perturbs the surface normal of the object at each pixel, and is applied before the illumination calculation. In addition, it has been proved an effective way to improve the appearance of reality and not increase the complexity of the geometric objects. The technique has already implemented in the field of real-time graphics with the graphics hardware progresses and the application of shading language. In order to render the water, we should implement the reflective and the refractive effects. Environment mapping can help us to render the effect. One can calculate the normal vector to get the corresponding environment texture. Then one can mix the texture with water color and get the final color. The technique is easy to implement by GLSL, and the result is more realism.

4. IMPLEMENTATION

The simulation of breaking wave has a lot of formulas that are very complicated to implement for hydrodynamics. In this paper, we use the finite difference method to describe the formula of NES. Then, we can get the velocity at the center of each face of a grid and the pressure at the center of a grid. According to the correctness of velocity and pressure one can construct a flow field and do the ocean simulation.

4.1 2D simulations

4.1.1 Initialization

Before the simulation, the computation domain can be represented via a fixed rectangular grid aligned with a Cartesian coordinate system. To initial configurations of space such as mesh size, velocity field and pressure field, and prescribe initial conditions for the 2D NSE, one can create an ocean wave by [20]. At first, we denote the period, T , wavelength, λ , and amplitude, A . If

the frequency, $w = 2\pi/\lambda$, and the phase-constant $\varphi = 2\pi/T$ which expresses speed, the surface displacement for a wave is given as $W(x,t) = A\cos(wx - \varphi t)$. And the velocity components are given by

$$\begin{aligned} u(x,t) &= A\varphi e^{-wz} \cos(wx - \varphi t) \\ v(x,t) &= A\varphi e^{-wz} \sin(wx - \varphi t) \end{aligned} \quad (4)$$

where z is the depth, and we can obtain the equation when time is zero.

$$W(x,0) = d + \frac{1}{2\pi} \left(wA\cos(2\pi x) + \frac{1}{2}(wA)^2 \cos(2\pi x) + \frac{3}{8}(wA)^3 \cos(6\pi x) \right) \quad (5)$$

4.1.2 Discretization

We expand the equation via the finite difference method because the 2D simulation is driven by NSE, The NSE contains three different applications of the symbol ∇ . The three applications are gradient, the divergence, and the Laplacian operators respectively, as shown in Table 4.

Table 4 : Three finite difference forms

Operator	Definition	Finite Difference Form
Gradient	$\nabla p = \left(\frac{\partial p}{\partial x}, \frac{\partial p}{\partial y} \right)$	$\left(\frac{p_{i+1,j} - p_{i-1,j}}{2\Delta x}, \frac{p_{i,j+1} - p_{i,j-1}}{2\Delta y} \right)$
Divergence	$\nabla \cdot v = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$	$\frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta y}$
Laplacian	$\nabla^2 p = \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2}$	$\frac{p_{i+1,j} - 2p_{i,j} + p_{i-1,j}}{(\Delta x)^2} + \frac{p_{i,j+1} - 2p_{i,j} + p_{i,j-1}}{(\Delta y)^2}$

The subscripts i and j used in the expressions in Table 4 is used to discrete locations on a Cartesian grid, Δx and Δy are the size of the grid in the x and y dimensions, respectively, and $u_{i+1/2,j}, v_{i,j+1/2}$ are the location of staggered velocity components on a grid. This discretization leads to an explicit finite difference approximation of Eq.(3).

4.1.3 Boundary conditions and surface grid

Boundary condition is set on the boundary between the obstacle and the water. Assume the obstacle is non-slip, one can make the gradient of pressure equal zero on the boundary and the velocity on the boundary is zero.

The pressure of the empty and surface grid are set to atmospheric pressure, and the velocity on each of its faces shared with another empty grid, which is set to zero and the velocity should satisfy the conservation of mass. As shown in Figure 6, Type 1 shows the velocity on four sides of the two dimensional surface grid. Three of them are surrounded three sides by grids containing fluid. In order to keep the conservation of mass, the velocity of air v_2 set to $v_2 = v_1 - (u_2 - u_1)$. When the grid has two sides which face an empty grid, it is type 2. Each open side velocity equals the velocity of the side of the grid opposite it. And when the grid has three open sides, it is type 3. The side velocity equals the velocity of the side that opposites it

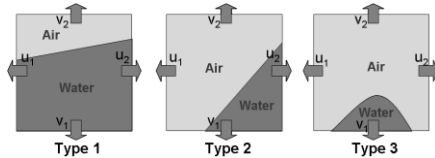


Figure 6 : Setting the velocity on the free surface

4.2 GPU implementation

The efficiency is lower when only using CPU, because of the complex operations for NSE. General-purpose computing on graphics processing units (GPGPU) is the technique of using a GPU, which typically handles computation only for computer graphics, to perform computation in applications traditionally

handled by the CPU.

We utilize GPU to improve for the complicated operations. In the GPU pipeline, the traditional end point of every rendering operation is the frame buffer, a special chunk of graphics memory from which the image that appears on the display is read. The frame buffer can get is 32 bits of color depth, shared among the red, green, blue and alpha channels. The data will always be clamped to the range of [0;1]. Fortunately, OpenGL extension provides 32-bit floating point values on GPUs. Each RGBA channel of offscreen buffer supports 32-bit floating point values for vector calculations. We commonly call the technique FBO.

Arrays in GPU memory are called textures or texture samplers. Texture dimensions are limited on GPU, the maximum value in each dimension is 2048 or 4096. The data elements which are accessed from texture memory, one needs to choose a special projection from the 3D world to the 2D screen. Pixels and texels are one to one mapping. To choose an orthogonal projection and a proper viewport, one will enable a one to one mapping between geometry coordinates and texture coordinates and pixel coordinates. To use a texture as render target to attach the texture to the FBO, the FBO extension provides a very narrow interface to render to a texture. Then we transfer data to a texture from CPU to GPU. The operations that are calculated on GPU can be achieved by shading language. The shading language divides into two parts(vertex and fragment shader), and we can code according to the requirement and transfer from GPU textures to CPU arrays or render.

GPGPU can help us to do the complicated arithmetic, but one thing should be noted. It is expensive to transfer data from GPU to CPU. So, we must consider very carefully before doing GPGPU. We calculate the two dimensional NSE solver on CPU and GPU respectively. Table 5 lists the comparison of computation time between CPU and GPU. Here we divide the GPU process into three major items, the NSE fragment, texture transmission and glReadPixels function. GPU spends most of time on glReadPixels function. The computation time on CPU is about 15.9 milliseconds and GPU is about 2 milliseconds. The result is calculated on the grid size 100×100 . It shows the excellence floating-point operation of GPU.

Table 5 : Computation time between CPU and GPU

Operation	Run NSE fragment	Texture transmission	glReadPixels
GPU	Less than 0.5ms	0.25ms	0.7ms
CPU	15.9ms		

4.3 Results

We generate a 2D initial wave and the parameters $\varphi=2$ and $w=3$. The grid size is 100×100 . Figure 7 shows the 2D wave. The left picture is at time 2.8 and the right picture is at time 9.5. The experiments were run in PC with 3.4GHz Pentium IV CPU and NVIDIA GeForce 8800GTS display card. The performance is about 44.15 frames per second.

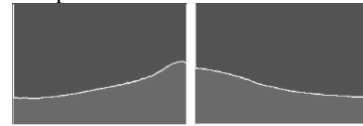


Figure 7 : Snapshots of the 2D wave

The 3D simulation result is through combining with the different 2D results of the proposed algorithm. Figure 8 shows the result. The advantage of the proposed algorithm is that users can choose

the 2D results to determine the variation of wave. Figure 9 shows the FPS measurements for the different slice counts. These slices would be uniformly distributed on the interval of z-axis grid size. And the grid size is $100 \times 100 \times 50$ and the number of triangles is about 148000.

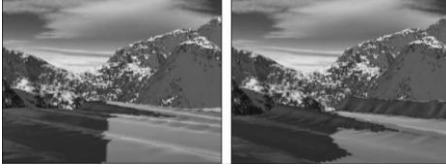


Figure 8 : Snapshots of the 3D wave

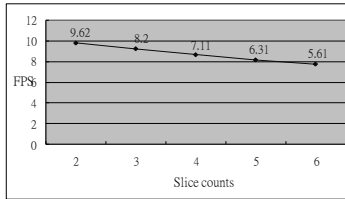


Figure 9 : FPS measurements for the different slice counts

NSE can be divided into four steps with the time complexity is listed in Table 6. Project step uses the PCG method to reduce both the space complexity and time complexity per iteration from $O(n^2)$ to $O(m)$, where n is the grid size, m is the number of nonzero entries of A , and κ is the iteration times. Our algorithm efficiency is competitive compared with the previous method of the slice method [20]. In [20], the free surface is calculated by CLSVOF. The result is more realism but needs more computational time. In the proposed algorithm, we use VOF to solve the free surface. The VOF method is known for its ability to conserve the "mass" of the traced fluid. Although the reality is less, the computational time is twice less than that of CLSVOF.

Table 7 lists comparisons of time complexity for the proposed and the slice methods. We assume the time complexity of 2D NSE and the surface reconstruction is $O(K)$, so the time complexity of the proposed method and the slice method is $tO(K)$ and $hO(K)$ respectively, where t is the number of key slice and h is the length of z axis ($t \leq h$). Here we assume the length of z -axis on 3D space is fifty. The slice method needs fifty slices and the proposed method would use 2~6 slices to expand 3D environment. The computation time of the proposed method is faster than the slice method. Although fewer slices would reduce the reality, the work efficiency would increase. And the proposed method is more flexible due to users can decide to control slices count. In order to reduce aliasing which producing by linear interpolation, we add the noise function to make the free surface more natural.

Table 6 : Time complexity for different solutions of NSE

NSE	Force	Diffuse	Advect	Project
Method	Finite difference	Finite difference	Method of characteristics	PCG
Time complexity	$O(n)$	$O(n)$	$O(n)$	$O(\kappa m)$

Table 7 : Time complexity of the proposed method and the slice method

Method	The proposed method	The slice method
Time complexity	$tO(K)$	$hO(K)$
Computation time	5~9 frames/s	1 frame/s

We render the environment mapping, the reflective and the refractive effect of the water by GLSL. The reflective and the refractive effect of the water are mix environment texture with water color. And we use bump mapping to increase the reality. The technique is simple and yields good results. Figure 10 shows the render effect. And Figure 11 increases the splash effect by the particle system.

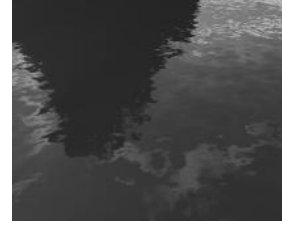


Figure 10 : Render effect

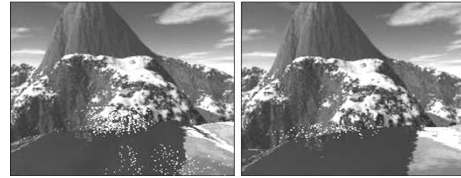


Figure 11 : Snapshots of the breaking wave

5. CONCLUSIONS

In this paper, we have presented a slice-based water simulation method for ocean breaking wave on natural simulation. By combining linear interpolation with noise function to construct a complete three dimensional ocean simulation, one can reduce computational time and achieve better efficiency. The proposed algorithm can improve the computational time of the slice method. And we also avoid the aliasing using linear interpolation with noise function. The proposed method is more flexible to users, because users can determine how many slices are used. Users can generate simple ocean waves by a few slices, or complicated ocean waves by more slices. No matter what users choose, the ocean wave is still realism.

People can design the different ocean waves by the slice method, but it would spend much time and resources. The proposed method can speed up the computation time as fast as 9 times and make the simulation efficiency. And this method still can reserve the reality because it uses the noise function to increase the variation of water.

6. REFERENCES

- [1]. C. Johanson, "Real-time water rendering," *Master of Science paper*, Lund University, 2004.
- [2]. D. Enright, S. Marschner, and R. Fedkiw, "Animation and Rendering of Complex Water Surfaces," *Proc. of the ACM Transactions on Graphics*, vol. 21, no. 3, pp. 736–744, 2002.
- [3]. D. Kim, O.Y. Song, and H.S. Ko, "A Semi-Lagrangian CIP Fluid Solver without Dimensional Splitting," *EUROGRAPHICS 2008*, vol. 27, no. 2, 2008.
- [4]. G. Irving, E. Guendelman, F. Losasso, and R. Fedkiw, "Efficient Simulation of Large Bodies of Water by Coupling Two and Three Dimensional Techniques," *Proc. of the ACM Transactions on Graphics*, vol. 25, 2006.
- [5]. J. Stam, "Stable Fluids," *Proc. of ACM SIGGRAPH '99*, pp. 121–128, 1999.
- [6]. J. Tessendorf, "Simulating Ocean Water," *Proc. of ACM SIGGRAPH 2002 Course Notes 9 (Simulating Nature: Realistic and Interactive Techniques)*, 2002.

- [7]. L.S. Jensen, and R. Goliás, “Deep-Water Animation and Rendering,” *In Gamasutra*, 2001.
- [8]. M. Kass, and G. Miller, “Rapid, Stable Fluid Dynamics for Computer Graphics,” *Proc. of the ACM Transactions on Graphics*, vol. 24, no. 4, pp. 49–55, 1990.
- [9]. N. Foster, and D. Metaxas, “Modeling the motion of a hot, turbulent gas,” *In Proc. of SIGGRAPH '97*, pp. 181–188, 1997.
- [10]. N. Foster, and R. Fedkiw, “Practical animation of liquids,” *Proc. of ACM SIGGRAPH*, pp. 23–30, 2001.
- [11]. N. Thürey, F. Sadlo, S. Schirm, M. Müller-Fischer, and M. Gross, “Real-time Simulations of Bubbles and Foam within a Shallow Water Framework,” *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2007.
- [12]. N. Thürey, M. Müller-Fischer, S. Schirm, M. Gross, and ETH. Zurich, „Real-time Breaking Waves for Shallow Water Simulations,” *15th Pacific Conference on Computer Graphics and Applications*, 2007.
- [13]. Q. Wang, Y. Zheng, C. Chen, T. Fujimoto, and N. Chiba, “Efficient rendering of breaking waves using mps method,” *Journal of Zhejiang University SCIENCE A*, 2006.
- [14]. R. LeVeque, “Finite volume methods for hyperbolic problems.” *Cambridge Texts in Applied Mathematics*, Cambridge University Press, Cambridge, 2002.
- [15]. R. Wang, P. Chen, and C. Ban, “Keeping Volume Fraction of Fluid in Reconstructing Moving-interfaces of VOF on Rectangular Meshes,” *Chinese Journal of Computational Physics*, vol. 25, no. 4, 2008.
- [16]. S. Jeschke, H. Birkholz, and H. Schmann, “A procedural model for interactive animation of breaking ocean waves,” *Proc. WSCG2003 POSTERS*, 2003.
- [17]. T. Klein, M. Eissele, D. Weiskopf, and T. Ertl, “Simulation, modelling and rendering of incompressible fluids in real time,” *Workshop on Vision, Modelling, and Visualization VMV '03* pp. 365-373, 2003.
- [18]. T. Nakayama, and M. Mori, “An Eulerian finite element method for time dependent free surface problems in hydrodynamics,” *Int. j. numer. methods fluids*, vol. 22, pp. 175–194, 1996.
- [19]. T. Takahashi, H. Fujii, A. Kunimatsu, K. Hiwada, T. Saito, K. Tanaka, and H. Ueki, “Realistic animation of fluid with splash and foam,” *Computer Graphics Forum*, vol. 22, no. 3, 2003.
- [20]. V. Mihalef, D. Metaxas, and M. Sussman, “Animation and Control of Breaking Waves,” *Proc. of the ACM SIGGRAPH/ Eurographics Symposium on Computer Animation*, pp. 315–324, 2004.

About the author

Chung-Nan Lee is a professor at National Sun Yat-sen University, Department of Computer Science and Engineering. His contact email is cnlee@cse.nsysu.edu.tw.

Jyun-Ming Wu is a Master student at National Sun Yat-sen University, Department of Computer Science and Engineering. His contact email is piiwu2002@yahoo.com.tw.

Yan-Zheng Huang is a Master student at National Sun Yat-sen University, Department of Computer Science and Engineering. His contact email is lewis005299@hotmail.com

Da-Jing Zhang-Jian is a Ph.D. student at National Sun Yat-sen University, Department of Computer Science and Engineering. His contact email is salmoner.tw@yahoo.com.tw.

Animated thumbnail for still image

Iliia V. Safonov, Victor V. Bucha

Samsung Research Center,

Moscow, Russia

ilia . safonov at samsung . com, v . bucha at samsung . com

Abstract

The conventional thumbnails do not provide a user-friendly way to view and browse still images on portable devices with a small display. The issue is that a still thumbnail is too small to recognize image details such as human faces, document title and photo quality.

To cope with this problem we propose an automatic algorithm for creation of animated thumbnail from still photos and scanned documents. The approach comprises image classification on photo and scanned document, detection of attention zones, creation of animation that simulates camera tracking-in, tracking-out, panning between detected zones and whole scene.

Short looped movie demonstrates whole image, main objects of the scene one by one as well as quality of the photo. User study demonstrates effectiveness of proposed animated thumbnail in comparison with conventional still thumbnail.

Keywords: *animated thumbnail, creation video from still image, vision model, attention zones, document segmentation*

1. INTRODUCTION

The thumbnails are used widely to browse image collection. It is essential part of the user interface for the various devices, PC and Web-applications. In general, thumbnail represents down-sampled copy of original image. However, often it is difficult to recognize original image from observed thumbnail. The small details and overall image quality are undistinguished especially when a size of original image is big enough or has a complex layout (Fig. 1).



Figure 1: Conventional thumbnails on various devices.

For example, the correlation between the image of a document and its thumbnail is not obvious. To browse photo collection with the small thumbnails is not so easy too. Often user is forced to make zoom-in in order to recognize the people on it.

In the paper we propose the approach for demonstration of still image thumbnail as an animation comprising the frames with the most important zones of the image. The resulting animation looks attractive and provides a user-friendly way to view and browse images especially on small displays widely used in the mobile devices. The animated thumbnail may look like a slide-show application for full-size photos. However a thumbnail animation for document images has no well-known examples.

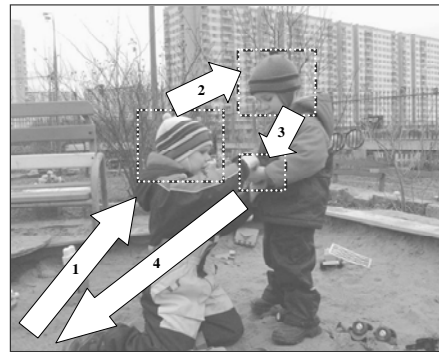


Figure 2: Frames of animated thumbnail for photo.

2. RELATED WORK

There are several approaches to create “intellectual” thumbnail as another still image. Way for creation of single still thumbnail by means of auto cropping for photo retrieval is described in [1]. The approach is based on the face detection and saliency map building. The technique has several drawbacks: the overall initial view is destroyed; it is not clear how thumbnail is created for photos with several faces as well as for photos with several salient regions; aspect ratio of images is altered.

Another automatic thumbnails creation process is discussed in [2]. The proposed approach does not modify the image composition and rather better reflects the image quality of the originals. The

main purpose of this approach is to reflect noise level and blurriness of photo on a generated thumbnail.

So called SmartNails for browsing of document images is described in [3]. SmartNail consists of a selection of cropped and scaled document segments that are recomposed in order to fit the available display space while maintaining recognizability of document image, readability of text and keeping the layout close to the original document layout. Evidently, the recognizability of SmartNail is high. Nevertheless the overall initial view is destroyed as well as sometimes layout alteration is estimated by observer ambiguously.

3. ANIMATED THUMBNAIL

3.1 General idea

In order to recognize image content as well as to estimate its quality confidently it is necessary to see zoomed-in fragments of the image. Sometimes several regions of the image should be inspected. We propose to create smooth animated transitions between *attention zones* of image. Obtained video frames are cropped from initial still image and scaled to thumbnail or display size. Further the frames are stored in appropriated animation file format (animated GIF, Adobe flash) or played on-the-fly.

Duration of the movie should not be too long. Optimal duration is less than 10 seconds. Therefore, the number of attention zones is limited to 3-5. Animation can be looped. We named such small size movie an *animated thumbnail*.

The algorithm for animated thumbnail generation has the following three key stages:

- detection of attention zones;
- selection of zone for quality estimation;
- generation of video frames which are transitions between the zones and whole image.

The attention zones are important for recognition of the image and they differ depending on image content. At least two types of still images can be selected: photos and document images. Title, headers, embedded pictures and other emphasized text elements are sufficient for identification of a document image. Human faces are adequate for identification of photos for the most part. For photos which do not contain faces the preattentive human vision model can be used for detection of visual attention zones.

There are plenty of methods for detection of faces and other visual attention zones as well as methods for document layout analysis. Below we describe one possible way for important zones detection without comparison with any alternatives. We demonstrate advantages of animated thumbnail as a concept.

It should be clear, that to select approaches for important zones detection one should take into account application scenario and hardware platform for implementation. Fortunately panning over image during animation allows to recognize image content even if important zones were detected incorrectly. So we prefer more simple techniques for attention zone selection rather than more comprehensive ones.

For visual estimation of blurriness, noise and compression artifacts observer should investigate fragment of image without any scaling or with modest magnification. We propose several simple rules for selection of appropriate fragment: the fragment should contain at least one contrasting edge and at least one flat region, histogram of the fragment should be wide enough but without clipping on limits of dynamic range. Proposed algorithm

uses these rules for automatic zone selection in the central part of an image or in the attention zones.

Fig. 2 demonstrates example of animated thumbnail for still photo. Two faces are detected on the first stage. Hands of kids are selected as a zone for quality estimation. Movie consists of four transitions between whole image and these three zones. The first sequence of the frames looks like a camera tracking-in to a face; after that frame is frozen on a moment for better observer's focus on zoomed face. The second sequence of the frames looks like a camera panning between faces. The third sequence of the frames looks like a camera panning and zooming-in between face and hands; after that the frame with the hands is frozen on a moment for visual quality estimation. Final sequence of frames looks like a camera tracking-out to whole scene and freeze frame takes place again.

3.2 Image classification

Every image type has its own approach to detect areas of attention. That is why the first processing step is image classification into two types: photo or document. The classification is performed by analyzing of average *energy* En of normalized co-occurrence matrices of *rgb* channels of downsampled image:

$$C_r(i, j) = C_r(i, j) + 1 \text{ if } r(x, y) = i \text{ and } r(x + dx, y + dy) = j, \\ \forall x, y$$

$$C_g(i, j) = C_g(i, j) + 1 \text{ if } g(x, y) = i \text{ and } g(x + dx, y + dy) = j, \\ \forall x, y$$

$$C_b(i, j) = C_b(i, j) + 1 \text{ if } b(x, y) = i \text{ and } b(x + dx, y + dy) = j, \\ \forall x, y$$

$$N_r(i, j) = \frac{C_r(i, j)}{\sum_i \sum_j C_r(i, j)}; \quad N_g(i, j) = \frac{C_g(i, j)}{\sum_i \sum_j C_g(i, j)};$$

$$N_b(i, j) = \frac{C_b(i, j)}{\sum_i \sum_j C_b(i, j)};$$

$$En = \frac{1}{3} \sum_i \sum_j (N_r^2(i, j) + N_g^2(i, j) + N_b^2(i, j));$$







where x, y – coordinates of pixels of image, dx, dy – displacements.

Table 1 demonstrates energy En for various images. Average energy of normalized co-occurrence matrices En differs for photos and document images on several orders. For photos En is less than 0.01, whereas for typical document images En is greater than 0.1.

Given approach works well for both contone and halftoned photos as well as for color and grayscale photos. Application of three co-occurrence matrices for all *rgb* channels allows to detect document images with a color background unlike the application of one co-occurrence matrix of gray channel.

Naturally, documents which contain huge built-in photo or several photographic illustrations are classified as a photo. The reason of such misclassification is that the relative area of the text is small. On the other hand black and white graphic arts and business graphics are classified as an image document. Despite the misclassification detected attention zones looks pretty good in general.

TABLE 1 NORMALIZED CO-OCCURRENCE MATRICES OF RGB CHANNELS FOR PHOTOS AND SCANNED DOCUMENT IMAGES

Photos	En	Document Images	En
	0.0009		0.3
	0.0001		0.1
	0.0005		0.2

3.3 Attention zones detection for photo

Information about humans on photo is important to recognize the scene. Thus, it is reasonable to apply the face detection algorithm to detect attention zones in the photo. There are many methods for face detection. For example, well-known OpenCV software library contains implementation of face detection for front and profile faces. The implementation is based on Viola-Jones face detector [4]. In general the methods which are based on state-of-the-art Viola-Jones face detector provide good results. However they can't detect rotated faces confidently and find out a lot of false positives. The number of false positives can be decreased with additional skin tone segmentation and processing of downsampled image [6]. In recent years, methods for multi-view face detection were proposed, for example, such an approach is described in [5]. Fig. 3 illustrates detected faces as attention zones.

Very often face may characterize the photo very well but a lot of photos do not contain face or faces may be non-informative as they relate to random person. Thus an additional mechanism has to be used to detect zones of attention. We propose an algorithm to detect the zones of attention based on human visual model of attention. Until now, universal model of human vision does not exist, but pre-attentive vision model based on feature integration theory is well-known [7], [8].

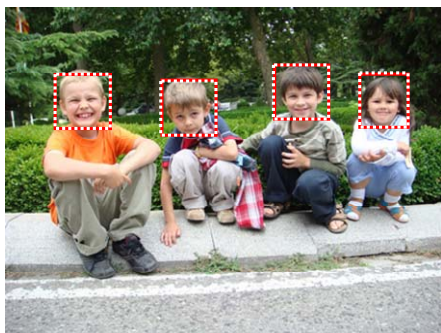


Figure 3: Example of detected zones for photo with people.

In the strict sense a model of human preattentive vision does not quite fit in this case, since the observer is on attentive stage while viewing thumbnail. However existing approaches for the detection of regions of interest are based on saliency map and they often provide reasonable outcomes, whereas the use of attentive vision model requires too much *prior* information about the scene and it is not generally applicable.

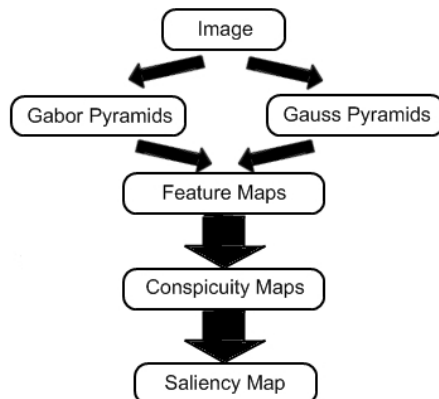


Figure 4: The schema of saliency map building.

We build the saliency map and detect attention zones as described below. The schema of saliency map generation is shown on Fig. 4. Let us define the intensity map I as:

$$I = (r + g + b) / 3.$$

Then four color channels R , G , B , Y are created from r , g , b in the following way:

$$R = r - \frac{g+b}{2}, G = g - \frac{r+b}{2}, B = b - \frac{r+g}{2}, Y = \frac{r+g}{2} - \frac{|r-g|}{2} - b.$$

For I , R , G , Y 8-level Gaussian pyramids are constructed using Gauss separable filter.

From intensity map 8-level Gabor pyramids for different orientations $\theta \in \{0, 45, 90, 135\}$ are created to obtain local orientation information. We compute 42 feature maps using center-surround difference:

$$\begin{aligned} I(c, s) &= |I(c) - I(s)| \\ RG(c, s) &= |(R(c) - G(c)) - (G(s) - R(s))| \\ BY(c, s) &= |(B(c) - Y(c)) - (Y(s) - B(s))| \\ O(c, s, \theta) &= |O(c, \theta) - O(s, \theta)| \end{aligned}$$

where $c \in \{2, 3, 4\}$ and $s = c + \delta$, $\delta \in \{2, 3\}$.

All feature maps are normalized using local maximum technique and combined into conspicuity maps using across-scale addition:

$$\begin{aligned} \bar{I} &= \sum_{c=2}^4 \sum_{s=c+3}^{c+4} N(I(c, s)), \\ \bar{C} &= \sum_{c=2}^4 \sum_{s=c+3}^{c+4} [N(RG(c, s)) + N(BY(c, s))], \\ \bar{O} &= \sum_{\theta \in \{0, 45, 90, 135\}} N \left(\sum_{c=2}^4 \sum_{s=c+3}^{c+4} N(O(c, s, \theta)) \right), \end{aligned}$$

where $N()$ is a normalization operator.

Normalization operator consists of two parts. In the first part Gaussian filter is applied to the image in order to decrease noise. In the second part average local maximum is computed and the whole image is multiplied by the difference of the maximum value on the image and local maximum value. This operation helps to prevent strong but the individual peaks and also helps not to take into account such things as the very bright background.

Conspicuity maps \bar{I} for intensity, \bar{C} for color, \bar{O} for orientation are summed with specific weights into final image which is called saliency map S :

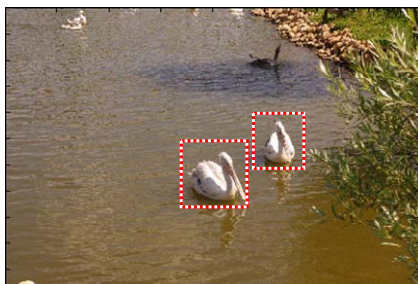
$$S = \frac{wI \cdot N(\bar{I}) + wC \cdot N(\bar{C}) + wO \cdot N(\bar{O})}{wI + wC + wO}$$

The main problem is to find weights wI , wC , wO since due to normalization, different conspicuity maps have different contribution to final result. To solve this problem we applied training by expert procedure. For each photo in the training set the attention zones were marked by several experts. In order to determine the best weights we found maximum of the following function using simplex algorithm:

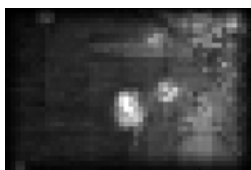
$$\sum_{p \in ROI} S(p) \rightarrow \max.$$

where ROI are marked zones on the image.

Mathematical expectations of weights were calculated after finding values for all images in the training set. The most salient regions correspond to attention zone. Fig. 5 demonstrates two detected zones as well as corresponding conspicuity and saliency maps.



A



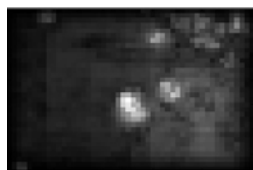
b



c



d



e

Figure 5: Detected attention zones on initial photo (a) and corresponding saliency map (b), intensity map (c), color map (d), orientation map (e).

For time optimization we work with down-sampled versions of original photos during all steps of the algorithm. Generation of saliency map works less than 0.2 second on PC with quad-core CPU 2.4 GHz for color image with size 500x500 pixels. Processing time can be decreased considerably with the fast approach for saliency map generation via quaternion transform [9] and parallel calculation on GPU [10].

3.4 Attention zones detection for document images

Most of the thumbnail images of documents look the same. It is difficult to distinguish from one another. To recognize the document it is important to see title, headers and embedded pictures. There are a lot of document layout analysis methods which allow to perform the segmentation and detection of different important regions of the document [11], [12]. However existing techniques are computationally expensive.

We propose simple and fast algorithm to detect a block of text from the very large size that relates to title and headers. Our algorithm for detection of such zones of attention includes the following steps. First of all, initial *rgb* image is converted to grayscale *I*. Next step is to downsample the original document image to a size that provides recognizability of text with the size 16-18 pt or more. For example, scanned document image with a resolution 300 dpi should be downsampled 5 times. The resulting image of A4 document has size 700x500 pixels. Handling of grayscale downsampled copy of initial image allows to decrease processing time significantly.

Downsized text regions look like a texture. The areas contain bulk of edges. So, to reveal text regions edge detection technique can be applied. For this purpose we use Laplacian-of-Gaussian (LoG) filtering and zero-crossing. LoG filtering is a convolution of downsampled image *I* with kernel k_g :

$$k(x, y) = \frac{(x^2 + y^2 - 2 * \sigma^2) * k_g(x, y)}{2\pi\sigma^6 \sum_{x=-N/2}^{N/2} \sum_{y=-N/2}^{N/2} k_g(x, y)}$$

$$k_g(x, y) = e^{-(x^2 + y^2) / 2\sigma^2}$$

where N is size of convolution kernel, σ - standard deviation, (x, y) - coordinates of Cartesian system with origin at the center of the kernel.

Zero-crossing approach with fixed threshold T is preferable for edge segmentation. The edges are labeled on the binary image BW using following rules:

$$BW(x, y) = 1 \text{ if } (|Ie(x, y) - Ie(x, y+1)| \geq T \text{ and } Ie(x, y) < 0 \text{ and } Ie(x, y+1) > 0) \\ \text{ or } (|Ie(x, y) - Ie(x, y-1)| \geq T \text{ and } Ie(x, y) < 0 \text{ and } Ie(x, y-1) > 0) \\ \text{ or } (|Ie(x, y) - Ie(x-1, y)| \geq T \text{ and } Ie(x, y) < 0 \text{ and } Ie(x-1, y) > 0) \\ \text{ or } (|Ie(x, y) - Ie(x+1, y)| \geq T \text{ and } Ie(x, y) < 0 \text{ and } Ie(x+1, y) > 0); \\ \text{ otherwise } BW(x, y) = 0,$$

where Ie is outcome of LoG filtering, (x, y) are coordinates of pixel.

For segmentation of text regions we look for the pixels which have a lot of edge pixels in vicinity according to the equation:

$$L(x, y) = \begin{cases} 1, & \sum_{i=x-dx/2}^{x+dx/2} \sum_{j=y-dy/2}^{y+dy/2} BW(i, j) > Tt \\ 0, & \text{otherwise} \end{cases}$$

where BW – image with detected edges, L – image of segmented text regions, dx, dy – sizes of block, Tt is a threshold.

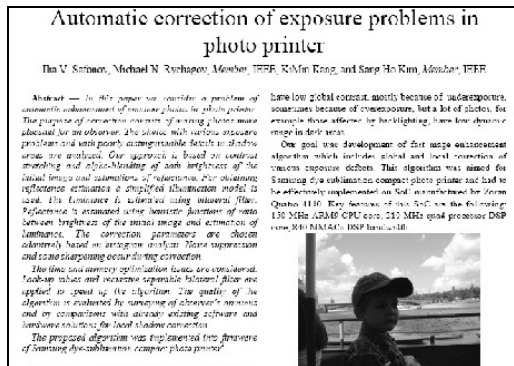
In addition, regions corresponding to vector graphics such as plots and diagrams are segmented too. Next step is Further for L labeling of connected regions in L and calculation of its bounding boxes. Regions with a small height or width are eliminated.

The calculation of the average size of character for each region of the text and selection multiple zones with a maximal average size of the character are performed on the next step. Let us consider how to calculate the average size of the character of the text region that corresponds some connected region in the image L . The text region can be designated as:

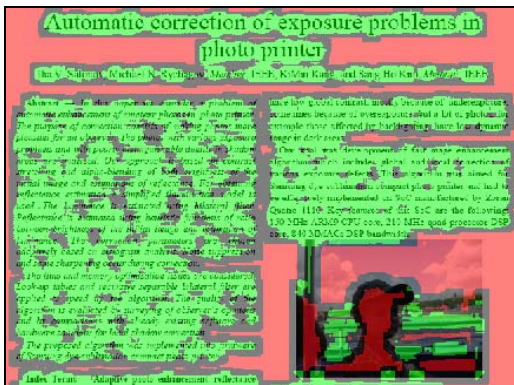
$$Z(x, y) = I(x, y) \times L(x, y), \quad \forall x, y \in \Omega$$

where Ω is some connected region in image L .

The image Z is binarized by threshold. We use well-known Otsu algorithm to calculate threshold T_z for histogram calculation the pixels which belongs to Ω are considered only. We apply optimized version of the algorithm as described in [13]. Thus connected regions on the binary image Zb are labeled. The size of bounding box is calculated for all connected regions in Zb . The average size of character is calculated as the average of the sizes for the Ω zone from L . If the number of connected regions in Zb is too small then the text region is eliminated.



a)



b)

Figure 6: Detection of text regions: (a) downsized scanned document, (b) detected text regions (in green).

Fig. 6 illustrates our approach for detection of text regions. Detected text regions L are marked in green color in fig. 6b. Image Z consists from all the connected regions. The average size of the character is calculated for the dark connected areas inside the green region. This is the right way to detect title of paper for image in fig. 6a.

At the final stage of our approach photographic illustrations are identified because they are important for the document recognition as well. The image I is divided into non-overlapped blocks with a size $N \times M$ for the detection of embedded photos. For each block Ei energy of the normalized co-occurrence matrix is calculated:

$$C_I(i, j) = C_I(i, j) + 1 \text{ if } I(x, y) = i \text{ and } I(x + dx, y + dy) = j, \quad \forall x, y$$

$$N_I(i, j) = \frac{C_I(i, j)}{\sum_i \sum_j C_I(i, j)}, \quad E_i = \sum_i \sum_j N_I^2(i, j),$$

where x, y are coordinates of pixels of a block, dx, dy are displacements.

If $E_i < 0.01$ then all pixels of the block are marked as related to photo. Further all adjacent marked pixels are combined to connected regions. Regions with the small area are eliminated. Regions with too large area are eliminated too because they also belong to the complex background of a document as a rule. The bounding box of region with the largest area defines zone of the embedded photo. Fig. 7 shows results of detection of the blocks related to photographic illustration.

Automatic correction of exposure problems in photo printer

Ilya V. Salimov, Michael N. Rykachev, Member, IEEE, Kirill Kozlov, and Sergey Ho Kim, Member, IEEE



Figure 7: Detection of photographic illustration inside document image.

3.5 Animation creation

First of all the sequence order of zones is selected for animation creation. Always first frame represents a whole downsampled image that is conventional thumbnail. The subsequent zones are selected to provide the shortest path across the image. The animation can be looped. In this case final frame is a whole image too. The animation simulates the following camera effects: tracking in, tracking out and panning between attention zones, slow panning across large attention zone and pauses on the zones. Tracking in, tracking out and panning effects between two zones are created by constructing a sequence from N frames. Each frame of the sequence is prepared with the following steps:

- calculate coordinates of bounding box for cropping zone using line equation in parametric form:

$$x(t) = x_1 + t \times (x_2 - x_1),$$

$$y(t) = y_1 + t \times (y_2 - y_1),$$

where (x_1, y_1) are coordinates of start zone, (x_2, y_2) are coordinates of end zone, t is increased from 0 to 1 with step $dt=1/(N-1)$;

- crop image using coordinates of calculated bounding box with preserving of aspect ratio;
- resize cropped image to size of destination thumbnail.

Fig. 8 demonstrates example of the animated thumbnail for the image of scanned document. The title and embedded photo are detected on the first stage. The fragment of image with a title is appropriate for quality estimation. Animation consists of 4 transitions between entire image and these two zones as well as viewing a relatively large zone of title. The first sequence of frames look like a camera tracking-in to the left side of title zone.

The second sequence of frames look like a slow camera panning across zone of title; after that the frame is frozen on a moment for visual quality estimation. The third sequence of frames look like camera panning from the right side of title zone to embedded photo; after that frame is frozen on a moment. The final sequence of frames look like a camera tracking-out to entire page; after that frame with entire page is frozen on a moment. The sequence of the frames allows to identify the image content confidently.

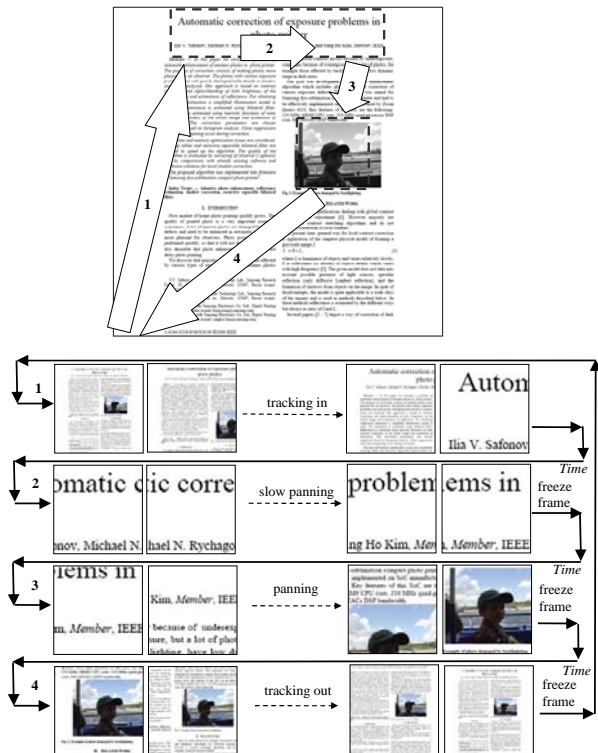


Figure 8: Frames of animated thumbnail for document image.

4. RESULTS AND DISCUSSION

We conducted user study to estimate effectiveness of animated thumbnail in comparison with conventional thumbnail. The study was focused on recognition of image content and its quality. The survey was held among ten persons. Surely ten people is not enough for deep and confident investigation. However it is enough to demonstrate advantages of animated thumbnail.

Survey participants were asked to complete three tasks on one laptop PC with WinXP independently of one another but in the similar conditions. Conventional thumbnails were viewed in the standard Windows Explore application. Animated thumbnails were created as animated GIF files. Each participant has one minute to solve the tasks.

First task was selection of two photos with the particular person known to the respondent. Total number of viewed thumbnail was eight. The photos used for thumbnail generation were never seen before by respondents. Fig. 9 shows conventional thumbnails. The majority of faces are too small for confident recognition. Nevertheless percentage of right answers was not so bad; 62% respondents selected photos with the person. It is probably explained by very high recognition and cognitive abilities of the people. Such characteristics as hair color, head form, build of body, height, typical pose and expression allow to identify known person even if size of photo is extremely small. However the recognition results for animation thumbnails are much better; 95% of respondents selected right photos. Fig. 10 shows animated thumbnails frames which contain enlarged faces. In most cases the enlarged face allows to identify person. Even if a face is not detected as attention zone, walk through zoomed in image fragments allows to see the face in detail often. Perhaps 5% of errors are explained by carelessness because faces were frozen on a moment only and time for task completion was limited.

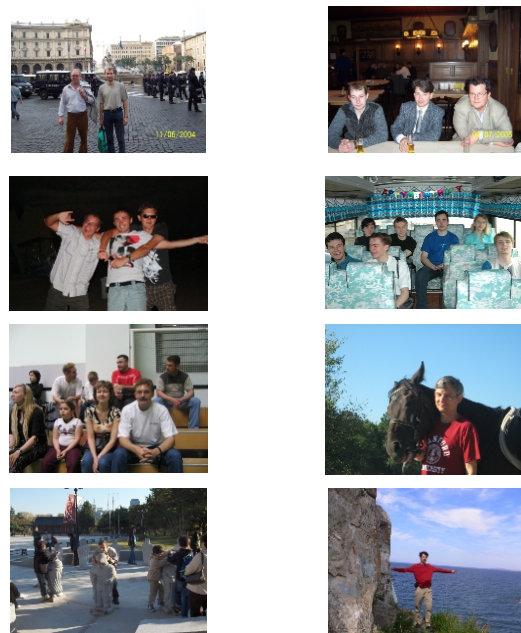


Figure 9: Conventional thumbnails in the survey for selection of photos with the certain person.

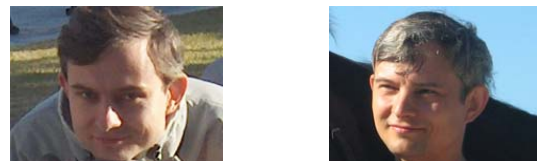


Figure 10: Frames of animated thumbnails in the survey for selection of photos with the certain person.

Second task was selection of two blurred photos from eight. Fig. 11 shows conventional thumbnails of the survey. It is almost impossible to detect blurred photos by means of thumbnail viewing. Only 35% gave the right answers. It is a little bit better than random guessing. Two responders had better results than another because they had a big experience in photography and understood shooting conditions which can cause a blurred photo. Animated thumbnails demonstrate zoomed in fragments of photo. It is allow to select low-quality photos. In our survey 89% of respondents detected right photos by viewing of animated thumbnail frames.

Fig. 12 shows enlarged fragments of sharp photos in top row and blurred photos in bottom row; difference is obvious and blurriness is detectable. 11% of errors are explained by subjective interpretation of blurriness concept probably. Indeed sharpness and blurriness are not formalized strictly and depends from viewing conditions.

Third task was selection two scanned images which represent the document related to “Descreening images” topic. Total number of documents was nine. The conventional thumbnails of the scanned images are shown on fig. 13. Percentage of right answers for conventional thumbnail is 22. It corresponds to random guessing. In general it is impossible to solve the task properly using conventional thumbnails of small size. Contrary animated thumbnails provide high enough level of right answers. 85% of respondents selected pages related to “Descreening” topic due to zooming and panning through title of the papers as it is shown on the fig. 14.

Summary percentage of right answers for the 3 tasks is shown on fig. 15. Animated thumbnail provides capability to recognize image content and to estimate quality confidently and outperforms conventional thumbnail considerably.

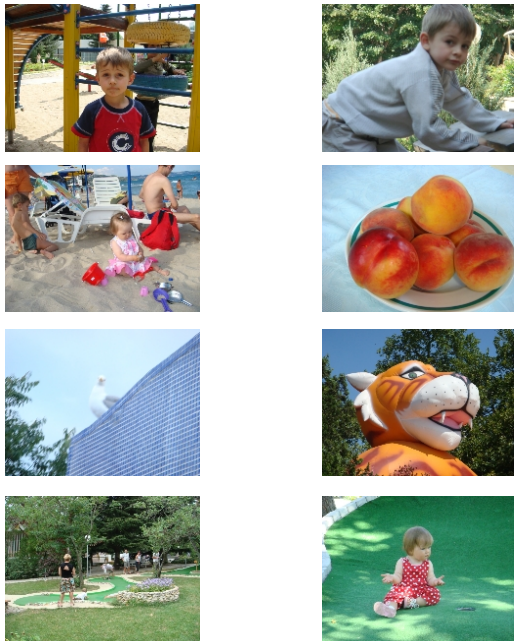


Figure 11: Conventional thumbnails in the survey for selection of blurred photos.

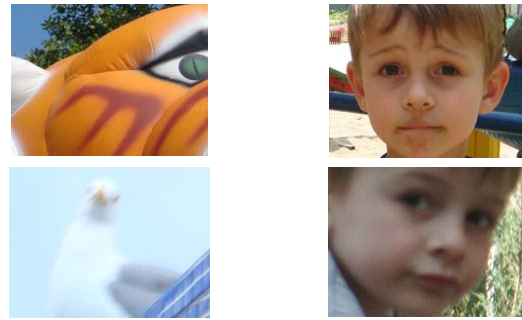


Figure 12: Frames of animated thumbnails in the survey for selection of blurred photos: top row for sharp photos, bottom row for blurred photos.

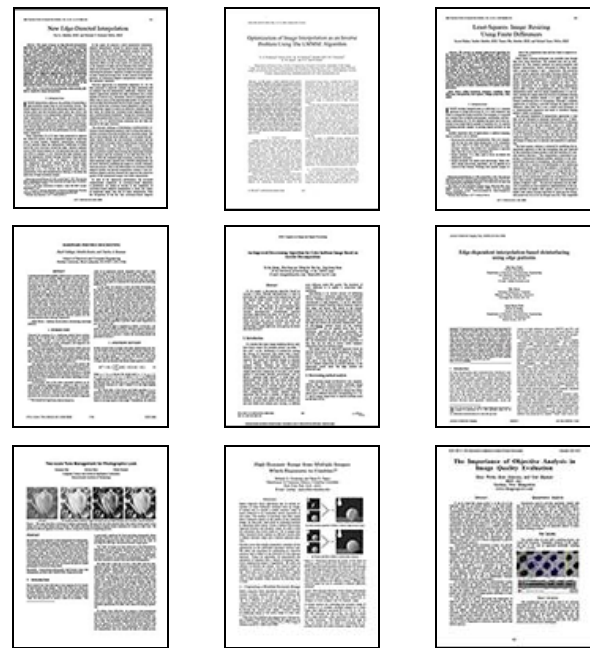


Figure 13: Conventional thumbnails in the survey for selection of documents related to particular topic.

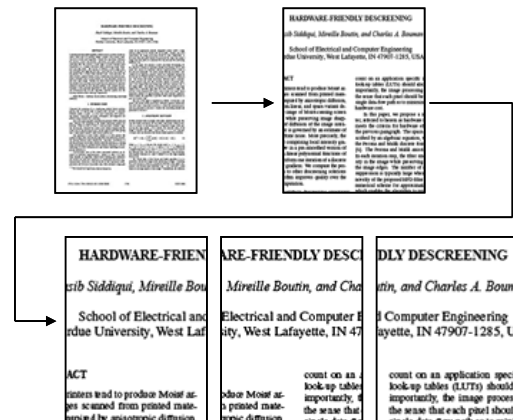


Figure 14: Frames of animated thumbnail with panning through title of the paper.

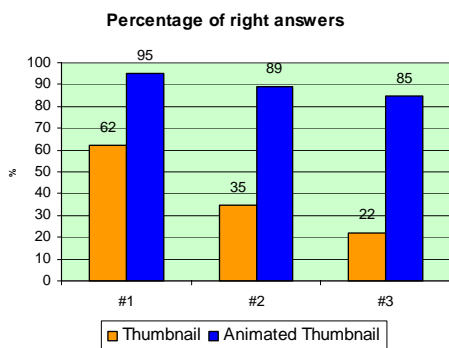


Figure 15: Summary results of the user study.

5. CONCLUSION

Animation thumbnail is applicable in devices with small display for user-friendly viewing of still image thumbnails. First of all it is valuable for mobile phones, Digital Still Cameras, Digital Photo Albums, Digital Photo Frames, Photo Printers and MFP. In addition it is impressive way for navigation through image collections in software and web-applications.

The general idea of animated thumbnail can be extended on thumbnails for separate frames of video and documents such as PDF, MS Word etc.

ACKNOWLEDGMENT

Authors would like to thank Ekaterina Tolstaya for discussions about this paper. Thanks also go to Graphicon's anonymous reviewers for their comments.

REFERENCES

- [1] B.Suh, H.Ling, B.B.Bederson, D.W.Jacobs "Automatic Thumbnail Cropping and its Effectiveness", *Proceedings of ACM UIST, 2003*.
- [2] R.Samadani, T.Mauer, D.Berfanger, J.Clark, B.Bausk "Representative Image Thumbnails: Automatic and Manual", *Electronic Imaging, 2008*.
- [3] K.Berkner, E.L.Schwartz, C.Marle "SmartNails - Display and Image Dependent Thumbnails", *Electronic Imaging, 2004*.
- [4] P.Viola, M.Jones, "Rapid object detection using a boosted cascade of simple features", *In Proc. of Conference Computer Vision and Pattern Recognition, 2001*.
- [5] B.Ma, W.Zhang, S.Shan, X.Chen, W.Gao, "Robust head pose estimation using LGBP", *Proc. of Conference on Pattern Recognition, 2006*.
- [6] M.A.Egorova, A.B.Muryinin, I.V.Safonov, "An improvement of face detection algorithm for colour photos", *Proc. of Conference on Pattern Recognition and Image Analysis, 2008*.
- [7] L.Itti, C.Koch, E.Niebur, "A model of saliency-based visual attention for rapid scene analysis", *IEEE Transactions on Pattern analysis and machine intelligence, Vol. 20, No. 11, pp. 1254-1259, 1998*.

[8] L.Itti, C.Koch, "A saliency-based search mechanism for overt and covert shifts of visual attention", *Vision Research 40, pp 1489-1506, 2000*.

[9] W.F. Lee, T.H. Huang, Y.H. Huang, M.L. Chu and H.H. Chen, "Efficient Construction of Saliency Map", *Proc. SPIE Vol. 7240, 2009*.

[10] P.Longhurst, K.Debattista, A.Chalmers, "A GPU-based saliency map high-fidelity selective rendering", *Proc. of ACM AFRIGRAPH, 2006*.

[11] S. Klink, A. Dengel, T. Kieninger, "Document structure analysis based on layout and textual features", *Proc. 4th IAPR International Workshop on Document Analysis, pp. 99-111, 2000*.

[12] M. Aiello, C. Monz, L. Todoran, M. Worring, "Document understanding for a broad class of documents", *International Journal on Document Analysis and Recognition, vol. 5, no. 1, pp. 1-16, 2002*.

[13] K.C.Lin, "On improvement of the computation speed of Otsu's image thresholding", *Journal of Electronic Imaging, 14(2), 2005*.

About the authors

Iliia V. Safonov received his MS degree in automatic and electronic engineering from Moscow Engineering Physics Institute/University (MEPhI), Russia in 1994 and his PhD degree in computer science from MEPhI in 1997. Since 1998 he is an assistant professor of faculty of Cybernetics of MEPhI (now National Research Nuclear University) while conducting researches in image segmentation, features extraction and pattern recognition problems. Since 2004, Dr. I. Safonov joined Image Enhancement Group, Samsung Research Center, Moscow, Russia where he is engaged in photo, video and document image enhancement projects.

Victor V. Bucha received his MS degree in computer science from Belorussian National Technical University in 2003 and his PhD degree in computer science from United Institute of Informatics Problems, Belarus in 2006. Since 2007, Dr. V. Bucha joined Samsung Research Center, Moscow, Russia where he is engaged in image processing and computer vision projects.

New Ridge Count Model based on Fingerprint Image Topology

Vladimir Gudkov

Department of Applied Mathematics

Chelyabinsk State University, Chelyabinsk, Russia

diana@sonda.ru

Abstract

The paper briefly describes a mathematical model of fingerprint based on ridge count vectors. They are computed from topological descriptors of minutiae point neighborhood and stored in the template with the list of minutiae. Templates are used to identify the fingerprint. The approach allows designing ridge count value as the fractional number.

Keywords: *Fingerprint, topology, minutiae, ridge count, fractional number.*

1. INTRODUCTION

In computerized systems (CS) verification and identification of fingerprint images (FI) usually realizes with templates. Template is a set of compact formal structures, for example, minutiae, such as bifurcation and ending. They are usually detected using skeleton lines [1–3], shown in figure 1, as a formal presentation of FI.



Figure 1. Skeleton and minutiae of fingerprint image

Identification procedure regards the template as mathematical model of FI, which should be based on necessary and sufficient number of minutiae [2, 3]. There are known different mathematical models that are used in CS to prove image individuality [1, 3, 5–8]. Each of those models focuses on identification accuracy enhancement; however, there is no any other better model free off defects [1, 3].

2. RIDGE COUNT

In dactyloscopy when compare the patterns they usually bear on minutiae and ridge count between them [3, 5]. Such method reflects not only realization of different designation CS but also standards requiring its implementation [9]. However, ridge count, which in Criminalistics measuring along the straight line between two minutiae as discrete random quantity X with integral values, is characterized with instability. The reason of that is stochastic nature of biometric parameter [3], conflicting to existing conception of ridge count measuring. It is evidently, if not to deny heavy noisiness of FI. However, there are many other factors, which are visualized in different operating conditions and not accounted just the noise.

During fingerprint latent formation, minutiae can mutate. At mutation, an ending is closing into the bifurcation, but bifurcation is tearing into ending. This is frequently met first type mutation. [1]. Minutiae type – bifurcation and ending – not always help to consider it even at trivial alteration of a pattern. For example, in figure 2 ridge count value 3 transforms into value 2 at closing of v ending into bifurcation (number of transitions from line to line is summarized). At such type of mutation of ending u ridge count becomes appreciably different.

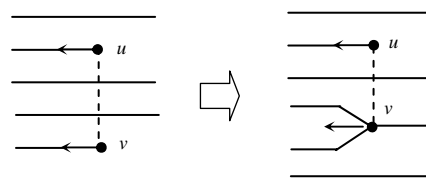


Figure 2. First type of mutation and change of ridge count between endings

Moreover, it is observed dependence of ridge count value x from lines curvature [7, 8]. At FI deformation lines curvature is changing. In figure 3 the value x measured along the dotted line is sensitive to deformation, which results in pattern lines straightening.

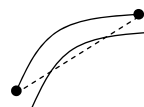


Figure 3. Instability of ridge count value in the area of significant curvature of lines

The other effects of latent print formation are also influence, they deform pattern saving good quality of FI. These are both minutiae displacement along lines, and dead zones appearing near minutiae, if measuring realized close to them, and indeterminacy of

minutiae multitude choice, up to which ridge count is measured, and ridge count ambiguity at its changing along the lines [3, 7, 8]. Thus, stochastic nature of biometric parameter should be considered in FI identification procedure, for example, by means of tolerance extension for discrete random quantity X deviation. As result, it is increase of identification errors.

3. NEW RIDGE COUNT MODEL

Ridge count rely on minutiae, multitude of which is in terms of

$$L_m = \{M_i = \{(x_i, y_i), \alpha_i, t_i\} | i \in 1..n\}, \quad (1)$$

where M_i – minutiae with i index number; (x_i, y_i) , α_i , t_i – coordinates, direction and type of minutiae; $n = |L_m|$ – cardinal number. Minutiae are detected in informative area of an image, which is darkened in figure 1.

Coordinates (x_i, y_i) are determined as coordinates of skeleton node [2]. Direction α_i is the angle determined with a chain of skeleton nodes for ending and a tree of chains for bifurcation. It points to the area of lines number increasing (look figure 2). Type $t_i \in \{0,1\}$ is determined with skeleton nodes valence like the nodes of graph [4].

3.1 Topological vectors

Minutiae topological vectors L_v model is based on multitude L_m (1) in terms of

$$L_v = \{V_i = \{(e_j, n_j)\} | i \in 1..n, j \in J = 1..m\}, \quad (2)$$

where V_i – minutiae topological vector; n – cardinal number, $n = |L_v| = |L_m|$; i – index as number of topological vector; j – link number; e_j – event, formed with minutiae with number n_j ; m – links number at section depth, equal to g lines, as follows

$$m = \begin{cases} 4g + 1, & \text{if } t_i \in \{1\} - \text{ending,} \\ 4g + 3, & \text{if } t_i \in \{0\} - \text{bifurcation.} \end{cases} \quad (3)$$

Let us dwell on the procedure of topological vectors synthesis. In the informative area of FI lines are marked out, skeleton is formed and two types of minutiae are detected: endings and bifurcations (look figure 1). Direction of minutiae M_i points to the area of lines number increase. It is parallel to tangent of papillary line image. Each minutia is numbered. Also for each minutia, we draw projections to the right and to the left transversely to the direction vector of the minutiae onto adjacent lines of skeleton and fix the projections. In figure 4 the projections are shown with dotted lines and two corresponding nodes of skeleton on the lines 1 and 2 are painted over.

Consider the node p of a skeleton, which conform to minutia M_i , and draw the section across its coordinates (x_i, y_i) to the right and to the left at a distance of g lines transversely to the tangent to lines being crossed and enumerate on gyrate the dissected lines (hereinafter ‘links’), which turn clockwise. The

section passes and traces the lines curvature [6]. The section g depth is varying from one to eight lines to the right and the same to the left. One line in a section forms two links, total number of which is calculated according to (3).

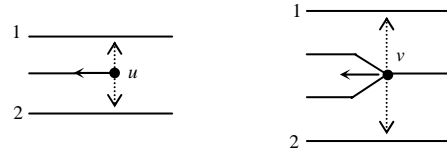


Figure 4. Projections for ending and bifurcation

Topological vector V_i is determined by the section. Then we follow the move of every link by turns on each link, not leaving it beginning from the section until finding another minutia, located on the link, or a minutiae projection, located on an adjacent line to the right or to the left of the link. Possible events are shown in figure 5 and indicated with number in binary code.

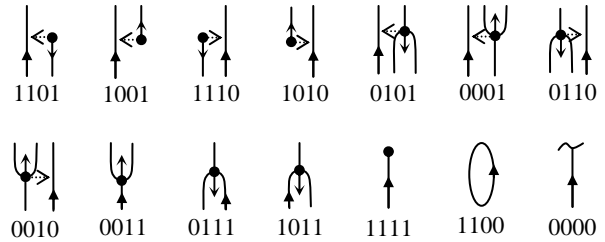


Figure 5. Events

The number of minutia, initiated the event is associated with the event as the number (14 in sum) detected on the link. The event is tied to the link number. For 0000 and 1100 events minutiae numbers are absent: the line either breaks in the uninformative area or closes and there are no either minutia no projection on the line. Event and minutia number form ordered pair (e_j, n_j) , corresponding to j -link.

Definition 1. Topological vector V_i is the numbered set of links with ordered pairs (e_j, n_j) .

Bits of codes for the event determines minutiae type, its direction and location, against the link direction and others (look figure 5). Topological vector is built for each minutia.

Hereon the topological vector formation is finished.

In figure 6 in section for ending 19 the links are enumerated as 0–16, and corresponding topological vector is shown in the table 1. Such section for bifurcation 19 is shown in figure 7, and topological vector, formed with links 0-16, is shown in the table 2. The sections are shown with dotted lines. They cut lines into links, enumerated on gyrate clockwise. These figures show well-known mutation of ending 19 into bifurcation 19.

Comparative analyses of topological vectors with Sparrow method [7] show a series of advantages. *Firstly*, the section is built along the curve, following the direction of lines curvature, not along the straight line. *Secondly*, during event determination minutia projection is taken into account, which in case of mutation avoids the loss of information in the form of event omission during link tracing. *Thirdly*, links are enumerated on gyrate without links omission, which allows increasing the section size keeping the contents of shorten and lengthen vectors common part. This increases stability and informativity of mathematical model.

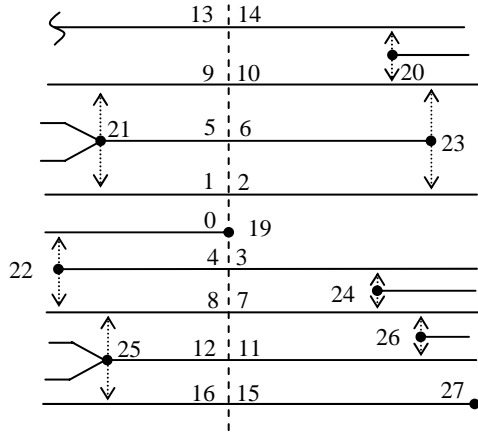


Figure 6. Section for ending

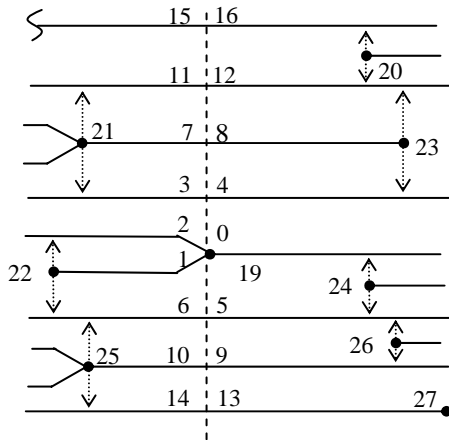


Figure 7. Section for bifurcation

Table 1. Topological vector of ending

Link number	Event	Index
0	1110	22
1	0001	21
2	1110	23
3	1001	24
4	1111	22
5	0011	21
6	1111	23
7	1010	24
8	0010	25
9	0010	21
10	1010	20
11	1010	26
12	0011	25
13	–	–
14	1001	20
15	1111	27
16	0001	25

Table 2. Topological vector of bifurcation

Link number	Event	Index
0	1001	24
1	1111	22
2	1110	22
3	0001	21
4	1110	23
5	1010	24
6	0010	25
7	0011	21
8	1111	23
9	1010	26
10	0011	25
11	0010	21
12	1010	20
13	1111	27
14	0001	25
15	–	–
16	1001	20

3.2 Ridge count vectors

Disadvantages of the classical ridge count are compensated in the model of ridge count based on FI topological vectors. Ridge count model L_r based on model L_v using (2) in the following form

$$L_r = \{ R_i = \{ (r_k, n_k) \} | i \in 1..n, k \in 1..t \}, \quad (4)$$

where R_i – minutiae ridge count vector is the ordered set of ordered pairs (r_k, n_k) with k index; n – cardinal number and $n = |L_r| = |L_v|$; i – index as a number of ridge count vector; k – link number; t – quantity of links, equal to quantity of minutiae in R_i vector; r_k – ridge count value, and n_k – minutiae number, using (1) on k -link.

Let us dwell on the procedure of ridge count vectors building. On the basis of L_v using (2) auxiliary topological vectors are calculated in terms of

$$L'_v = \{ V'_i = \{ (e_j, n_j, r_j) \} | i \in 1..n, j \in J = 1..m \}, \quad (5)$$

where r_j – ridge count value at j -link and

$$r_j = c_j + f(j, e_j); \quad (6)$$

c_j – transition count from line to line up to j -link in the section for topological vector V_i using (2); function f determines the correction value depending on j link and e_j event value.

Let s_j – state of the j -link, $s_j = \varphi(e_j)$, $s_j \in \{0,1\}$. Here $s_j = 0$ if the direction vector of minutiae directed along the link course; $s_j = 1$ if the direction vector of minutiae directed contrariwise to the link course. Function φ is realized in terms of

$$\varphi(v_i) = \{ _ , 0, 0, 0, _ , _ , 1, 1, _ , 0, 0, 1, _ , 1, 1 \},$$

where $\{_ \}$ – skip code. We can see that $\varphi(1)=0$ or $\varphi(5)=1$. So k -minutiae in vector V_i' linked with $s_j = \varphi(e_j)$ can be associated with state of the k -minutiae s_k .

Let the next in order k -minutiae in vector V_i' is met for N times and its number is $h = n_j$. Then it is possible to calculate an average value of ridge as follows

$$r_k = \frac{1}{N} \sum_{\substack{j \in J \\ n_j = h}} r_j, \quad (7)$$

corresponding to the minutia M_{n_j} in vector V_i' . The quantity t of such average values equal to the quantity of minutiae in vector V_i' . Ridge count vector is calculated as follows

$$R_i = \{(s_k, r_k, n_k)\}, \quad i \in 1..n, \quad k \in 1..t,$$

where for every k -minutiae there is ordered triplet (s_k, r_k, n_k) , where n_k – minutiae number, for which it is calculated the following r_k ; s_k – state of the k -minutiae; $n_k = n_j$. Quantity of t links using (4) not exceeds the quantity of m links using (3).

Definition 2. Ridge count vector R_i is the numbered set of links with ordered triplets (s_k, r_k, n_k) .

Links can be ordered by an angle in polar coordinate system, oriented along the direction of minutiae M_i , for which auxiliary vector V_i' is determined. In this case, an order of minutiae enumeration is similar to [5], but quantity of them is substantially less.

3.3 Calculation of correction function

Auxiliary topological vector from (5) allows determination of the value r_j of ridge count. Let us determine the set $L = \{1110, 1010, 0110, 0010\}$ of «left» projections and set $R = \{1101, 1001, 0101, 0001\}$ of «right» projections in the form of enumeration of events. Let us assign a set O_i of links of basic line for the end $O_1 = \{0\}$ and bifurcation $O_0 = \{0, 1, 2\}$, where $t \in \{0, 1\}$ – minutiae type from (1, 3). Let us make four sets of links Q_1, Q_2, Q_3 и Q_4 for the quadrants in the coordinate system, oriented along the direction of minutiae M_i . This is curvilinear coordinates; ordinate axis of which is curving along the section of topological vector, and an abscissa – along the basic line. These sets of links are shown in the table 3 for the ending in figure 6 and bifurcation in figure 7.

Table 3. Links in quadrants

Ending	Bifurcation
$Q_1 = \{4, 8, 12, 16\}$	$Q_1 = \{6, 10, 14\}$
$Q_2 = \{3, 7, 11, 15\}$	$Q_2 = \{5, 9, 13\}$
$Q_3 = \{2, 6, 10, 14\}$	$Q_3 = \{4, 8, 12, 16\}$
$Q_4 = \{1, 5, 9, 13\}$	$Q_4 = \{3, 7, 11, 15\}$

Relying on the following sets L, R, Q_1, Q_2, Q_3 и Q_4 , function of ridge count value correction can be written down in terms of

$$f(j, e_j) = 0,5 \operatorname{sign}(j, e_j), \quad (8)$$

where

$$\operatorname{sign}(j, e_j) = \begin{cases} +1, & \text{if } e_j \in L \wedge j \in Q_1, \\ -1, & \text{if } e_j \in R \wedge j \in Q_1, \\ -1, & \text{if } e_j \in L \wedge j \in Q_2, \\ +1, & \text{if } e_j \in R \wedge j \in Q_2, \\ +1, & \text{if } e_j \in L \wedge j \in Q_3, \\ -1, & \text{if } e_j \in R \wedge j \in Q_3, \\ -1, & \text{if } e_j \in L \wedge j \in Q_4, \\ +1, & \text{if } e_j \in R \wedge j \in Q_4, \\ +1, & \text{if } j \in Q_i. \end{cases} \quad (9)$$

Correction function from (8) for every projection on link adds or subtracts a half of transition count from one line to another line in a section for auxiliary topological vector using (5). Correction sign from (9) is determined with projection direction. If an event e_j corresponds to projection, which approaches the minutiae M_{n_j} to the basic line with links from O_i , then correction sign is negative, otherwise – positive. Intuitively it seems to be understood. However, the result of calculation using (7) is equal to fractional values. Actually, correction function and average values calculation help to convert ridge count values into fractional numbers. Under mutation, ridge count modification can remain the same as in classic approach, or become less. This positive affect is shown in the table 4 and table 5, where are shown the results of ridge count calculation for ending 19 in figure 6 and bifurcation 19 in figure 7. Minutiae in tables are ordered by angle in polar coordinate system, which corresponds to Shmakov's approach [5]. We can see from tables that for minutiae of the same name changing of ridge count Δr in case of mutation of first kind not exceed a unity. Values Δr calculation is shown in the table 6.

If minutiae located on links are subject to mutations, then ridge count change, less than a unity. It is being met more often. For example, if bifurcation 21 in figure 6 will tear to ending between 5-th and 9-th links, and in figure 7 will remain, then $\Delta r = 0,5 < 1$; for classic ridge count $\Delta r = 1$.

Table 4. Ending ridge count vector

Number of link	State	Ridge count	Index
0	1	0,75	22
1	0	3	25
2	1	4	27
3	0	2,5	26
4	0	1,5	24
5	1	1,75	23
6	0	3,5	20
7	0	2	21

Table 5. Bifurcation ridge count vector

Number of link	State	Ridge count	Index
0	1	0,(3)	22
1	0	2	25
2	1	3	27
3	0	1,5	26
4	0	0,5	24
5	1	1,75	23
6	0	3,5	20
7	0	2	21

Table 6. Ridge count change in vector

Ridge count change	Index
0,41(6)	22
1	25
1	27
1	26
1	24
0	23
0	20
0	21

Let us indicate advantages of ridge count vector. *Firstly*, at minutiae mutation, on both basic line and links, fractional ridge count allows reducing the changes of its values. *Secondly*, ridge count values do not depend on lines curvature, as if the procedure of ridge counts measuring along the straight line to be excluded. *Thirdly*, ridge count values are measured along the lines the same stably as across them. *Fourthly*, minutiae set, which are the part of ridge count vector, is unambiguously determined and limited with topological structure of pattern, but not conditional radius-vectors and the same mechanisms. *Fifthly*, state of minutiae is steady to line curvature in the area of loops, deltas and wholes. *Sixthly*, ridge count vector is steady to image deformation, which distorts the line curvature, as since it inherit positive features of topological vector. This increases the stability and informativity of ridge count vectors.

More over, ridge count vector is invariant to scale and rotating of FI the same as classical ridge count.

4. CONCLUSION

In the current work, the mathematical model of FI based on ridge count vectors and topological vectors is suggested. In spite of different missions in construction of either of the two models, it is indicated the total feature of them: the models use topological description of FI. As a result, such models describe image more completely as compared with models, suggested earlier [3, 5, 7, 8]. For the first time ridge count value means fractional number.

Requital for mentioned advantages is heightened complexity of models calculation. However, the models contain mechanisms that compensate influence of minutiae mutation, image deformation and lines curvature. This increases the stability of FI template.

Further direction of research work is seen in synthesis of mathematic models, oriented on FI presentation in the form of lines ridge count vectors, but not minutiae, with fractional ridge

count. Such researches will help in realization of FI indexation mechanisms, which multiply accelerate procedures of fingerprint images identification.

5. REFERENCES

1. Gudkov, V.U. Mathematical models of fingerprint image on the basis of lines description / V.U. Gudkov // GraphiCon'2009: Conf. proc. – 2009. – P. 223–227.
2. Honsales, R. Digital processing of images / R. Honsales, R. Wuds; tran. from Eng.; under edit. P.A. Chochia. – M.: Technosthere, 2006. – 1072 c.
3. Maltoni, D. Handbook of fingerprint recognition / D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar. – London: Springer Verlag, 2009. – 496 p.
4. Novikov, F.A. Discrete mathematics for programmers: textbook for institutes / F.A. Novikov. – Spb.: Piter, 2008. – 384 p.
5. Pat. 2054197 Russian Federation, MPK G 06 K 9/46. Papillary pattern print encoding method / V.L. Shmakov. – № 5061547/09; declared 03.09.1992; publication 10.02.1996; Bull. № 4. – 21 p.
6. Pat. 2321057 Russian Federation, MPK G 06 K 9/52, A 61 B 5/117. Papillary pattern print encoding method / V.U. Gudkov. – № 2006142831/09; declared 04.12.2006; publication 27.03.2008; Bull. № 9. – 13 p.
7. Pat. 5631971 USA, Int.Cl. G 06 K 9/00. Vector based topological fingerprint matching / M.K. Sparrow (Winchester). – Field: Jul. 15, 1994; Date of patent: May. 20, 1997; U.S.Cl. 382/125. – 17 p.
8. Sparrow, M.K. A topological approach to the matching of single fingerprints: development of algorithms for use on latent finger marks / M.K. Sparrow, P.J. Sparrow // US dep. comer. nat. bur. stand. spec. pub. – 1985. – № 500–126. – 61 p.
9. Ushmayev, O.S. Methods and development tools of multibiometric systems of double application: abstract of dissertation. ... Doctor of Technical Science/ O.S. Ushmayev. – M.: Tcifrovichok, 2009. – 42 p.

About the author

Vladimir Gudkov is a doctor at Chelyabinsk State University, Department of Applied Mathematics. His contact e-mail is diana@sonda.ru.

Automated Atlas-Based Segmentation of Nissl-Stained Mouse Brain Slices

Olga Senyukova, Alexey Lukin, Dmitry Vetrov*

Department of Computational Mathematics and Cybernetics

Moscow State University, Moscow, Russia

{osenyukova, lukin}@graphics.cs.msu.ru, vetrovd@yandex.ru

Abstract

We introduce a novel automated technique for segmentation of anatomical structures in the images of Nissl-stained histological slices of mouse brain. Segmentation method includes atlas-based supervised learning. An experimental mouse brain slice is preliminarily associated with the corresponding slice from the mouse brain atlas. Then a preprocessing procedure is performed in order to enhance the quality of the experimental image and make it as close to the corresponding atlas image as possible. An effective method of luminance equalization, which is an extension to Retinex algorithm, is proposed.

A supervised learning is performed on the atlas image associated with the experimental slice. A Random Forest is trained on a data derived from the atlas image along with its annotated map, and experimental image pixels are then classified into anatomical structures. The result is refined by Markov random field.

Preprocessing and segmentation procedures have been tested and evaluated on real experimental Nissl-stained slices.

Keywords: *automated segmentation, luminance equalization, Retinex, random forest, Markov random field.*

1. INTRODUCTION

Segmentation of mouse brain images into anatomical structures is a very important problem. It is an essential step of any analysis procedure performed on these images. Manual segmentation turns out to be rather tedious and time-consuming process which requires expert knowledge. Therefore, automated segmentation of mouse brain images is a challenging task.

Fortunately, several publicly available mouse brain atlases exist which can effectively aid the automated segmentation procedure.

Allen Brain Atlas [1] consists of 132 coronal sections evenly spaced at 100 μm intervals and annotated to a detail of numerous brain structures. The images were obtained using Nissl-staining protocol [1] and were manually enhanced using contrast adjustment, etc. Both histological images and corresponding colored annotated maps are presented in the Allen Brain Atlas.

In the present work, we deal with real experimental Nissl-stained slices of mouse brain provided by P. K. Anokhin Institute of Normal Physiology.

Each slice is considered separately. Assuming that for a given experimental slice we know which slice in the atlas it corresponds to, we can use atlas anatomical annotation as a reference. If we put an atlas annotated map directly over the experimental image, it will give a very rough approximation of anatomical structures of the experimental slice because individual local differences in each brain

structure's shape and size are not taken into account. Therefore, an annotated map from the atlas should not be used directly for automated segmentation. But, along with this annotated map, a corresponding histological image from the atlas can serve as a labeled training set for supervised learning algorithms for classification of image pixels into anatomical structures. In this work, 17 basic brain structures are considered, such as cerebral cortex, hippocampal formation, thalamus, etc. A Random Forest [2] was chosen as a classification algorithm because it is a fast and highly accurate method which can deal with a very large number of features.

Classification of individual pixels can produce some clutter, especially along the boundaries of anatomical structures. In order to overcome this problem, probabilistic outputs from the random forest are incorporated into Markov random field for further refinement of automated segmentation result.

In order to make the classification algorithm work properly, an experimental image should be preprocessed so that it looks as close to the corresponding atlas image as possible. Preprocessing is always necessary for real experimental images due to non-uniform lighting, low contrast, and other artifacts introduced by the slice image acquisition procedure.

The rest of the paper is organized as follows. An overview of existing mouse brain segmentation methods is given in section 2. Automated preprocessing procedure for experimental images, including luminance equalization, is described in section 3. Automated atlas based segmentation algorithm is described in section 4. Experimental results are presented in section 5. Discussion and conclusion are in section 6.

2. PREVIOUS WORK

Several works devoted to automated mouse brain segmentation exist.

The paper [3] describes a registration scheme to automatically annotate hierarchical brain structures in Nissl-stained mouse brain images using Allen Brain Atlas as a reference. A non-rigid deformation field which best maps the atlas onto the subject image is obtained and it can be used to warp the atlas annotation image thus automatically creating annotations for the subject. Although in our case we know which slice from the atlas an experimental slice corresponds to, differences in shape and size of anatomical structures can be too big for nonlinear warping methods. So, this class of methods can hardly be used for our problem.

A method for segmentation of anatomical structures in histological data is presented in [4]. Segmentation is carried out slice-by-slice by minimizing a cost functional that enforces a compatibility of partitions with corresponding models of region of interest and background (based on intensities and spatial locations) together with the alignment of boundaries with image gradients. This method was designed for segmentation of only one anatomical structure at a time; it does not take into account spatial relations between structures. Therefore it is only suitable for segmentation of anatomical struc-

*This work has been supported by Russian Foundation for Basic Research, project No. 09-04-12215. The authors thank Konstantin Anokhin, Alexander Velizhev and Anton Osokin for a useful discussion.

tures with clear boundaries.

In [5], the classification of voxels in the mouse brain into structures is accomplished by combining magnetic resonance intensity features with spatial priors into a discriminant Bayes classifier. These spatial priors integrate information about the location of structures in the brain as well as the contextual relationship between structures. This contextual information is modeled using the concept of Markov random field (MRF).

The authors of [6] proposed a segmentation method of mouse brain magnetic resonance microscopy (MRM) images that guides the Markov random field clustering with edge information of the MRM images combined with information on the intensity distribution of the various brain structures taken from the atlas.

In [7], authors developed an extended MRF method for automated segmentation of 3D MRM mouse brain images. The eMRF employs the posterior probability distribution obtained from a support vector machine (SVM) to generate a classification based on the MR intensity. To maximize the classification performance, the eMRF uses the contribution weights optimally determined for each of three potential functions: observation, location, and contextual functions, which are traditionally equally weighted.

Magnetic resonance microscopy images of mouse brain differ from Nissl-stained histological images. The latter have a grained, but more accurate pattern. Due to such granulation, classification of individual pixels into anatomical structures based on intensity gives poor results in our case. Nevertheless, the boundaries of anatomical structures are observable. Taking into account that we deal with 2D, not a 3D data, a superpixel-based approach, similar to [8] is introduced in this paper.

3. LUMINANCE EQUALIZATION

Brain slice images obtained from the scanner have very non-uniform illumination (Fig. 1a), which presents a problem for the process of segmentation. A luminance equalization stage of our algorithm restores the luminance uniformity and makes the appearance of the image closer to reference images from the atlas.

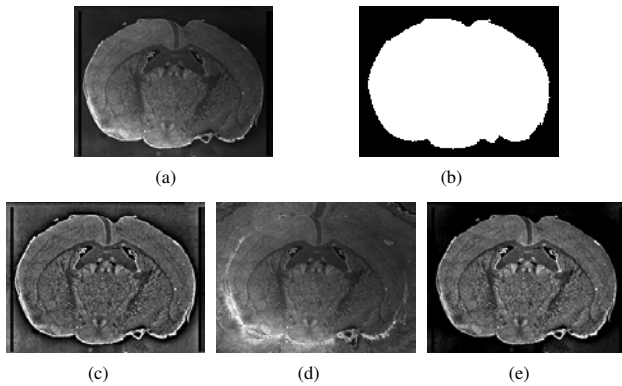


Figure 1: a: raw brain slice image from the scanner; b: brain slice mask; c: retinex algorithm; d: padding of the image outside the slice contour; e: proposed luminance equalization algorithm.

A well-known method for luminance equalization is Retinex [9]. It can be roughly summarized as division of the image by its Gaussian-smoothed copy. The rationale behind the method is that the observed image $Z(x, y)$ can be represented as a product $Z(x, y) = R(x, y)I(x, y)$ of the ground-truth (“reflectance”) image $R(x, y)$ and a non-uniform illumination $I(x, y)$. In the absence of accurate information about the illumination image $I(x, y)$,

it is estimated as a Gaussian smoothing of the observed image: $I(x, y) = \mathcal{G}(Z(x, y))$. So, $R(x, y)$ is estimated as $R(x, y) = \frac{Z(x, y)}{\mathcal{G}(Z(x, y))}$.

This method has two drawbacks when applied to our problem (Fig. 1c). First of all, since there is an abrupt change of luminance at the boundary contour of the slice, Gaussian smoothing underestimates the luminance of outermost slice pixels. This results in a bright halo at these areas after Retinex. Secondly, Retinex produces visible halos around sharp image edges, for a similar reason.

To address the first problem, we only apply smoothing to the area inside the previously localized brain slice contour (Fig. 1b). Contour localization is performed with EDISON segmentation algorithm [10]. Filtering can be done either by directly constraining Gaussian filter taps to cover the interior area only. But a more computationally effective procedure is possible. We are extending the inner area of the brain slice into the outside (black) area of the image by circular mirroring (Fig. 1d). After that, a fast filtering code that works on the extended rectangular image $E(x, y)$ can be used.

To address the second problem, we have replaced a Gaussian filtering step \mathcal{G} in the Retinex algorithm with a nonlinear kernel. A median filter has been chosen for such a nonlinear kernel. Similarly to a multiscale Retinex algorithm [9], we are using 3 median filters of different size $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3$ in our implementation (whose radii are chosen as 4%, 7%, and 11% of the image width). The resulting formulas for luminance equalization of the extended image are given below:

$$R(x, y) = \frac{E(x, y)}{\mathcal{M}(E(x, y)) + \varepsilon}, \quad (1)$$

$$\mathcal{M}(x, y) = \frac{1}{3} \sum_{i=1}^3 \mathcal{M}_i(E(x, y)). \quad (2)$$

Here ε (chosen as 2% of peak image brightness) is used to prevent over-amplification of the noise in dark image regions.

The final stage of the process is histogram normalization (“auto-levels” operation). An example of the resulting image can be seen in Fig. 1e.

4. AUTOMATED ATLAS-BASED SEGMENTATION

4.1 Features calculation

As mentioned before, a mouse brain atlas can be used as a labeled training data for classification algorithms. A training set is constructed of a number of training samples. Each sample is a pixel’s feature vector along with its label — the ID of the anatomical structure which this pixel belongs to.

This section is devoted to calculation of pixel features. We propose a new combination of features for classification — superpixel features together with location priors for each pixel.

4.1.1 Superpixel features

An individual pixel can only be characterized by three features: (x, y) -position and intensity value. This information is not enough to draw conclusions about its belonging to a particular anatomical structure because a pixel neighborhood is much more informative since it also contains texture information. If we consider an arbitrary neighborhood of the pixel, for example, a square window, it may belong to more than one brain structure in the case when it contains a structure boundary(-ies). Therefore, such pixel neighborhoods cannot serve as training samples — they will confuse a

classifier. Pixel neighborhoods should be image patches (“superpixels”) lying inside the boundaries of anatomical structures. Such superpixels are acquired by an extended Mean Shift segmentation algorithm [11]. Then, a set of features is calculated for each superpixel. These features, taken from [12], are divided into 4 categories:

- average color statistics (mean intensity and intensity variance)
- average geometry statistics (square, X-Y covariance, etc.)
- neighborhood statistics (intensity difference with neighboring superpixels)
- sorted color statistics (intensity quantiles).

Each pixel’s feature vector is assigned with features of a superpixel it belongs to.

4.1.2 Location priors

Assuming that the experimental slice under consideration is associated with a corresponding slice from the mouse brain atlas, we can assume that its configuration is the same as the atlas slice’s one, i.e. the same set of anatomical structures stays in similar positions. Therefore, it is possible to calculate location priors for each pixel $X = (x, y)$ using the atlas annotated map. Location priors is a vector L of prior probabilities of belonging of a pixel (x, y) to each of the anatomical structures:

$$L = [P(a(X) = s_1), P(a(X) = s_2), \dots, P(a(X) = s_N)], \quad (3)$$

where N is the number of anatomical structures and $a(X)$ is a mapping function from pixels space to anatomical structures labels space.

The idea of location priors was previously used in several works, for example in [5] and [7], but calculation of a probability vector for each spatial location required a training set of several brain instances in order to gather statistics. In our case there is only one brain instance (one slice) so we introduce a new method of calculation of location priors.

The probability of belonging of pixel X to an anatomical structure s_i , $i = 1, \dots, N$, is calculated as follows:

$$P(a(X) = s_i) = \begin{cases} 1, & a(Y) = s_i, \forall Y \in U(X) \\ 0, & \forall Y \in U(X) \nexists Y : a(Y) = s_i \\ 0.5, & \exists Y_1 : a(Y_1) = s_i \wedge \exists Y_2 : a(Y_2) \neq s_i, \\ & Y_1, Y_2 \in U(X) \end{cases} \quad (4)$$

$U(X) = \{Y : |Y - X| \leq R\}$, i.e. $U(X)$ is a circular neighborhood of X with radius R .

4.1.3 Construction of a feature vector

The final feature vector for each pixel is constructed by concatenation of superpixel features and location priors.

4.2 Classification of pixels

Random forest is a combination of tree predictors, such that each tree depends on the values of a random vector of training samples taken independently and with the same distribution for all trees in the forest. Random forest outputs the class that is the mode of the class outputs by individual trees. An output from this classifier for a sample X can be represented in a probabilistic form, similar to (3):

$$C = [P(a(X) = s_1), P(a(X) = s_2), \dots, P(a(X) = s_N)], \quad (5)$$

where

$$P(a(X) = s_i) = \frac{N_{trees : a(X) = s_i}}{N_{trees}}, \quad i = 1, \dots, N. \quad (6)$$

So, after random forest classification for each pixel we can obtain a vector of probabilities of its belonging to each of the anatomical structures.

4.3 Incorporation of the classifier output into MRF

Classification of individual pixels can produce some noise, especially near the boundaries of anatomical structures, because it does not take into account spatial relations between pixels. Introduction of Markov random field can effectively address this problem. An image can be described by a Markov network where x_j are observable variables — pixels, and t_j are latent variables — their labels.

The most probable values of latent variables are reconstructed from observable variables:

$$T_{MP} = \arg \max_T P(T|X) \quad (7)$$

The problem of finding the most probable distribution can be reduced to energy minimization problem:

$$E(T|X) = \sum_{(i,j) \in E} E_{ij}(t_i, t_j) + \sum_i E_i(x_i, t_i) \rightarrow \min, \quad t_i \in \{1, \dots, N\} \quad (8)$$

Minimization of energy by k -valued latent variables is an NP problem, nevertheless it is possible to construct an iterative procedure which approximately converges to a global optimum.

Unary term in (8) stands for a correspondence of the pixel with each of the classes. In our case it is calculated as follows:

$$E(x, t) = -\log_{10} C, \quad (9)$$

where C is the random forest output from equation (5).

Binary term in (8) reflects correlation between classes of neighboring pixels. In the present work, Potts model is used where non-coincidence of neighboring pixel classes is penalized by a constant value of 1:

$$E_{ij}(t_i, t_j) = 1 - \delta(t_i, t_j) \quad (10)$$

5. EXPERIMENTS

The described atlas-based segmentation algorithm was tested on real experimental Nissl-stained mouse brain slice images preprocessed by the proposed luminance equalization method. Results are shown in Fig. 2. It can be seen that introduction of MRF is necessary to make anatomical structures continuous.

Implementation details are given below.

Brain slice image size is 270×204 pixels. During random forest classification each pixel was represented by a vector of 75 superpixel features and 17 location priors corresponding to 17 anatomical structures. Radius of the location prior is equal to 8 pixels. Random forest classifier was constructed of 100 trees with the maximum depth of 30.

The proposed segmentation algorithm was evaluated on 10 pairs of neighboring slices from the Allen Brain Atlas. One slice from each pair was used for training and the other one, its neighbor, for testing. Pairs of slices were taken randomly from different parts of the Allen Brain Atlas. Overall precision and precision per structure averaged from 10 different pairs are presented in Table 1.

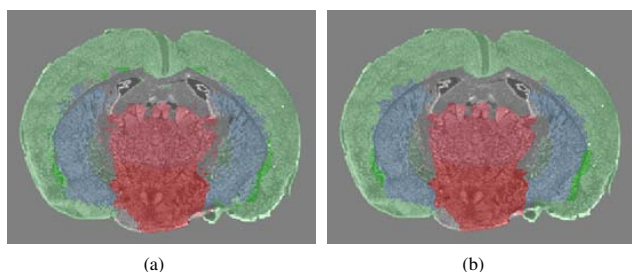


Figure 2: a: segmentation result after pixel classification with random forest; b: segmentation result after MRF refinement.

Table 1: Precision

Overall	CB _y	FT	G	CB _v
0.860589	0.902196	0.568939	0.500531	0.921488
MY	MB	P	CLX _g	6b
0.764072	0.948688	0.816145	0.929592	0.309266
HPF _g	HPF	TH	HY	STR
0.905144	0.607194	0.952606	0.867327	0.949691
PAL	CLX _v			
0.616833	0.738971			

Precision was calculated as follows. First of all, the confusion matrix for each pair of slices was calculated. It is a square matrix M whose elements m_{ij} reflect the number of pixels of i -th true anatomical structure that were (mis-)classified as j -th anatomical structure during the automated segmentation procedure. Ideally, M is a diagonal matrix, which means that all the pixels were correctly annotated. Precision for i -th anatomical structure is calculated by the formula:

$$p_i = \frac{m_{ii}}{\sum_j m_{ij}} \quad (11)$$

Overall precision for a pair of slices is calculated by the formula:

$$p = \frac{\sum_i m_{ii}}{\sum_{i,j} m_{ij}} \quad (12)$$

6. CONCLUSION AND DISCUSSION

We have proposed a method for automatic atlas-based segmentation of Nissl-stained mouse brain slice images which takes into account texture peculiarities of images obtained by this protocol. An effective technique for luminance equalization was introduced. It significantly enhances the quality of real experimental images and makes them close to Allen Brain Atlas images, which is essential for segmentation based on supervised learning.

A novel segmentation framework is introduced. Visual analysis and numerical results show that the algorithm performs rather well on large anatomical structures such as cerebral cortex (CLX_g), thalamus (TH), etc. Thin and small structures, especially with fuzzy boundaries, such as $6b$ are difficult to segment. Classification errors can occur across the boundaries of anatomical structures in cases when these boundaries are not clear or when a boundary between structures lies far from the corresponding boundary on the training brain slice. Similar problems are discussed in previous works devoted to automated segmentation of MRM brain volumes. Numerical results represented in Table 1 are comparable to those from previous works.

In the future, we will add textons (texture elements) to feature vectors in order to include local texture information, which is probably helpful for classification of anatomical structures. Shape priors will also be incorporated into the automated segmentation framework.

7. REFERENCES

- [1] Allen Brain Atlas
<http://mouse.brain-map.org/atlas/index.html>
- [2] Leo Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2004.
- [3] Lydia Ng, Michael Hawrylycz, and David Haynor, "Automated high-throughput registration for localizing 3D mouse brain gene expression using ITK," *Insight Journal*, vol. 1, 2005.
- [4] T. Riklin-Raviv, N. Sochenz, N. Kiryati, N. Ben-Zadok, S. Gefen, L. Bertand and J. Nissanov, "Propagating Distributions for Segmentation of Brain Atlas," *4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 1304–1307.
- [5] Anjum Ali, Anders Dale, Alexandra Badea and Allan Johnson, "Automated Segmentation of Neuroanatomical Structures in Multispectral MR Microscopy of the Mouse Brain," *Neuroimage*, vol. 27, no. 2, pp. 425–435, 2005.
- [6] Alize Scheenstra, Jouke Dijkstra, Rob van de Ven, Louise van der Weerd and Johan Reiber, "Automated Segmentation of the Ex Vivo Mouse Brain," *Proceedings of the SPIE*, vol. 6511, pp. 651106, 2007.
- [7] Min Bae, Rong Pan, Teresa Wu and Alexandra Badea, "Automated Segmentation of Mouse Brain Images Using Extended MRF," *Neuroimage*, vol. 46, no. 3, pp. 717–725, 2009.
- [8] Brian Fulkerson, Andrea Vedaldi and Stefano Soatto, "Class Segmentation and Object Localization with Superpixel Neighborhoods," *International Conference on Computer Vision*, pp. 670–677, 2009.
- [9] Z. Rahman, D. Jobson and G. A. Woodell, "Multiscale Retinex for color image enhancement," *Proceedings of the IEEE Int. Conf. In Image Processing*, pp. 1003–1006, 1996.
- [10] Christopher Christoudias, Bogdan Georgescu and Peter Meer, "Synergism in Low Level Vision," *International Conference on Pattern Recognition*, vol. 4, pp. 150–155, 2002.
- [11] Sylvain Paris and Fredo Durand, "Topological Approach to Hierarchical Segmentation using Mean Shift," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.
- [12] Sergey Sudakov, Olga Barinova, Alexander Velizhev and Anton Konushin, "Semantic segmentation of road images based on cascade classifiers," *Proceedings of the ISPRS XXI Congress*, pp. 601–604, 2008.

ABOUT THE AUTHORS

Olga Senyukova is a Ph.D. student at Graphics & Media Lab within the Moscow State University's CMC Department.

Her contact email is osenyukova@graphics.cs.msu.ru

Alexey Lukin, Ph.D., is a member of scientific staff at the Laboratory of Mathematical Methods of Image Processing within the Moscow State University's CMC Department.

His contact email is lukin@graphics.cs.msu.ru

Dmitry Vetrov, Ph.D., is a member of scientific staff at the Chair of Mathematical Methods of Forecasting within the Moscow State University's CMC Department.

His contact email is vetrovd@yandex.ru

Improvement of background subtraction by mask constraints

Vadim Konushin^{1,2}, Anton Konushin¹

¹Graphics and Media Lab, Moscow State Lomonosov University, Moscow, Russia

²Keldysh Institute of Applied Mathematics, Russian Academy of Sciences

E-mail: {vadim, ktosh}@graphics.cs.msu.ru

Abstract

In this paper we propose an improvement for background subtraction algorithms for specific video surveillance scenario. We consider a case, when a video camera is attached to a wall and observes people walking by or coming up to the camera. We propose 2 foreground mask models and show how to integrate these new mask constraints into common background subtraction techniques – pixel-based algorithms and methods, based on graphical models.

The proposed modifications have no parameters and add little computational overhead. Experiments, conducted on our own video sequences, demonstrate segmentation accuracy improvement of the modified algorithms.

Keywords: *Foreground extraction, Background subtraction, Markov Random Field, GraphCut.*

1. INTRODUCTION

Human-computer interfaces, based on video surveillance, are a rapidly developing field of research and have a wide range of applications, e.g. smart advertisement. Such systems can react differently based on subject's estimated relative position, gaze direction, gender, height, age, race, gestures, etc. However, an accurate estimation of these parameters are required, otherwise reaction can be awkward and confusing.

Foreground extraction is an important preprocessing step of most operations in video-based human interfaces. Many detection and tracking algorithms detect foreground blobs first and then match them to the tracked objects. Person classification and recognition algorithms could benefit from an accurate foreground mask by using the context: hair color, clothes texture, person height could bring the additional information for a more accurate classification in comparison with a common face classification.

Most foreground extraction techniques are designed for a general video surveillance scenario. In this work we concentrate on a single video surveillance scenario, when a camera is attached to a wall at 1-2m height and observes people coming up to it, see Fig. 1 (a). This is a common case for smart advertisement systems, smart kiosks, etc. For this particular surveillance scenario we propose two different foreground mask models. These models constraint background subtraction: by allowing a smaller set of possible masks, they make the whole algorithm more stable for this scenario. We show how to modify standard pixel-based algorithms, and algorithms, based on Markov Random Fields to account for this model.

The rest of the paper is organized as follows: section 2 describes current background subtraction techniques, in section 3 we define the proposed masks models. Modifications of standard background subtraction methods are given in section 4. In section 5 we report the obtained results. Section 6 concludes the paper.

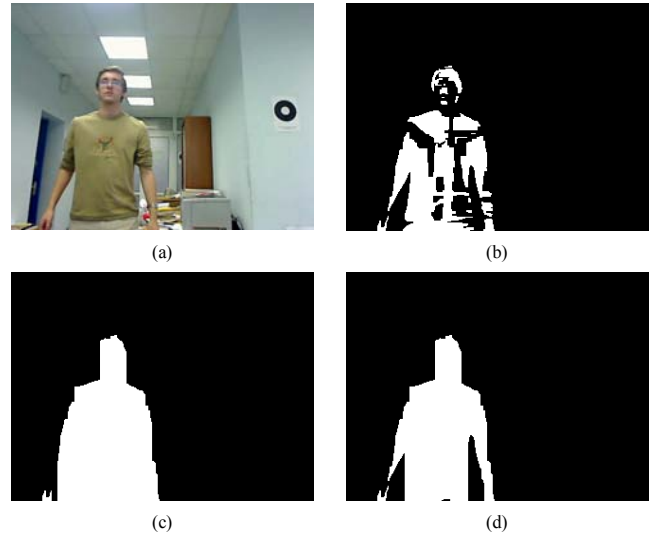


Figure 1: Example of background subtraction (a) a source frame, (b) result by an algorithm, based on Markov Random Field, (c) result of algorithm's modification, Mask Model 1, (d) result of algorithm's modification, Mask Model 2. See text for the details.

2. RELATED WORK

We can coarsely classify most background subtraction techniques into 3 classes: pixel-based [1], [2], [3], patch-based [4] and the algorithms that are based on some graphical models – Markov Random Field (MRF) and Conditional Random Field (CRF) [5], [6], [7].

Pixel-based algorithms model background color distributions for each pixel independently and for every video frame decide, whether the pixel is occluded by comparing it with a stored background model. Common examples are to model pixel color distribution with a single gaussian [1], gaussian mixture model [2] or non-parametrically [3].

Patch-based algorithms [4] model distribution of image patches. By using patches instead of single pixels, they can consider image texture in addition to pixel color, so they can be more robust to noise and in some cases to illumination change. But again, the classification into foreground/background is made independently of other patches.

The last group of the algorithms treats the whole image as a graphical model [5], [6], [7]. They fuse different features (colour, contrast, motion) by means of MRF or CRF. Then maximum a posteriori (MAP) solution is found with the use of GraphCut [8].

Most algorithms don't impose any constraints on the resulting foreground mask, so they can be applied to any video surveillance scenario, though most tests consider the security applications: camera watching a parking, a shop, an entry to some office.

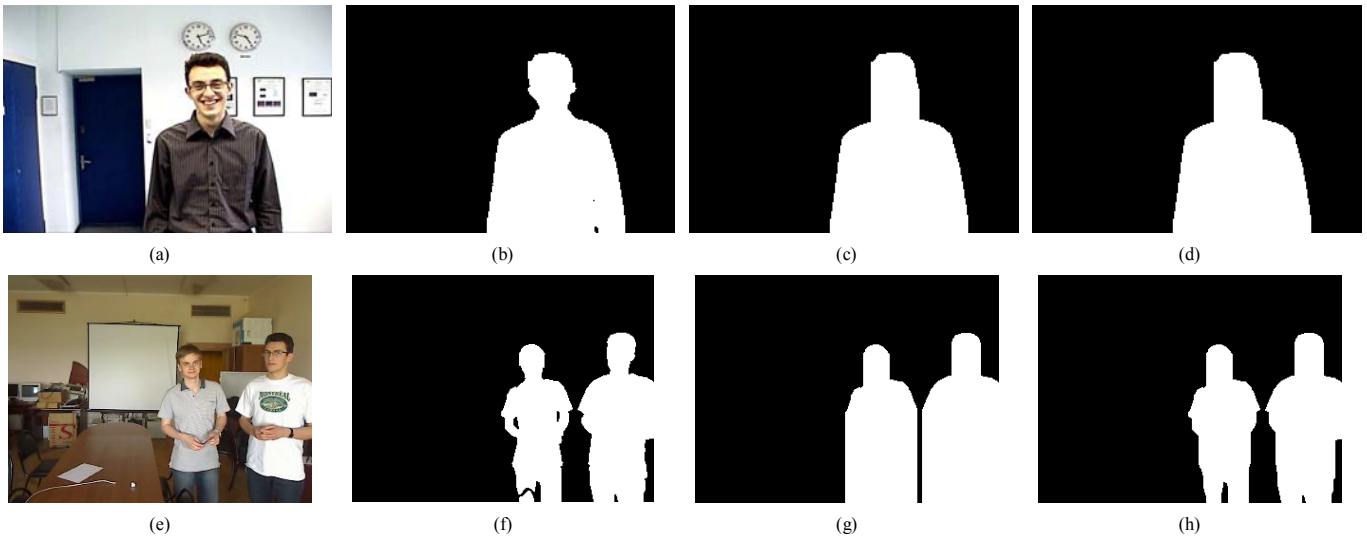


Figure 2: Mask models example. (a, e) – source frames, (b, f) – ground truth foreground masks, (c, g) – mask model 1, (d, h) – mask model 2.

Some mask constraints can be used only in a postprocessing: for choosing a radius of morphology operations or classification of resulting blobs into the objects of different classes or noise based on their size.

Some CRF-based methods [6], [7] specifically consider the case of video-chat type sequences as they are designed for possible background substitution during video conferences. They learn motion cues to differentiate a closest person from people, walking behind him during the video conference.

In this paper, we focus on a different scenario. A video-camera is attached to a wall or some stand, approximately on a human's height and records people walking by or coming up to the camera.

We propose 2 types of mask models for this scenario. We show how to integrate these mask constraints into different kinds of existing background subtraction techniques and demonstrate our results on 2 specific algorithms: a pixel-based and a MRF-based algorithm. Any other algorithm of such type or a patch-based algorithm could be modified in such way as well.

3. MASK MODEL

Let I be a video frame, $M_{x,y}$ – mask of the foreground layer, $M_{x,y} = 1$ for foreground pixels, and $M_{x,y} = 0$ for background pixels.

Considering the described scenario and the shape of objects of interest (people), we propose to limit possible foreground masks to one of the 2 following mask models:

Model 1

- **Formal definition:** for every pixel column x , there is such $h(x)$, so that $M_{x,y} = 1$ for every $y \leq h(x)$ and $M_{x,y} = 0$ for every $y > h(x)$.

So if any pixel is marked as foreground then all pixels under it must also be marked as foreground. An example of such mask is

demonstrated on Fig 1 (c), 2 (c, g). As can be seen, it is rather adequate for examples, where a person is standing or walking frontally (doesn't bend). Most errors occur in cases of bending or outstretched arms.

Model 2

- **Formal definition:** for every pixel column x , there are such $h_1(x)$ and $h_2(x)$, $h_1(x) < h_2(x)$ so that $M_{x,y} = 1$ for every $y \in [h_1(x), h_2(x)]$ and $M_{x,y} = 0$ for $y < h_1(x)$ or $y > h_2(x)$.

Examples of this model are given on Fig. 1 (d), Fig. 2 (d, h). This model places fewer constraints on the foreground mask. As can be seen in Fig 2 (h), this mask is much closer to the ground truth foreground mask, than mask of model 1. The remaining errors occur when there are more, than 2 foreground/background transitions in a single pixel column – for example near a neck (from bottom to top: shoulder-background near a neck – face – background again).

3.1 Model validation

To validate the proposed mask models we conducted the following experiments. We transformed ground truth segmentations of the 100 manually annotated frames from 20 different video sequences, so that they conformed to our 2 models. For the Model 1 we just marked every pixel under any foreground pixel as foreground, and for the Model 2 we found highest and lowest foreground pixels for every column, and marked every pixel between them as foreground.

Then we computed a number of misclassified pixels. In Table 1 we show these results compared to results, obtained by common pixel-based and MRF-based approaches.

As can be seen, number of misclassified pixels for the transformed masks is much less than for the algorithms results, which leaves us a hope, that by constraining the algorithms to produce only masks of these types we could achieve a better segmentation.

	Misclassified pixels, %
Ground truth – Model 1	2.4
Ground truth – Model 2	1.0
Pixel-based algorithm	4.3
MRF-based algorithm	4.0

Table 1. Number of misclassified pixels comparison between background subtraction results and the proposed mask models. See text for the details.

4. PROPOSED ALGORITHM

In this section we show, how to constrain pixel-based and MRF-based algorithms to produce only the proposed kind of masks.

4.1 Pixel-based algorithm

Let $p(I_{x,y} | M_{x,y})$ be a probability of pixel (x, y) to have color $I_{x,y}$, being labeled $M_{x,y}$.

Common pixel-based algorithms mark a pixel as foreground if $p(I(x, y) | 1) > p(I(x, y) | 0)$ and as background otherwise.

Now, let's consider the case of mask Model 1. In this case for every x we need to find $h(x)$ – such that all pixels above $h(x)$ belong to a background, and all pixels under $h(x)$ belong to a foreground. So we need to maximize the following likelihood:

$p(h(x) = Y) = p(I_1, \dots, I_{Y-1}, I_Y, \dots, I_H | 0, \dots, 0, 1, \dots, 1)$, where we have skipped index x , H - is a frame height, lower y-indices correspond to higher pixels. As I_1, \dots, I_H are conditionally independent of mask labels M ,

$$p(h(x) = Y) = \prod_{y < Y} p(I_y | 0) \cdot \prod_{y \geq Y} p(I_y | 1)$$

Instead of probability itself we can maximize its logarithm: $\ln p(h(x) = Y) = \sum_{y < Y} \ln p(I_y | 0) + \sum_{y \geq Y} \ln p(I_y | 1)$

For fast computation we just need to precompute following integral projection images:

$$F_{x,y} = \sum_{y \geq Y} \ln p(I_{x,y} | 1) \text{ and } B_{x,y} = \sum_{y < Y} \ln p(I_{x,y} | 0)$$

Now $\arg \max(\ln p(h(x) = Y)) = \arg \max(F(x, Y) + B(x, Y))$ and we need only to find a maximum value in each column of a precomputed image.

In case of mask Model 2, for every x we need to find 2 values $h_1(x)$ and $h_2(x)$. Now the likelihood is $p(h_1(x) = Y_1, h_2(x) = Y_2) = \prod_{y < Y_1} p(I_y | 0) \cdot \prod_{Y_1 \leq y \leq Y_2} p(I_y | 1) \cdot \prod_{y > Y_2} p(I_y | 0)$

Loglikelihood is

$$\ln p(h_1(x) = Y_1, h_2(x) = Y_2) = B(x, Y_1) + (F(x, Y_1) - F(x, Y_2 + 1)) + (B(x, H + 1) - B(Y_2 + 1))$$

We can optimize it either globally by searching through all possible pairs (Y_1, Y_2) or by searching for the best Y_1 at first (getting mask Model 1), fixing it, and searching for the best Y_2 .

As can be seen from the algorithm, this modification adds very little computational overhead. Another advantage is no additional parameters.

4.2 Constraining the algorithm, based on MRF or CRF

These algorithms minimize an energy of the form $E(M) = \sum_p \phi(I | M_p) + \sum_{\{p,q\} \in N} \phi(I | M_p, M_q)$,

where $\phi(I | M_p)$ and $\phi(I | M_p, M_q)$ are some unary and pair-wise potentials. Minimization of this energy is equivalent to maximizing a maximum a posteriori probability. This energy is usually globally minimized by means of binary graph cut.

To constrain this algorithm to produce only masks of Model 1, we perform the following:

- Add 2 artificial hidden rows of pixels (variables in the random field) to the image – one at the top of it, the other at the bottom.
- Modify unary potentials for the added rows: $\phi(I_p | 1) = Inf$ for the top row and $\phi(I_p | 0) = Inf$ for the bottom row.
- Vertical pair-wise potentials between the added rows and previous border rows is set equal to the nearest pair-wise vertical potential in the image.
- Add to all vertical pair-wise potentials some ‘big value’:
 $\phi'(I_p | M_p, M_q) = \phi(I_p | M_p, M_q) + BigValue$

Such unary potentials for the top and bottom rows ‘oblige’ the algorithm to make at least 1 cut in every image column, and modified vertical pair-wise potentials lead to maximum 1 cut in every image column. *BigValue* must be larger than sum of unary potentials for any column. So the artificial bottom row is marked as foreground, top row is marked as background, and for every column there is a single label transition – and the mask is conformed with mask Model 1.

In case of Model 2, we perform operations described above, so that the mask becomes of type 1. Then we fix the top border of the foreground mask, and perform the same operation with foreground-background mask labels switch. However, this doesn't give the global minimum.

5. RESULTS AND COMPARISON

We have tested our modifications on 2 specific algorithms:

- A pixel-based algorithm, where a pixel background color probability is modeled as a single 3d Gaussian. Foreground color model is uniform.

We have also experimented with different uniform foreground color probability values (so mathematically they are not probabilities, because they do not sum up to 1), which is equivalent to comparing background color probability to a threshold.

- MRF-based algorithm. Unary potentials are computed from the same pixel color probabilities, as in the pixel-based

algorithm. Pair-wise potential is standard for segmentations with GraphCut and uses background attenuation from [5].

For the experiments we have collected 20 video sequences, shot by a camera, placed at 1.3m – 2m above the floor level. Video resolution varies between 320x240 and 640x480, but we downsampled all video sequences to 320x240. Video recording was conducted in 3 different rooms with different illumination.

To be able to evaluate background subtraction results we have manually segmented 100 random frames into foreground and background.

In Table 2 we show the results of the experiments. For all algorithms (original and modified) we used the same parameters. It can be seen, that the proposed mask constraints improve segmentation accuracy for both algorithms. Both types of constraints decrease the algorithm precision, but increase the recall.

From Table 2 we can note, that on our data MRF-based algorithm made rather modest segmentation accuracy improvement, compared to a simple pixel-based approach. But modified versions of both algorithms showed almost similar results.

By changing foreground probability (i.e. threshold) we can make precision/recall curves for the pixel-based algorithm and its modifications (for MRF-based algorithm a dependency on this parameter is more complex). These curves are demonstrated on Fig. 3. We can see, that mask model 2 shows the best precision/recall ratio.

Algorithm	Misclassified pixels, %	Precision, %	Recall %
Pixel-based	4.3	96.4	80.5
Pixel-based, Mask Model 1	3.1	88.9	93.4
Pixel-based, Mask Model 2	2.9	94.7	88.6
MRF-based	4.0	96.8	81.4
MRF-based, Mask Model 1	3.0	90.2	92.8
MRF-based, Mask Model 2	3.4	95.4	84.3

Table 2. Mask accuracy comparison between original and the modified algorithms.

6. CONCLUSION

In this paper, we have considered a special case of video surveillance scenario, where a camera is attached to a wall on 1-2m height and records people walking by or coming up to the camera. We proposed 2 special foreground mask models for this scenario and showed how to integrate new mask constraints into typical background subtraction algorithms. The proposed modifications have no parameters and add little computational overhead. Experiments, conducted on our own video sequences, demonstrated segmentation accuracy improvement of the modified algorithms.

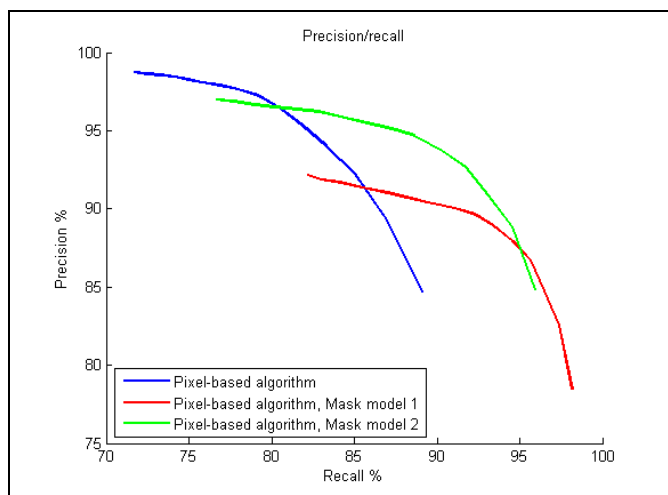


Figure 3. Precision/recall curves for the original pixel-based background subtraction algorithm and its 2 modifications. See text for the details.

7. ACKNOWLEDGEMENTS

This work was supported by Russian Foundation for Basic Research, project N. 09-01-92474-MNKS_a, Presidential Grant for the Support of Leading Scientific Schools N. NSh-8129.2010.9

8. REFERENCES

- [1] C. Wren, A. Azarbayejani, T. Darrel, and A. Pentland, Pfunder: Real time tracking of the human body. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19 (7), pp. 780–785, 1997
- [2] C. Stauffer and W. E. L. Grimson, Adaptive background mixture models for real-time tracking. In *Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 246–252, 1999
- [3] A. Elgammal, D. Harwood, and L. Davis, Non-Parametric Model for Background Subtraction, In *Proc. Sixth European Conf. Computer Vision (ECCV)*, 2, pp. 751-767, 2000
- [4] M. Heikkila, M. Pietikainen, J. Heikkila, A texture-based method for detecting moving objects, In *British Machine Vision Conference (BMVC)*, 2004
- [5] J. Sun, W. Zhang, X. Tang, and H. Y. Shum, Background cut, In *Proc. Europ. Conf. on Computer Vision (ECCV)*, pp. 628-641, 2006
- [6] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov, Bilyer segmentation of live video, In *Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 53–60, 2006
- [7] P. Yin, A. Criminisi, J. Winn, and I. Essa, Tree-based classifiers for bilyer video segmentation, In *Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007
- [8] Y. Boykov and M. Pi. Jolly, Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *Proceedings of ICCV*, pp. 105–112, 2001

Automatic Extraction of Regular Grids from Rectified Facade Image

Anton Yakubenko, Ivan Mizin, Anton Konushin
 Graphics & Media Lab, Department of Computational Mathematics and Cybernetics
 Lomonosov Moscow State University, Moscow, Russia
 {toh, ivan.mizin, ktosh}@graphics.cs.msu.ru



Figure 1: Input rectified image of building's facade (left) and automatically recognized structure, presented as a set of non-overlapping grids of similar windows (right).

Abstract

Automation of facade images interpretation is a crucial task for efficient 3D modeling of urban areas. Information about regularities in facade images can be used for image completion of texture areas occluded with foreground objects, 3D model compression and texture quality enhancement. Existing approaches cannot handle the whole variety of buildings facades and often fail in cases of occluded textures. We present a novel algorithm for automatic extraction of facade structure from single rectified image. We describe facade structure as a set of non-overlapping grids of similar facade parts, usually containing windows. The most frequent possible grid cells sizes and grid cells centers are estimated by detecting and analyzing similar rectangles in input image. Greedy iterative algorithm sequentially chooses grids with maximum quality measure that incorporates empirical observations about similarities and symmetries. Comparison on the database, created from publically available sources with additional complex examples from our photos, shows that the proposed algorithm outperforms two other state-of-the-art methods and can handle occlusions.

Keywords: *Facade, Structure, Regularity, Grid, Lattice, Urban, 3D City Map, Image-based, Rectified, Texture, Single-view, Occlusion, Recognition, Reconstruction.*

1. INTRODUCTION

Efficient mass-production of 3D cities maps is one of the topical problems at the junction of geo informatics and computer vision. 3D maps are natural evolution of ordinary flat 2D maps. They can be used in a variety of common geo informational tasks, like navigation, city planning, etc., and also afford new opportunities for virtual and augmented reality, location-based services, etc.

3D maps topic has gained a lot of attention in last few years. Common Internet users use 3D geo information web services like Google Maps [36] and Bing Maps [37]. Industrial software leaders like Autodesk develop special products for 3D city maps management like LandXplorer [38]. Joint community develops

CityGML [39] standard for the modeling and exchange of 3D city and landscape models.

3D models of urban buildings are the main content for 3D geo information systems. There are multiple ways for acquisition of these models. They can be created from sparse terrestrial photos [3, 40] automatically or manually using photogrammetric and 3D modeling software. These geometry modeling tools are usually coupled with Adobe Photoshop or similar photo editing software for texture editing. More automated approaches include processing of point clouds [4] and video streams [5] from land based mobile mapping systems. Aerial imagery and aerial LIDAR point clouds can also be used as source data [6].

3D model of urban building can be roughly divided in two parts – roof and facades. Roofs can be modeled from aerial imagery [7] or LIDAR point clouds using manual tools or by automatic planes fitting or template matching. Reconstruction of building facades is a more complicated task. But buildings facades frequently have structure and regularity. This semantic knowledge can help solving a number of problems arising in facade modeling, processing and interpretation.

Facade textures are often occluded by foreground objects like vegetation, advertisement billboards, electrical cables, other buildings, bulging parts of the building, people, cars, etc. Occlusion areas can be segmented and reconstructed [8, 13] by comparing similar parts of the facade. Textures created from aerial views often lack quality and resolution on the lower floors of the building. If building's floor were extracted from image these textures can be enhanced [9] by transferring texture information from higher floors to the lower.

Detected facade elements, like windows and balconies, can be replaced by more detailed template 3D models from a database for overall visual quality improvement [10]. Knowledge of repeated texture parts and according geometry elements can be used to compress 3D model geometry and texture without visible loss of quality [11] that is required in web applications. Facade structure information can be also used for image-based geo locating [12], and other tasks.

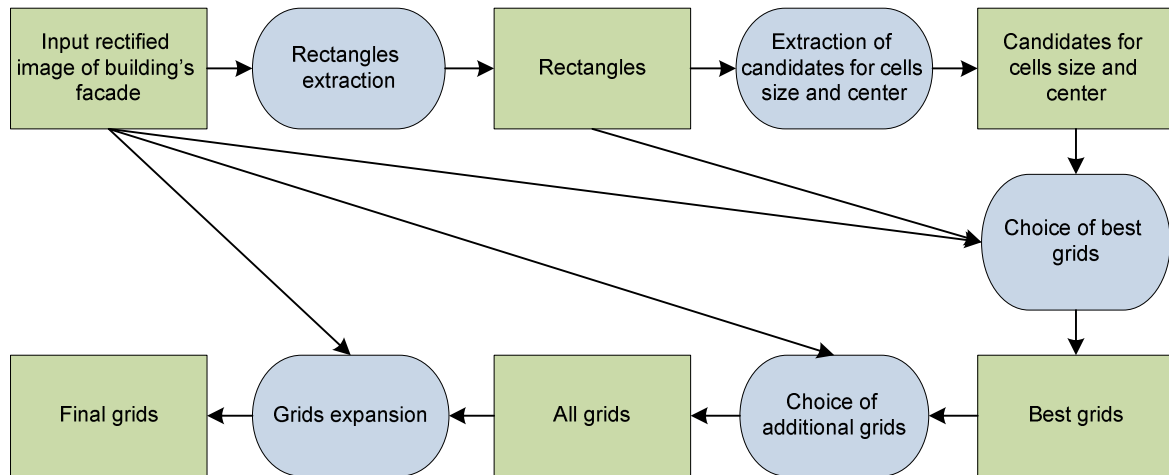


Figure 2: Scheme of the proposed algorithm. Green rectangles show data flow. Blue rounded rectangles show algorithm's parts.

The paper is organized as follows. Brief analysis of the related work is given in Section 2. We present a novel method for automatic determining of a set of non-overlapping grids of similar rectangular facade parts in a given single rectified image of urban building's facade in Section 3. Results of our algorithm and comparison with two state-of-the-art methods [1, 2] results are given in Section 4. Finally, we give the conclusions and discussion in Section 5.

2. RELATED WORK

Repeatability and regularity are the main cues to deal with in facade interpretation tasks. They can be frequently met in man-made environments, not only in urban scenes. The classic solution for extraction of regularity from a single image of a plane object without perspective distortions is analysis of autocorrelation function peaks [19]. This approach treats image globally and works only if regularity is strict, occupies large part of image and there are almost no occlusions.

To deal with near-regular textures, perspective distortions and partial occlusions sparse set of image patches can be analyzed. These patches can be sampled randomly [20] or chosen by interest point [1] or rectangles detector [8]. Similar patches can be grouped together to fit homography [12], grid [8, 22], near-regular lattice [1, 20] or more complex models [2]. Patches are compared using metrics like SSD (sum of squared distances) [21] or NCC (normalized cross-correlation) [1, 22], or by matching descriptors like SIFT [12]. Depending on the complexity of model it can be estimated using Hough transform [23], RANSAC [12], iterative lattice spreading [1] or MRF (Markov random field) optimization [8]. These approaches have shown good results on a variety of images. Due to general problem formulation and using only small local patches for analysis results of these algorithms dramatically deteriorate on facades with complex look of similar parts, for example older buildings, several regular grids, large occlusions.

Approaches, mentioned above do not focus on facade images particularly. A group of methods aims to detect windows, which frequently form the facade structure. This can be done by training Viola-Jones like detector using Haar features and SVM ([24] or AdaBoost [25]). Windows differ greatly from each other, thus these approaches work acceptably for small databases of similar images. To deal with this problem incremental learning approaches were suggested [26], but they require user interaction.

Heuristic approaches make use of similarity of windows corners [27] and presence of gradients on windows perimeter [28]. Akaike's information criterion [27] or minimum description length [2] can be used to consider rows and columns regularities. Windows detection can be also formulated as optimal labeling problem [29] or as problem of finding optimal non-overlapping subset or rectangles [30]. Current windows detection results do not allow using them as final result. But detected windows can be used as lower level information for more intelligent algorithms, which will take regularities into account. General semantic image segmentation [31] and wall area extraction [32] results are also too far from desirable.

State-of-the-art algorithms in automatic facade interpretation try to describe input rectified facade image as a higher level model. This model can be presented as partition of facade image into floors, tiles, windows, etc. More generally this partition can be described as grammar inference [9, 33]. The choice of grammar rules (for example vertical or horizontal splits) and their parameters (coordinates of splits) can be done by reversible jump Markov Chain Monte Carlo (rjMCMC) sampling [33] and comparing neighboring candidates, for example, using mutual information and edges inside and between candidate patches [9]. Metric information can also reduce parameter space [9]. This is one of the most promising approaches, but current results can hardly handle occlusions and strongly depend on the grammar rules formulation and inference criteria.

3. PROPOSED METHOD

We take rectified image of building's facade as input. In assumption of plane facade, rectified image (Figures 3, 4) compensates all projective distortions. This image can be acquired directly from the already created 3D model as its texture or from input photo. The latter can be done by either manually selecting four points in photo, which correspond to a rectangle in the 3D scene, or by automatic algorithms [14], which find lines in image and group them according to vanishing points.

Our aim was to develop algorithm which would work properly on a wide variety of rectified facade images with occlusions. Thus it should use not very strict model. We have chosen a set of non-overlapping grids of similar rectangular image parts (Figure 1) as a simple and flexible model, which describes the absolute majority of urban buildings. Grid lines are parallel to image axes.

Thus each grid is described by 6 parameters:

(cx, cy) – X and Y coordinates of the top left corner of the grid,

(sx, sy) – cells size,

$(rows, columns)$ – number of rows and columns in the grid.

The scheme of the proposed algorithm is shown in Figure 2. We find rectangles in input image (Section 3.1). We choose the most frequent distances between them as possible grid cells size (sx, sy) . In assumption that some of the rectangles are windows we remember their centers as possible grid cells centers. In combination with grid cell size this information reduces number of possible (cx, cy) values. Thus we can significantly reduce the total number of grids for further analysis (Section 3.2). We sequentially choose the best grid non-overlapping with already extracted grids considering neighboring cells similarity, cells symmetry and existence of rectangle in the middle of cells (Section 3.3). Due to sampling and grid quality constraints some grids can be missed. We search for additional grids with cells, which are similar to already extracted grids (Section 3.4). We then expand extracted grids using weaker thresholds to handle occlusions (Section 3.5). Algorithm's stages are described in details below.



Figure 3: Input image.



Figure 4: Rectified image.

3.1 Rectangles extraction

Existing approaches like Hough transform [34] can be used for extraction of rectangles with sides parallel to image sides. But we can extract even better rectangles. We build image pyramid by performing image downsampling with factor 2 and factor 4. We perform separate vertical and horizontal edge detection for each image in the pyramid by using Sobel operator. Pixel in the original image is labeled as edge pixel only if it is edge pixel in all the images of image pyramid. This approach filters out small and noisy edges, including undesirable edges between regular structures like wall bricks (Figure 5).

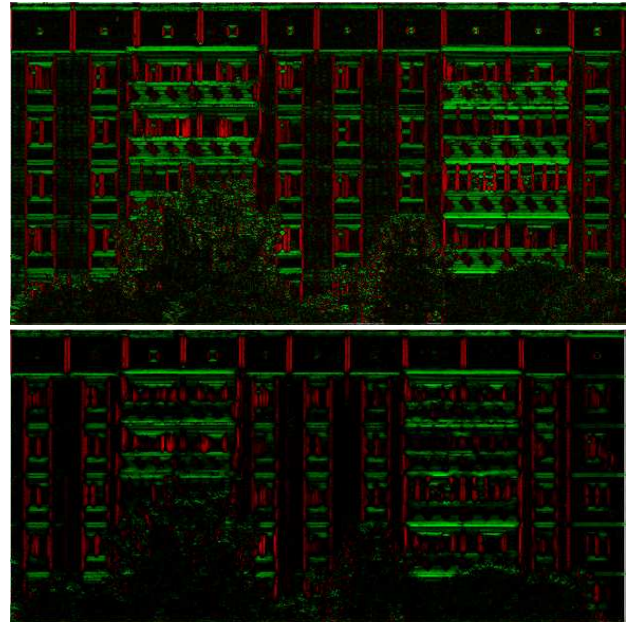


Figure 5: Extracted vertical (red) and horizontal (green) edges applied only to original resolution (top) and applied to image pyramid (bottom).

If pixel has enough horizontal edges in its right neighborhood and enough vertical edges in its bottom neighborhood, it is labeled as top left corner candidate pixel. Other three corners types can be processed in the same way. We iterate through all pairs of corners. If this pair can form a rectangle, i.e. corners lie on the same horizontal or vertical line, and have proper type (for example, are “top left” and “bottom left” corner accordingly), the distance between the corners is stored. There are about ten thousand of such corners in current example. We can then perform exhaustive search of rectangles with selected corners and sizes, which form the resulting set of rectangles (Figure 6). There are about thousand rectangles in current example.



Figure 6: Some of the extracted rectangles.

3.2 Extraction of candidates for cells size and center

We compare each pair of image areas corresponding to rectangles with equal sizes using normalized cross-correlation (NCC) and store the distances between them if NCC value is large enough. The most frequent distances are selected as candidates for cells sizes (Figure 7).



Figure 7: Extracted most frequent distances between rectangles.

Resulting grids are naturally looking if its cells have facade element, for example, a window, in their center. Such centering of grids is not obligatory, but it reduces possible grids locations and avoids losing of a single side row or column near image bounds due to large grid shift.

3.3 Greedy iterative choice of best grids

The number of grids with previously determined sizes and centers can be analyzed exhaustively. Grid quality G is computed as follows:

$$G = (M * N)^{-t_1} \sum_{i=1}^M \sum_{j=1}^N Q_{i,j},$$

Where M is the number of rows in the grid, N is the number of columns in the grid, $t_1 = -0.8$ is a regularizing parameter that penalties too small grids and $Q_{i,j}$ is quality measure for cell $c_{i,j}$:

$$Q_{i,j} = \text{Similarity}(c_{i,j}, c_{i-1,j}) + \text{Simimilarity}(c_{i,j}, c_{i,j-1}) + \text{Symmetry}(c_{i,j}) + \text{HasRect}(c_{i,j}).$$

$\text{Similarity}(c_{i,j}, c_{k,l})$ is similarity measure between cells $c_{i,j}$ and $c_{k,l}$. It is equal to NCC of corresponding image patches if it is larger than threshold t_2 , or $-t_2$ otherwise. $\text{Symmetry}(c_{i,j})$ is $c_{i,j}$ cell's measure of symmetry along horizontal direction. It is equal to NCC of left and right parts of the cell if it is larger than threshold t_2 , or $-t_2$ otherwise. $\text{HasRect}(c_{i,j})$ is equal to t_2 if there is a rect inside $c_{i,j}$ cell and $-t_2$ otherwise. We took $t_2 = 0.4$ in all of our experiments. Thresholds t_1 and t_2 were selected empirically. One can vary thresholds values in parts cell quality terms.

We sequentially choose the best grid that does not overlap already extracted grids (Figure 8). We stop when new grid quality is much lower than the quality of the best grid.

3.4 Search for additional grids

In some cases some grids are missed. So we perform a search for additional grids (Figures 10, 12). We make an assumption, that they are similar to already extracted grid. We compute median cell for each already extracted grid and correlate it with the part of the image not covered by grids. If the maximum of correlation map is relatively large we start a new grid with the center in the point of this maximum. We iterate this algorithm until no new grids can be found.

3.5 Grids expansion

Extracted grids do not usually spread over the occluded regions. For each grid we analyze four grids which are created from the original grid by adding single row or column at its side. If this grid does not intersect other grids and new cells are similar to their neighbors with a lower threshold, then we replace original grid with the expanded one (Figures 9, 11, 13, 14). We apply this algorithm several times. First it is applied after best grids are

chosen to join partly occluded cells to them before searching for additional grids (Figure 9). Then it is applied after any new additional grid has been found (Figures 11, 13). Finally it is applied with extremely low threshold at the very end to recover much occluded cells (Figure 14).



Figure 8: Best grid.



Figure 9: Expansion of best grid.



Figure 10: Start of additional grid.



Figure 11: Expansion of additional grid.



Figure 12: Starting another additional grid.



Figure 13: Expansion of additional grid.



Figure 14: Final expansion to occluded areas.

4. EVALUATION

4.1 Database

Unfortunately, there is neither conventional metrology nor at least image database for testing and comparison of facade interpretation algorithms. We have chosen 25 images from Facade Data Base Vienna [15], eTRIMS Image Database [16], ZuBuD Image Database [17], CGTextures.com [18] web portal, and our own photos. Though it is a small database, but it is representative, including images of old and modern architecture, built of various materials, with different numbers of floors and tiles, different regularities types, taken under various lighting conditions. All images were manually rectified.

4.2 Competitors

Problem statement notably differs among various scientific papers. We have chosen [1] as a modern representative of algorithms for recovering general regularities. It is based on interest points extraction and comparison. Similar shifts between similar points are stored. Frequent shifts become hypotheses for starting lattice size. Lattice is build iteratively by adding new

nodes, which are similar to existing ones and met lattice size condition. Some parts of this algorithms inspired extraction of candidates for cells sizes and centers positions in our algorithm.

Algorithm [2] represents the trend of using more complex model for facade description as a linear combination of axis parallel basis translations and the appropriate coefficients and uses MDL principle for regularization. This algorithm extracts SIFT features and get the pairs of similar features. Best parameters of the model move similar points to each other.

4.3 Grid cells sizes extraction evaluation

Competitor methods do not try to center grids relatively to windows, thus they can lose some rows or columns. We evaluate grids location and number of cells in Section 4.4. In this section we evaluate only the results of cells sizes extraction. The proposed algorithm finds right grids cells sizes among 10 most frequent steps for all images in test database. But some of proper grids are missed in the final result. We consider result of algorithm right if there is a grid with proper cells size in the final set of grids. Results of grids steps extraction evaluation are shown in Figure 15.

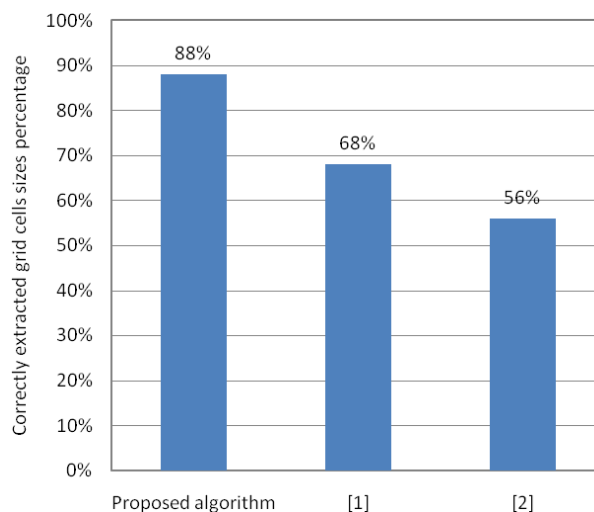


Figure 15: Grid cells size extraction evaluation results.

4.4 Evaluation of percentage of correctly extracted similar elements

We consider extracted grid to be correct if it has visually similar cells, which cannot be divided further, and has right number of rows and columns. Local shifts of grids are not taken into account, because it is not vital for further processing of extracted information, for instance, for performing texture completion. But these shifts can influence the number of rows and columns in the extracted grid. Total number of cells in each image is known. For evaluation we can count ratio of number of correctly extracted cells to total number of cells (see Figure 16). The results are shown in Figure 17. We do not provide comparison to [2], because it requires manual setting of starting point for grid positioning.



Figure 16: Example of [1] results. Ground truth grid contains 15 cells; this grid contains only 12 cells. It means that 80% of cells were detected correctly.

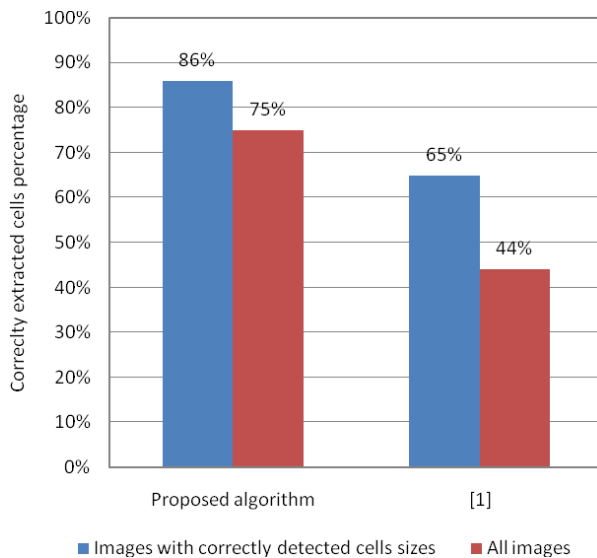


Figure 17: Evaluation of percentage of correctly extracted similar elements.

5. CONCLUSIONS AND DISCUSSION

We have proposed a novel fully automatic algorithm for extraction of information about repeatability of similar rectangular image patches from a single rectified image of urban building's facade. Comparison with two competing algorithms showed advantages of our method.

This can be first of all explained by focusing only on rectified facade images, but not a general class of images with more complex repeatability models and image distortions. In contrast to sophisticated grammar-based models we have introduced a simple model for facade description – a set of non-overlapping grids, which can describe the majority of buildings facades. Proposed model quality measure incorporates empirical knowledge about building's facade, including similarity between neighboring cells, horizontal grid symmetry and existence of rectangle in the center of each cell.

We make use of analyzing rectangles, which can be frequently met in facade images and showed to be more robust than feature points or random patches.

We reduce possible number of grid cells size by gathering the most frequent distances between extracted rectangles. Rectangles positions give us candidates for possible grid position. Due to such sampling of parameters space we can use exhaustive search for best grids. The proposed method can handle occlusions by lowering thresholds for grid quality in grid spreading process.

Though current results are promising (75% of cells are extracted correctly) there are still some problems left. Due to iterative greedy choice of best grids some grids are missed and not all occlusions are handled. Also a grid divided by another grid is considered as two separate grids. Global optimization formulation could do better. Rectangles are rather robust and easy to find, but additional detection of windows, based on machine learning techniques, can improve the overall robustness. We assume facade and facade elements to be almost flat. Bulging balconies and deeply sunken windows requires 3D analysis [35]. Non-overlapping grids model is enough for texture completion and compression tasks. But it can be enhanced by adding hierarchy and more complex models, for example, repeated pairs of elements. Higher level semantic information could be also extracted for more intelligent tasks.

6. ACKNOWLEDGMENTS

The work was partially supported by the Federal Target Program "Scientific and scientific-pedagogical personnel of innovative Russia" and Russian Foundation for Basic Research grants 09-01-92470-MHHC_a and 08-01-00883-a.

7. REFERENCES

- [1] Minwoo Park, Robert T. Collins, and Yanxi Liu "Deformed Lattice Discovery Via Efficient Mean-Shift Belief Propagation", Proc. of the 10th European Conf. Computer Vision (ECCV), Vol. 2, pp. 474 – 485, 2008.
- [2] S. Wenzel, M. Drauschke, W. Forstner "Detection of Repeated Structures in Facade Images", Pattern Recognition and Image Analysis, Vol. 18, No. 3, pp. 406 – 411, 2008.
- [3] F Remondino, S El-Hakim "Image-based 3D modelling: a review" Photogrammetric Record, Vol. 21, No. 115, pp. 269 – 291, 2006.
- [4] C. Frueh, A. Zakhor "An Automated Method for Large-Scale, Ground-Based City Model Acquisition" International Journal of Computer Vision, Vol. 60, No. 1, pp. 5 – 24, October 2004.
- [5] Nico Cornelis, Bastian Leibe, Kurt Cornelis, Luc Van Gool "3D Urban Scene Modeling Integrating Recognition and Reconstruction" International Journal Of Computer Vision, Vol. 78, No. 2 – 3, pp. 121 – 141, 2008.
- [6] J. Hu, S. You, U. Neumann, K.K. Park "Building modeling from LIDAR and Aerial Imagery" ASPRS'04, Denver, Colorado, USA. May 23-28, 2004.
- [7] Theo Moons, David Frere, Jan Vandekerckhove, Luc Van Gool "Automatic Modelling and 3D Reconstruction of Urban House Roofs from High Resolution Aerial Imagery" In Proc. 5th European Conference on Computer Vision (ECCV), 1998, Freiburg, Germany, pp. 410–425.
- [8] T. Korah, C. Rasmussen "Analysis of Building Textures for Reconstructing Partially Occluded Facades", Proc. of the 10th European Conference on Computer Vision (ECCV), Part I, pp. 359–372, 2008.
- [9] P. Muller, G. Zeng, P. Wonka, and L. V. Gool. "Image-based Procedural Modeling of Facades", Proceedings of ACM

- SIGGRAPH'07, ACM Transactions on Graphics. 2007. Vol 26. N. 3. P. 1 – 9.
- [10] Julien Ricard, Jerome Royan, Olivier Aubault “Visualization of Real Cities Based on Procedural Modeling”, IEEE Virtual Reality Workshop on Virtual Cityscapes: Key Research Issues in Modeling Large-Scale Immersive Urban Environments, 2008.
- [11] Huamin Wang, Yonatan Wexler, Eyal Ofek, Hugues Hoppe “Factoring Repeated Content Within and Among Images”, ACM Transactions on Graphics, vol. 27, no. 3, pp. 1–10, 2008.
- [12] Schindler G., Krishnamurthy P., Lublinerman R., Liu Y., Dellaert F. “Detecting and Matching Repeated Patterns for Automatic Geo-tagging in Urban Environments” Proceedings of CVPR 2008, pp. 1 – 7.
- [13] Vadim Konushin, Vladimir Vezhnevets “Automatic building texture completion”, GraphiCon'2007.
- [14] Liebowitz D., Criminisi A., Zisserman A. “Creating Architectural Models from Images”, In Proc. EuroGraphics, Vol. 18, pp. 39–50, 1999.
- [15] Facade Data Base Vienna
<http://info.tuwien.ac.at/ingeo/research/FdbV/index.htm>
- [16] eTRIMS Image Database
http://www.ipb.uni-bonn.de/projects/etrimis_db/
- [17] ZuBuD Image Database
<http://www.vision.ee.ethz.ch/showroom/zubud/index.en.html>
- [18] <http://cgtextures.com>
- [19] Hsin-Chih Lin, Ling-Ling Wang, and Shi-Nine Yang “Extracting periodicity of a regular texture based on autocorrelation functions” Pattern Recognition Letters, Vol. 18, pp. 433 – 443, 1997.
- [20] James Hays, Marius Leordeanu, Alexei A. Efros, Yanxi Liu “Discovering Texture Regularity as a Higher-Order Correspondence Problem”, In ECCV, Vol. 2, pp. 522 – 535, 2006.
- [21] Thomas Leung, Jitendra Malik “Detecting, localizing and grouping repeated scene elements from an image”, In Proc. of European Conf. Computer Vision (ECCV), pp. 546 - 555, 1996.
- [22] Frederik Schaffalitzky, Andrew Zisserman “Geometric Grouping of Repeated Elements within Images”, In Proc. of BMVC, pp. 13 - 22, 1998.
- [23] Andreas Turina, Tinne Tuytelaars, Theo Moons, Luc Van Gool “Grouping via the Matching of Repeated Patterns”, In Proc. of CAPR, pp. 250 – 259, 2001.
- [24] Bjorn Johansson, Fredrik Kahl “Detecting Windows in City Scenes”, ICPR 2002.
- [25] Haider Ali, Christin Seifert, Nitin Jindal, Lucas Paletta, Gerhard Paar “Window Detection in Facades”, International Conference on Image Analysis and Processing (ICIAP), pp. 837-842, 2007.
- [26] Susanne Wenzel, Wolfgang Forstner “Semi-Supervised Incremental Learning of Hierarchical Appearance Models”, 21st Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS), Beijing, China, 2008.
- [27] Helmut Mayer, Sergiy Reznik “Implicit Shape Models, Model Selection, and Plane Sweeping For 3D Facade Interpretation”, Photogrammetric Image Analysis (PIA) 2007.
- [28] Sung Chun Lee, Ram Nevatia “Extraction and Integration of Window in a 3D Building Model from Ground View Images”, In CVPR, IEEE Computer Society, pp. 113 – 120, 2004.
- [29] Jan Cech, Radim Sara “Windowpane Detection based on Maximum A Posteriori Probability Labeling”, In International Workshop on Combinatorial Image Analysis (IWCIA), pp. 3–11, 2008.
- [30] Filip Korc, Wolfgang Forstner “Finding Optimal Non-Overlapping Subset of Extracted Image Objects”, 2008.
- [31] Floraine Grabler “Parsing Images of Architectural Scenes”, In ICCV, pp. 44 – 57, 2007.
- [32] R. G. Laycock, A. M. Day “Towards the Automatic Integration of Ground Level Images into a Virtual Urban Environment”, WSCG 2004.
- [33] Nora Ripperda, Claus Brenner “Reconstruction of Facade Structures Using a Formal Grammar and RJMCMC”, in Pattern Recognition, pp. 750–759, 2006.
- [34] Claudio Rosito Jung, Rodrigo Schramm “Rectangle Detection based on a Windowed Hough Transform” Proceedings of the XVII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'04), pp. 17-20.
- [35] Luc Van Gool, Gang Zeng, Filip Van den Borre, Pascal Muller “Towards Mass-Produced Buildings Models”, Photogrammetric Image Analysis (PIA) 2007.
- [36] Google Maps <http://maps.google.com>
- [37] Bing Maps <http://www.bing.com/maps/>
- [38] Autodesk LandXplorer
<http://usa.autodesk.com/adsk/servlet/pc/index?siteID=123112&id=12561246>
- [39] CityGML http://www.citygmlwiki.org/index.php/Main_Page
- [40] Autodesk Photofly
<http://labs.autodesk.com/technologies/photofly/>

Generalized gamma mixtures for supervised SAR image classification

Vladimir A. Krylov
Faculty of Computational Mathematics and Cybernetics,
Lomonosov Moscow State University,
vkrylov@cs.msu.ru

Josiane Zerubia
EPI Ariana, INRIA/CNRS/UNSA,
INRIA Sophia Antipolis, France,
Josiane.Zerubia@sophia.inria.fr.

Abstract

In this paper we develop a new statistical model for supervised classification of high resolution synthetic aperture radar (SAR) amplitude images. This model is based on the recently proposed generalized gamma distribution (G Γ D) for statistics of amplitude SAR images. In order to improve the fit of G Γ D when dealing with inherently heterogeneous high resolution SAR imagery, we model the statistics of thematic classes as mixtures of G Γ D. This enables to consider not homogeneous thematic classes, which is an often requirement in practice. We complete the developed method by proving the identifiability of the developed G Γ D finite mixture model and the consistency of the involved parameter estimation scheme (method of log-cumulants) for G Γ D, which renders the developed approach mathematically correct. In order to improve the computational performance of the G Γ D mixture estimation we suggest the use of an approximative solution of the equations involved, thus, avoiding time-consuming iterative processes. The accuracy of the developed approach is validated on a high resolution TerraSAR-X image and compared to related finite mixture-based SAR classification techniques.

Keywords: *Generalized gamma distribution, finite mixtures, identifiability, method of log-cumulants, consistency, synthetic aperture radar, supervised classification, Markov random field.*

1. INTRODUCTION

The recent progress in sensor and antenna construction enabled the remotely sensed satellite imagery to become widely available and to find its daily applications in fields such as: ecology, meteorology, oceanography, cartography, natural risk management and many others. After optical data, the second most used type of remote sensing imagery is the synthetic aperture radar (SAR) imagery. It has some very important advantages, such as insensitivity to Sun-illumination and meteorological conditions [1]. Yet the SAR image processing has some challenges, which are not faced when dealing with optical data. This is due to the speckle noise, which is an inherent phenomenon of the microwave data [1]. In case of the high resolution (HR) SAR, which can be up to 1 m/pixel resolution, the appreciation of details and statistics are heavily affected by speckle.

In this paper we develop an accurate and fast statistical model to deal with one of the basic SAR image processing applications - classification. The purpose of statistical modeling is to get an accurate and concise probability density function (pdf) model that estimates the statistics of a SAR image. In the literature a number of pdf models have been suggested, see in details in [1] and [2], for this purpose, yet in heterogeneous HR case most of them fail. The only solution is to consider mixture-based models and some successful attempts have been reported [2, 3]. In this paper we approach this problem by considering a SAR amplitude pdf as a finite mixture of generalized gamma distributions (G Γ D). This is a very flexible family of pdfs [4] that has already found its applications in fields like speech signal processing [5], health and economical applications [6]. G Γ D has also been considered for

SAR amplitude statistics in a recent work [7]. Here we focus on the mixtures of G Γ D, that have not been studied so far mostly due to parameter estimation problems [5].

The paper is organized as follows. In Section 2 we develop the finite G Γ D mixture model, state its identifiability (with the outline of the proof in Appendix A), the consistency of the parameter estimation scheme involved (outline of the proof in Appendix B) and suggest an approximative solution of the equations involved (derivation in Appendix C). In Section 3 we merge the statistical pdf model with a Markov random field model to get a contextual SAR classification approach robust with respect to speckle. Section 4 reports the experiments on a high resolution TerraSAR-X image and comparison to related finite mixture-based SAR classification techniques. In Section 5 the conclusions are drawn.

2. GENERALIZED GAMMA MIXTURES

The pdf of the G Γ D takes the following form:

$$\mathcal{G}(r|\nu, \kappa, \sigma) = \frac{\nu}{\sigma\Gamma(\kappa)} \left(\frac{r}{\sigma}\right)^{\kappa\nu-1} \exp\left[-\left(\frac{r}{\sigma}\right)^\nu\right], \quad r \geq 0, \quad (1)$$

where ν, κ and σ are positive parameters corresponding to the power, shape and scale, respectively, and $\Gamma(\cdot)$ is the gamma function [8]. The G Γ D is a very flexible family of distributions: it includes Gamma, exponential, χ^2 , Nakagami, Half-normal, Rayleigh, Maxwell, Weibull, Lévy distributions as special cases and lognormal as an asymptotic case [4, 7].

The use of G Γ D to amplitude (intensity) statistics of SAR has been recently suggested by Li et. al [7] and reported good results for HR SAR. Yet the problem of mixed multimodal SAR statistics can not be dealt with even with such a flexible pdf. Thus, in order to take into account this heterogeneity scenario we propose to use a finite mixture model [9] for the distribution of grey levels. We assume an amplitude SAR image \mathcal{I} to be a set $\{r_1, \dots, r_N\}$ of N independent samples drawn from a G Γ D mixture pdf with K components:

$$p(r) = \sum_{i=1}^K \alpha_i \mathcal{G}(r|\nu_i, \kappa_i, \sigma_i), \quad r \geq 0, \quad (2)$$

with mixing proportions $\sum_{i=1}^K \alpha_i = 1$, and $\forall i : \alpha_i \in (0, 1)$.

The problem of mixture (2) estimation is equivalent to the estimation of \hat{K} and $\{\hat{\alpha}_i, \hat{\nu}_i, \hat{\kappa}_i, \hat{\sigma}_i\}_{i=1}^{\hat{K}}$. In order for this finite mixture estimation problem to be well-posed we show that the following statement holds.

Theorem 1. Finite mixtures of G Γ D are identifiable, i.e. if there are two finite G Γ D mixtures:

$$H_1 = \sum_{i=1}^{k_1} \alpha_{1i} \mathcal{G}_{1i}, \quad H_2 = \sum_{i=1}^{k_2} \alpha_{2i} \mathcal{G}_{2i},$$

$\mathcal{G}_{ai} \equiv \mathcal{G}_{aj}$, i.e. $(\nu_{ai}, \kappa_{ai}, \sigma_{ai}) = (\nu_{aj}, \kappa_{aj}, \sigma_{aj}) \Leftrightarrow i = j$, for $a = 1, 2$, and $H_1 \equiv H_2$, then $k_1 = k_2$ and $\{(\alpha_{1i}, \mathcal{G}_{1i})\}_{i=1}^{k_1}$ is a permutation of $\{(\alpha_{2i}, \mathcal{G}_{2i})\}_{i=1}^{k_1}$.

The outline of the corresponding proof is given in Appendix A.

The problem of finite mixture estimation is usually too complicated to find a solution directly by a maximum likelihood (ML) approach. Thus, a family of iterative Expectation-Maximization (EM) methods is usually used to numerically find the local maximum of the likelihood function [9]. There exists a considerable variety of EM-method modifications each suitable for some specific problem settings. In our case the optimal approach is presented by a Stochastic EM (SEM) approach [9], which enables to avoid an unfeasible (due to very complex shape of likelihood function [5]) GFD pdf maximization and to improve the exploratory properties of EM, which is very critical in case of inaccurate (random) initialization. For the SEM algorithm the complete data is represented by a set $\{(r_i, s_i), i = 1, \dots, N\}$, where r_i are the observations (SAR amplitudes) and s_i - the missing labels: given a mixture with K components, a label $s_i \in \{\sigma_1, \sigma_2, \dots, \sigma_K\}$ assigns the i -th observation to one of the K components.

We embed the Method of Log-Cumulants (MoLC) [10] method for parameter estimation in the M-step of SEM scheme. This is done instead of the ML-estimate which come at too high computational price for GFD [5]. MoLC represents a parametric pdf estimation technique suitable for distributions defined on $[0, +\infty)$, and it has been widely applied in the context of SAR-specific parametric pdf families due to its analytical adequacy for the multiplicative models, like “signal-speckle” system for SAR [2, 3, 7, 10]. MoLC adopts the Mellin transform [10] by analogy to the Laplace transform in the moment generating function. Given a non-negative random variable u , one defines the “second-kind characteristic function” ϕ_u of u as the Mellin transform [10] \mathcal{M} of the pdf of u , i.e.:

$$\phi_u(s) = \mathcal{M}(p_u)(s) = \int_0^{+\infty} p_u(u) u^{s-1} du, \quad s \in \mathbb{C}.$$

The derivatives $\kappa_\nu = [\ln \phi_u]^{(\nu)}(1)$ are the ν th order log-cumulants, where (ν) stands for the ν th derivative, $\nu \in \mathbb{N}$. In case of the transform convergence, the following MoLC equations take place [10]:

$$\kappa_1 = E\{\ln u\}, \quad \kappa_j = E\{(\ln u - \kappa_1)^j\}, \quad j = 2, 3$$

First, we find sample estimates for these log-cumulants [11]:

$$\begin{aligned} \tilde{\kappa}_1 &\approx k_1 = \frac{1}{N} \sum_{i=1}^N \ln r_i, & \tilde{\kappa}_2 &\approx k_2 = \frac{1}{N-1} \sum_{i=1}^N [\ln r_i - k_1]^2, \\ \tilde{\kappa}_3 &\approx k_3 = \frac{N}{(N-1)(N-2)} \sum_{i=1}^N [\ln r_i - k_1]^3. \end{aligned}$$

Then, we analytically express κ_j as functions of unknown parameters and replacing then κ_j by their sample estimates k_j we finally get the system of MoLC-equations. In case of GFD it writes [7]:

$$k_1 = \log \sigma + \frac{\Psi(0, \kappa)}{\nu}, \quad (3)$$

$$k_j = \frac{\Psi(j-1, \kappa)}{\nu^j}, \quad j = 2, 3, \dots, \quad (4)$$

where $\Psi(n, x)$, $n \in \{0\} \cup \mathbb{N}$, denotes the polygamma function [8].

The following theorem explores the asymptotical properties of the MoLC estimates for GFD.

Theorem 2. The sequence of MoLC estimates $\{(\hat{\nu}_n, \hat{\kappa}_n, \hat{\sigma}_n)\}_{n=1}^\infty$ calculated for GFD from (3)-(4) and observations $\{r_i\}_{i=1}^n$ is consistent, i.e. it converges in probability to the true parameter values $(\nu^*, \kappa^*, \sigma^*)$ as $n \rightarrow \infty$.

The outline of the corresponding proof is given in Appendix B.

As demonstrated in Appendix B the system (3)-(4) is invertible, yet its analytical solution is unfeasible. We propose to use an approximate solution of this system that is very simple computationally and it is given by:

$$\begin{cases} \hat{\kappa} = \left[\left(\left[\frac{\alpha}{2} \right]^2 + \left[\frac{\beta}{3} \right]^3 \right)^{1/2} - \frac{\alpha}{2} \right]^{1/3} + \\ \quad + \left[- \left(\left[\frac{\alpha}{2} \right]^2 + \left[\frac{\beta}{3} \right]^3 \right)^{1/2} - \frac{\alpha}{2} \right]^{1/3} - \frac{c_2}{3c_3}, \\ \hat{\nu} = \left[\frac{\Psi(1, \hat{\kappa})}{k_2} \right]^{1/2} \\ \hat{\sigma} = \exp \left[k_1 - \frac{\Psi(0, \hat{\kappa})}{\hat{\nu}} \right] \end{cases}$$

with

$$c_0 = k_3^2 - 8k_2^3, \quad c_1 = 6k_3^2 - 16k_2^3, \quad c_2 = 12k_3^2 - 8k_2^3, \quad c_3 = 8k_3^2, \quad (5)$$

$$\alpha = \frac{27c_3^2 c_0 - 9c_3 c_2 c_1 + 2c_2^3}{27c_3^3}, \quad \beta = \frac{c_1}{c_3} - \frac{1}{3} \left(\frac{c_2}{c_3} \right)^2. \quad (6)$$

The derivation of these solutions is presented in Appendix C.

Thus, the t th iteration of histogram-based SEM for GFD mixture estimation goes as follows:

• **E-step:** compute, for each greylevel z and i -th component, the posterior probability estimates corresponding to the current pdf estimates, i.e. $z = 0, \dots, Z-1$:

$$\tau_i^t(z) = \frac{\alpha_i^t \mathcal{G}_i^t(z)}{\sum_{j=1}^{K_t} \alpha_j^t \mathcal{G}_j^t(z)}, \quad i = 1, \dots, K_t,$$

where $\mathcal{G}_i^t(\cdot)$ is the σ_i -conditional pdf estimate on the t th step.

• **S-step:** sample a label for each greylevel z according to the current estimated posterior probability distribution $\{\tau_i^t(z) : i = 1, \dots, K_t\}, z = 0, \dots, Z-1$.

• **M-step:** $\forall i$, compute the sample estimates of the log-cumulants (k_{1i}, k_{2i}, k_{3i}) and mixture proportion α_i . Find then the parameter estimates $(\alpha_i^{t+1}, \nu_i^{t+1}, \kappa_i^{t+1}, \sigma_i^{t+1})$ as above.

• **K-step:** $\forall i$: if $\alpha_i^{t+1} < \gamma$, eliminate the i -th component, update K_{t+1} . The choice of threshold γ is not critical, e.g. $\gamma = 0.005$.

3. SUPERVISED SAR CLASSIFICATION WITH GFD MIXTURES

The problem of classification consists in attributing to each pixel of the considered image \mathcal{I} a label assigning it to one of the M thematic classes. Working in the mainframe of the supervised classification we consider some training pixels to be available for each of the M classes. First, we perform the learning of the statistical properties of each class by estimating a GFD mixture pdf on its training pixels, thus getting pdfs $p_i(r)$, $i = 1, \dots, M$. It is possible then to get the first classification by picking for each pixel the label with the maximal value p_i , i.e. the ML-classification, however this gives very noisy results. This is especially critical for SAR images, damaged by speckle [1]. Thus, some regularization has to be performed. Here, we suggest the use of a Markov random field (MRF) model which takes into account the local context information [12]. To implement the MRF-energy minimization we run the Modified Metropolis Dynamics (MMD) algorithm, which is a compromise

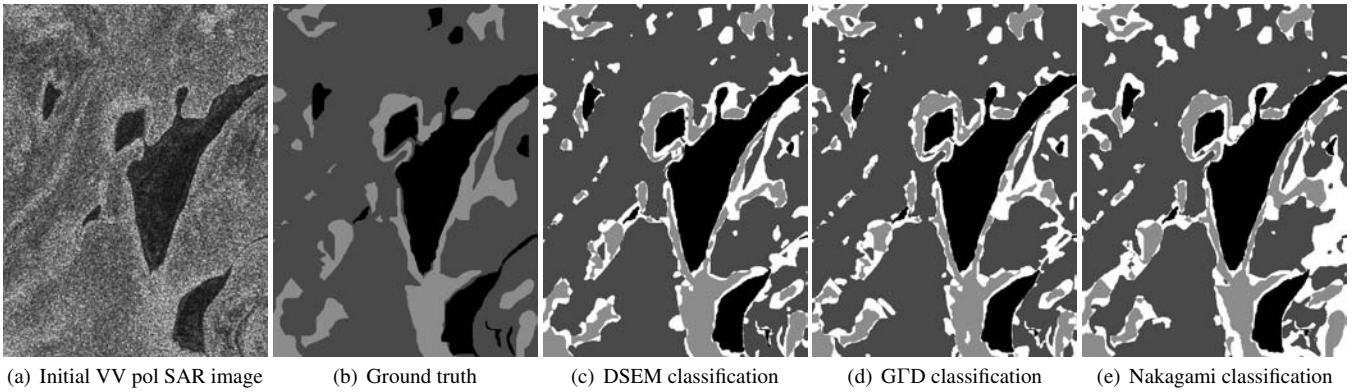


Figure 1: (a) Initial image, 800x1000 pixels, (b) Ground truth (black - water, dark grey - wet soil, light grey - dry soil), (c) DSEM, (d) GFD mixture and (e) Nakagami mixture classifications maps (black - water, dark grey - wet soil, light grey - dry soil, white - misclassification).

Table 1: Automatically estimated numbers of mixture components K^* and classification accuracies obtained by considered finite mixture methods: class accuracies and overall accuracies.

Mixture method	Water	Wet soil	Dry soil	Overall			
GFD	88.55%	2	90.31%	3	68.46%	3	85.95%
DSEM [2]	88.67%	2	88.86%	2	72.51%	3	86.19%
Nakagami [3]	90.24%	2	84.97%	2	64.51%	3	81.38%

solution between Iterative Conditional Modes and Simulated Annealing, and, as such, is computationally feasible and provides reasonable results in real classification problems [13]. The employed here MRF and MMD algorithms and their settings are not novel and we refer the reader to an earlier work for more details [14].

4. EXPERIMENTAL RESULTS

The algorithm settings are the following. We initialize $K_0 = 5$ as an overestimate for each class, so that SEM finds K^* by annihilating the redundant components. The number of iterations for SEM is set to $T = 300$. We run the MRF coefficient estimation on the ML-classification and the MMD settings are as in [14].

The experiments were performed on a VV polarization, 6.5 m ground resolution, 2.66-look TerraSAR-X (©Infoterra GmbH, 2008) image acquired over Sanchagang, China. The application was to epidemiological monitoring and the thematic classes were: water, wet soil and dry soil. Figure 1 presents the classification results and comparisons with other mixture-based SAR statistics models, i.e. DSEM model [2] and Nakagami-gamma mixture model based on SEM (as described above), analogical to [3] for amplitudes. Table 1 reports K^* estimates and the obtained accuracies. We observe that DSEM slightly outperforms our model, however this was to be expected as DSEM is based on a mixture of several pdf families, of which GFD is part. The Nakagami-gamma fit is worse due to a lower flexibility of the parametric pdf model, and we remind that Nakagami (and gamma) pdfs are special cases of GFD corresponding to $\nu = 2$ ($\nu = 1$) in (1). From the performance point of view, the estimates (averaged over $M = 3$ classes) were obtained in: $T_{\text{DSEM}} = 93s$, $T_{\text{GFD}} = 22s$, $T_{\text{Nakagami}} = 18s$ on an Intel Core 2 Duo 1.83GHz, 1Gb RAM, WinXP system. This result confirms our GFD mixture model's flexibility and fast performance.

5. CONCLUSIONS

In this paper we have introduced a novel GFD mixture estimation approach and demonstrated its efficiency in the application to high

resolution SAR image classification. We have proved the identifiability of this type of finite mixtures and demonstrated the consistency of the exploited parameter estimation procedure. In order to improve the performance of the GFD mixture estimation we suggested the use of an approximative solution of the equations involved, thus, avoiding time-consuming iterative processes. The experimental comparison demonstrated a high potential of this model in the problem of high resolution SAR statistical modeling: it performs almost as well as the most general dictionary-based stochastic expectation maximization (DSEM) problem, and as fast as a very simple Nakagami-gamma mixture-based one. We want to point out that the use of GFD mixtures and the developed estimation procedure can be interesting to other applications, such as, e.g., speech signal processing, health management and economical applications.

APPENDIX A. OUTLINE OF THEOREM 1 PROOF

To prove the identifiability of finite mixtures of GFDs we show that the necessary identifiability conditions are fulfilled in the form proposed by Atienza et al. [15]. In order to do so we demonstrate a linear transform $M_G(t) : \mathcal{G}(x) \rightarrow \phi_G$, and a point t_0 from the support of ϕ_G that enable us to find the total parametric-space ordering of pdfs $\mathcal{G}(t|\nu, \kappa, \sigma)$ that fulfills the following condition:

$$F_1 \prec F_2 \Leftrightarrow \lim_{t \rightarrow t_0} \frac{\phi_{F_2}(t)}{\phi_{F_1}(t)} = 0, \quad (7)$$

where F_1, F_2 are two GFDs.

We define the transform M as follows:

$$M : \phi_G(t) = \int_0^\infty t^x \mathcal{G}(x) dx = \sigma^t \frac{\Gamma(\kappa + t/\nu)}{\Gamma(\kappa)}$$

and $t_0 = \infty$. We then show that when $t \rightarrow t_0$ we have

$$\frac{\phi_{F_2}(t)}{\phi_{F_1}(t)} \sim C \exp\left\{\left[\frac{1}{\nu_2} - \frac{1}{\nu_1}\right] t \log t + [\kappa_2 - \kappa_1] \log t + \left[(\log \sigma_2 - \log \sigma_1) - \left(\frac{1}{\nu_2} - \frac{1}{\nu_1}\right) - \left(\frac{\log \sigma_2}{\nu_2} - \frac{\log \sigma_1}{\nu_1}\right)\right] t\right\}$$

with a constant C , and prove the following ordering satisfies (7):

$$F_1 \prec F_2 \Leftrightarrow \begin{cases} [\nu_2 > \nu_1] \\ [\nu_2 = \nu_1, \sigma_2 < \sigma_1] \\ [\nu_2 = \nu_1, \sigma_2 = \sigma_1, \kappa_2 < \kappa_1]. \end{cases}$$

Thus, finite mixtures of GFDs are identifiable.

APPENDIX B. OUTLINE OF THEOREM 2 PROOF

To prove the consistency of the sequence of MoLC estimates $\{\hat{\xi}_n\}_{n=1}^\infty = \{(\hat{\nu}_n, \hat{\kappa}_n, \hat{\sigma}_n)\}_{n=1}^\infty$, i.e. that it converges in probability to the true parameter values $(\nu^*, \kappa^*, \sigma^*)$, we first demonstrate the invertibility of $\Theta : (0, +\infty)^3 \rightarrow \mathbb{R}^3$ and the continuity of mapping $\hat{\xi}_n = \Theta^{-1}(\hat{k}_n)$, where $\hat{k}_n = (\hat{k}_{1n}, \hat{k}_{2n}, \hat{k}_{3n})$ is the sequence of sample log-cumulants [11]. According to (3), (4),

$$\Theta(\xi) = \left(\log \sigma + \frac{\Psi(0, \kappa)}{\nu}, \frac{\Psi(1, \kappa)}{\nu^2}, \frac{\Psi(2, \kappa)}{\nu^3} \right).$$

This is proved by referring to the implicit function theorem and properties of polygamma functions [8].

Then we take advantage of the fact that the expressions for variances of the sample log-cumulants are well known [11] and show that for any value of $\varepsilon > 0$, if $\|\hat{k}_n - k^*\| < \delta(\varepsilon)$, and the central moments $\mathbb{E}[(\log X)^j]$, $X \sim \mathcal{G}$, till order $j = 6$ exist and are finite, then we get

$$\mathbb{P}\{|\hat{\nu}_n - \nu^*| < \varepsilon, |\hat{\kappa}_n - \kappa^*| < \varepsilon, |\hat{\sigma}_n - \sigma^*| < \varepsilon\} \geq 1 - \frac{\sqrt{L(n)}}{\delta(\varepsilon)},$$

where $L(n) = O(n^{-1})$. The derivation of explicit expressions for the necessary log-moments of GFD demonstrates them to be finite whatever the values of parameters. Thus, we get the required

$$\lim_{n \rightarrow \infty} \mathbb{P}\{|\hat{\nu}_n - \nu^*| < \varepsilon, |\hat{\kappa}_n - \kappa^*| < \varepsilon, |\hat{\sigma}_n - \sigma^*| < \varepsilon\} = 1.$$

APPENDIX C. DERIVATION OF THE CLOSED-FORM ESTIMATOR FOR GFD PARAMETERS

In order to find a numerical solution of the system (3)-(4) and render the solution computationally as fast as possible we suggest to use an asymptotic decomposition of the polygamma functions [8]:

$$\begin{aligned} \Psi(1, x) &= x^{-1} + 0.5x^{-2} + \bar{o}(x^{-2}), \\ \Psi(2, x) &= -x^{-2} - x^{-3} + \bar{o}(x^{-3}), \end{aligned}$$

with $x \in \mathbb{R}$, $x \rightarrow \infty$.

Isolating ν in (4) for $j = 2, 3$ we search for the solutions of the algebraic equation $c_3 \kappa^3 + c_2 \kappa^2 + c_1 \kappa + c_0 = 0$, with c_i as in (5). By substituting $y = \kappa + c_2/(3c_3)$ we get to the reduced equation $y^3 + \beta y + \alpha = 0$ with α and β as in (6). Now we apply the Cardano's formula and pick the positive solution. Thus, we get the formula for $\hat{\kappa}$, and the solutions for $\hat{\nu}$, $\hat{\sigma}$ follow from (3)-(4).

We point out that to have a required real value solution $\hat{\kappa}$ using the involved decomposition and Cardano's formula we need the weak following condition fulfilled: $3k_2^3 \leq 8k_3^3$. Dealing with real SAR imagery we never stumbled upon a case when this condition fails.

6. ACKNOWLEDGMENT

The work of the first author was partially supported by the federal target program "Scientific and scientific-pedagogical personnel of innovative Russia in 2009-2013" and by EPI ARIANA. The authors would like to acknowledge the helpful advice of Dr. Gabriele Moser and Prof. Sebastiano B. Serpico, Univ. of Genoa, Italy, in the preparation of this work. The employed in experiments TerraSAR-X image of Sanchagang was taken from <http://www.infoterra.de/>.

7. REFERENCES

[1] C. Oliver and S. Quegan, *Understanding Synthetic Aperture Radar Images*, Artech House, Norwood, 1998.

- [2] V. Krylov, G. Moser, S. B. Serpico, and J. Zerubia, "Enhanced dictionary-based SAR amplitude distribution estimation and its validation with very high-resolution data," *IEEE Geosci. Rem. Sens. Lett.*, 2010, (in press). DOI:10.1109/LGRS.2010.2053517.
- [3] J.-M. Nicolas and F. Tupin, "Gamma mixture modeled with "second kind statistics": application to SAR image processing," in *Proceedings of IGARSS*, Toronto, Canada, 2002, pp. 2489–2491.
- [4] E. W. Stacy, "A generalization of the gamma distribution," *Ann. Math. Statist.*, vol. 33, pp. 1187–1192, 1962.
- [5] K. Song, "Globally convergent algorithms for estimating generalized gamma distributions in fast signal and image processing," *IEEE Trans. on Image Process.*, vol. 17, no. 8, pp. 1233–1250, 2008.
- [6] C. Kleiber and S. Kotz, *Statistical Size Distributions in Economics and Actuarial Sciences*, Wiley-Interscience, New York, 2003.
- [7] H.-C. Li, W. Hong, and Y.-R. Wu, "Generalized Gamma distribution with MoLC estimation for statistical modeling of SAR images," in *Proceedings of APSAR*, Huangshan, China, 2007, pp. 525–528.
- [8] M. Abramowitz and I. Stegun, Eds., *Handbook of Mathematical Functions*, Dover, New York, 1964.
- [9] G. Celeux, D. Chauveau, and J. Diebolt, "Some stochastic versions of the EM algorithm," *J. Statist. Comp. Sim.*, vol. 55, 1996.
- [10] C. Tison, J.-M. Nicolas, F. Tupin, and H. Maitre, "A new statistical model for Markovian classification of urban areas in high-resolution SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 10, pp. 2046–2057, 2004.
- [11] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, New York, 3rd edition, 1991.
- [12] Z. Kato, J. Zerubia, and M. Berthod, "Unsupervised parallel image classification using Markovian models," *Pattern Recognition*, vol. 32, no. 4, pp. 591–604, 1999.
- [13] Z. Kato, J. Zerubia, and M. Berthod, "Satellite image classification using a modified Metropolis dynamics," in *Proceedings of ICASSP*, San Francisco, USA, 1992, pp. 573–576.
- [14] G. Moser, V. Krylov, S. B. Serpico, and J. Zerubia, "High resolution SAR-image classification by Markov random fields and finite mixtures," in *Proceedings of SPIE*, San Jose, USA, 2010, pp. 753308–01–753308–12.
- [15] N. Atienza, J. Garcia-Heras, and J. M. Munoz-Pichardo, "A new condition for identifiability of finite mixtures," *Metrika*, vol. 63, pp. 215–221, 2006.

About the authors

Vladimir A. Krylov is a PhD student at the Dept. of Mathematical Statistics, Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University.
Email: vkrylov@cs.msu.ru

Josiane Zerubia is a senior researcher (Directeur de Recherche), head of the ARIANA research team, INRIA/CNRS/UNSA, INRIA Sophia Antipolis, France.
Email: Josiane.Zerubia@sophia.inria.fr

Efficient Super-Resolution Up-Conversion Algorithm for HDTV

Vadim Vashkelis, Natalia Trukhina, Ivan Chirkov
 {vashkelis, ntrukhina, chirkov.ivan}@gmail.com

Abstract

Video up-conversion takes significant place in various application areas. One of important application areas is standard-definition television (SDTV) video processing to get high-definition television content (HDTV) for broadcast. However, high-quality up-conversion is a challenging task. Most practical implementations use spatial domain processing such as video frame interpolation for video up-scale. Meanwhile, due to sampling limitation the high-frequency component of output HD video cannot be efficiently reconstructed by applying only the spatial domain processing and high-quality up-conversion usually requires temporal domain processing as well. The authors propose practical implementation of such up-conversion technique providing significantly better visual results in comparison to traditional methods of SDTV to HDTV up-conversion.

Keywords: *Super-Resolution, video up-conversion, HDTV, video enhancement.*

1. INTRODUCTION

There are many up-conversion algorithms widely used. Most of them use spatial signal processing to construct new data points within a set of existing pixels. These methods are based on general interpolation approach, namely on construction of new data points within a set of existing data points with fixed sampling rate. As an example of such interpolation algorithms the nearest neighbour, bilinear, bicubic, spline, sinc, lanczos and some others can be mentioned. These methods use various mathematical interpretation of the spatial signal to construct necessary points.

Another widely used set of methods is frequency domain processing, usually fast Fourier transform or wavelet analysis based algorithms. These methods are based on the shifting property of the Fourier transform, the aliasing relationship between the continuous Fourier transform and the discrete Fourier transform. Accurate processing of the Fourier transform results can give us the frequency domain coefficients of the original scene, which may then be recovered by inverse Fourier transform. [1] However, frequency domain processing has several important disadvantages. These methods require the existence of a transformation which is the Fourier domain equivalent of the spatial domain motion model what is not always feasible. Also, it is difficult to include spatially varying degradation models in the frequency domain reconstruction formulation. [2]

For the last years the variety of works were dedicated to relatively novice approach to video up-conversion addressing to resolve the bandwidth limitation of other methods. This set of algorithms is called super-resolution (SR). This work represents the computational efficient high-quality implementation of super-resolution technology for video up-conversion aimed on television and broadcast applications.

2. TRADITIONAL APPROACH

SR is technique that enhances the resolution of an imaging system. This technique uses additional image information for high resolution. It may be information from single or multiple input images. Single image SR method extracts high-resolution image de-

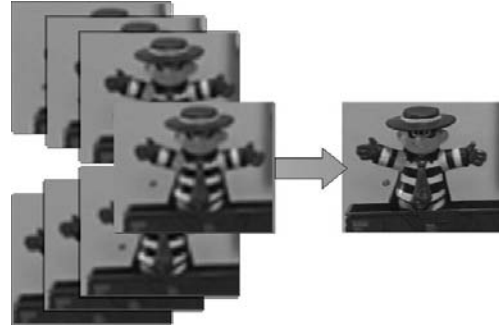


Figure 1: Reconstruction of single super-resolution image based on analysis of several images.

tails from a single low-resolution image, which cannot be achieved by simple sharpening [3] and traditional interpolation. It uses other parts of the low resolution images to guess how the high resolution image should look like.

Multiple-frame SR is a method based on idea of using information from several images to create one up-scaled image. In particular, the source video sequence contains similar, but not identical information. The additional information available in these frames makes possible the reconstruction of visually superior frames at higher resolution than that of the original data. This method tries to extract details from one frames to reconstruct other frames. The SR algorithms are possible only if aliases exist, and the images have sub-pixel shifts. [3] This approach differs a lot from some sophisticated image up-scaled methods which try to synthesize artificial details.

Generally, there are three critical factors affecting super-resolution restoration. Firstly, reliable sub-pixel motion information is essential. Poor motion estimates are more detrimental to restoration than a lack of motion information. Secondly, observation models must accurately describe the imaging system and its degradations. Thirdly, restoration methods must provide the maximum potential for inclusion of a priori information. [4]

3. GENERAL SUPER-RESOLUTION APPROACH

Multiple-frame SR for video sequences uses information from the sub-pixel shifts between several frames of the same scene within a video. This pixel shift is caused by a relative motion between the scene and camera. The video with improved resolution can be created by merging the data from a set of low-resolution frames taking the relative pixel shifts into the account. SR works when several low resolution images $LR(x, y)$ contain slightly different views of the same objects. In this case total information about the object is much higher than information in one frame. Using existing information from current frame $LR_i(x, y)$ and getting additional sub-information from several previous $.., LR_{i-n}, .., LR_{i-2}, LR_{i-1}$ and several next $LR_{i+1}, LR_{i+2}, .., LR_{i+n}, ..$ frames we can reconstruct high resolution image $HR_i(x, y)$. Simplified method of reconstruction may be defined as function F from several frames(1).

$$\begin{aligned}
HR_i(x, y) = F(& LR_{i-n}(x_{i-n}, y_{i-n}), \\
& \dots, \\
& LR_{i+n}(x_{i+n}, y_{i+n}))
\end{aligned} \quad (1)$$

The methods discussed in other papers usually describe the reconstruction of HR in ideal conditions. Under these conditions all compensated objects precisely assist in concerned frames LR_j and every sub-pixel shifts are found and position in other frames (x_j, y_j) are exactly known. Actually it is important to know if we afford to use sub-pixels from neighbour frames or we have to amount current low-resolution frame information only. It involves irregular structure of reconstruction on non-uniformly spaced sampling grid and smart image recognition. The registration of low-resolution image sequence results in a composite image of samples on a non-uniformly spaced sampling grid. These sample points are interpolated and positioned over the high-resolution sampling grid. However, despite the simplicity of such model it does not take into consideration the fact that samples of the low resolution images cannot be results of ideal sampling and relative pixel shifts cannot be known a priori. This results in the fact that the reconstructed image does not contain the full range of frequency content that ideally could be reconstructed. Practical implementation of SR shows that high resolution and quality are unachievable without strong model of data points recognition and detection. Also high quality interpolation base is necessary. The expanded formalization of SR implementation can be described as

$$\begin{aligned}
HR_i(x_{hr}, y_{hr}) = F(& LR_{i-n}(x_{lr_{i-n}}, y_{lr_{i-n}}), \\
& \dots, \\
& LR_{i+n}(x_{lr_{i+n}}, y_{lr_{i+n}}), \\
& INT(LR_i, x_{hr}, y_{hr}))
\end{aligned} \quad (2)$$

Due to all these limitations the most important key factors for efficient super-resolution processing are quality of generalized up-scaling, accurate motion estimation and robustness of the super-resolution construction procedure that creates uniform set of data points from non-uniform points mesh. Below, in this work, we will consider all of these three factors separately.

3.1 Non-uniformly sampled grid interpolation

The first step of SR is high quality up-scaling. It is fundamental point for both the motion estimation and super-resolution construction procedure. The best decision is to use multivariate harmonic interpolation with non-uniform mesh nodes. In signal processing, a sinc filter is an idealized filter that removes all frequency components above a given bandwidth, leaves the low frequencies alone, and has linear phase. The filter's impulse response is a sinc function in the time domain, and its frequency response is a rectangular function. [3] In digital signal processing and information theory, the normalized sinc function is commonly defined by

$$\text{sinc}(x) = \frac{\sin \pi x}{\pi x} \quad (3)$$

The Lanczos filter is a windowed form of the sinc filter. Its impulse response is the normalized sinc function $\text{sinc}(x)$ windowed by the Lanczos window. The Lanczos window is itself the central lobe of a scaled sinc , namely $\text{sinc}(x/a)$ for a from $-a$ to a (the central lobe scaled to run from $-a$ to a). The resulting function is then used as a convolution kernel to resample the input field. [5] Its formula is given by:

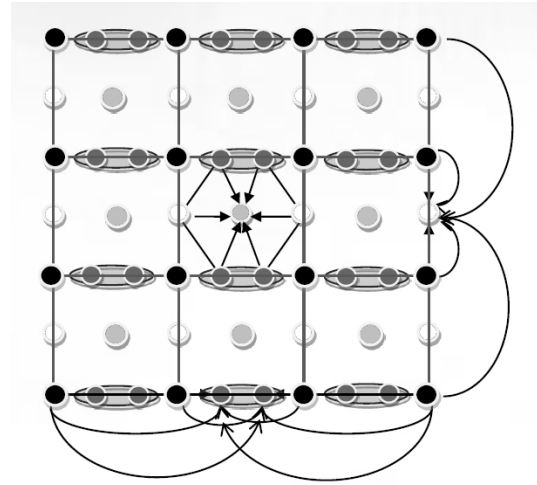


Figure 2: Smart Lanczos interpolation with non-uniform mesh nodes.

$$L(x) = \begin{cases} \text{sinc}(x) \cdot \text{sinc}(\frac{x}{a}), & -a < x < a, x \neq 0 \\ 1, & x = 0 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

The Lanczos filter has been compared with other filters, particularly other windowing of the sinc filter. Lanczos is the best compromise in terms of reduction of aliasing, sharpness, and minimal ringing. Nevertheless, the regular structure and linear nodes do not give the best results.

Proposed approach uses different weight coefficients for different mesh nodes to improve visual quality of interpolated images. Analysis of the morphological structure of image and individual choice of weight coefficients allow to select correct nodes for high quality interpolation and make precise motion estimation. One way of possible adapted smart Lanczos interpolation with non-uniform mesh nodes is described in Figure 2. It is an example of 4x up-scaling where:

- black nodes are pixels of based input image,
- white nodes are pixels of regular vertical Lanczos interpolation,
- dark gray nodes are pixels of possible horizontal interpolation,
- light gray nodes are results of total non-uniform interpolation.

Schema of non-uniform mesh nodes can be different and adapts for input image structure and combines different Lanczos windows for different types of images.

3.2 Heterogeneous motion estimation

The large regions overlap that usually exists between successive frames of the same sequence and the multiple sampling of this regions in several frames, yield the conclusion that it is possible to combine this information to achieve higher spatial resolution images. Motion estimation techniques are used to find this overlapping areas from frame to frame. [6] The resulting motion vectors

must be at least sub-pixel precision to provide useful information for SR. For the best quality results quarter-pixel precision is used.

Most papers dedicated to super-resolution construction claim that image registration is known a priori. Meanwhile, image registration or in other words accurate inter frame motion estimation is a crucial component of super-resolution processing. Insufficient accuracy of image registration inevitably leads to significant quality degradation and makes super-resolution approach nearly useless.

Most popular yet powerful enough practical motion estimation approach utilizes the sum of absolute differences (SAD) as a criterion for image templates matching (5).

$$SAD = \sum_{j=0}^{j=n-1} \sum_{i=0}^{i=m-1} |I_1(i, j) - I_2(i, j)| \quad (5)$$

This technique is used in many video coding applications [7] and characterized by high computational simplicity. However, there are number of know lacks of SAD approach what makes it less applicable for super-resolution image registration then for video compression [8]. A Most noticeable problem of SAD-based matching is inconsistency in the case of sufficient noise additions and average brightness (DC component) change.

Many papers dedicated to the problems of image registration and template matching point on morphological hit-or-miss criteria for image matching [9, 10, 11, 12]. The proposed approach combines the power of both methods for creating computational efficient and effective image registration approach. The proposed method of motion estimation combines computational simplicity of SAD based methods and efficiency of morphological analysis MSC (Morphological search criteria)(6) It as defined as morphological SAD - $MSCSAD$ (7) where $kSAD$ and $kMSC$ are weighting factors.

$$MSC = \sum_j (MAX_i |I_1(i, j) - I_2(i, j)| - MIN_i |I_1(i, j) - I_2(i, j)|) \quad (6)$$

$$MSCSAD = (SAD \quad MSC) \times \begin{pmatrix} kSAD \\ kMSC \end{pmatrix} \quad (7)$$

Position that turns out to be the most similar to the current image pixel in the reference frame, is given by the candidate with the minimum $MSCSAD$ value:

$$MSCSAD(x_{lr}, y_{lr}) = MIN_{x,y} (MSCSAD(x, y)) \quad (8)$$

For precise search and future accurate reconstruction it is important to select the best candidate from operating positions and to understand if this candidate affords to give additional resolution indeed or such position does not exist at all. It is proposed to use complicated pyramid structure of motion estimation with several steps for parcelling out input images and separating background and foreground with objects combining.

As a result of motion estimation and object detection we have half/quarter pixel motion vectors, values of $MSCSAD$ and map of objects and theirs motion. This parameters give us information for SR reconstruction and possibility to construct strong criteria for its employment.

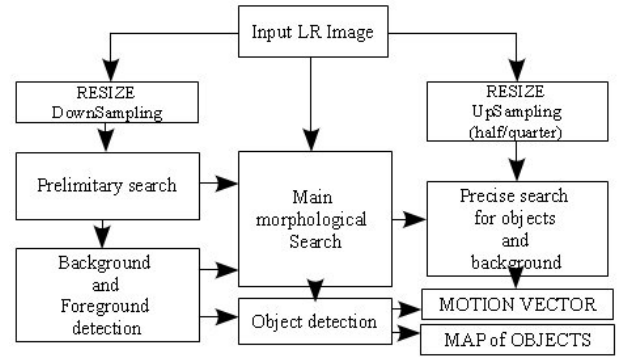


Figure 3: Precise motion estimation.

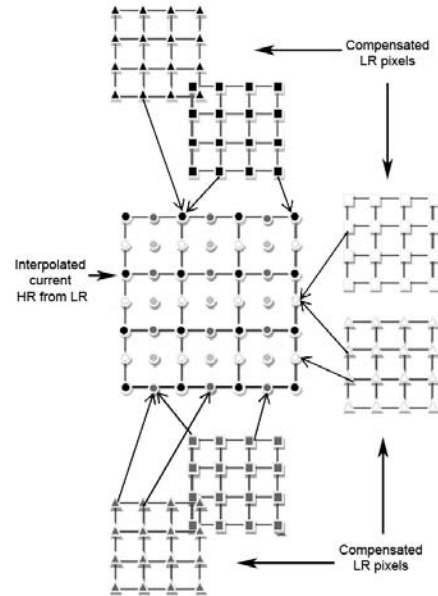


Figure 4: Precise motion estimation.

3.3 Super-Resolution frame construction

Registering a set of low-resolution images using motion compensation results in a single, dense composite image of non-uniformly spaced samples. The super-resolution image can be constructed from this composite using techniques for reconstruction from non-uniformly spaced samples. Restoration techniques are sometimes applied to compensate for degradations [11]. Description of iterative reconstruction techniques can also be noticed [13]. Such interpolation methods are unfortunately overly simplistic. Since the observed data result from severely down-sampled, spatially averaged areas, the reconstruction step (which typically assumes impulse sampling) is incapable of reconstructing significantly more frequency content that is present in a single LR frame. Degradation models are limited, and no a priori constraints are used. There is also question of the optimality of separate merging and restoration steps.

Hence, the construction of super-resolution frame can be formulated as an approximation of non-uniform mesh by the uniformly positioned set of samples. Proposed approach uses different weight coefficients for different mesh nodes to improve visual quality of

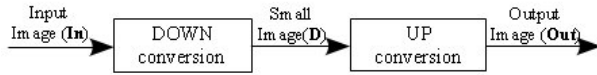


Figure 5: Schema of down-up-conversion for objective quality measurement.

interpolated images. Analysis of morphological structure of image and individual choice of weight coefficients allows to select correct nodes for high quality SR reconstruction.

4. QUALITY MEASUREMENTS

We will rely on two quality measurement criteria subjective and objective. As for objective quality metric a peak signal to noise ratio (*PSNR*) is usually used in practice. It is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. This criterion is usually defined via mean squared error (*MSE*)

$$MSE = \frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} [I_1(i, j) - I_2(i, j)]^2 \quad (9)$$

The PSNR can be defined as

$$PSNR = 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \quad (10)$$

Meanwhile, in case of up-conversion we do not have the reference video with the appropriate frame size because such video can be only obtained by some other up-conversion method from the input and any alternative up-conversion approach will add its own conversion error. To avoid this problem we can use the results of some typical down-conversion as input data (see Figure 5).

In this case the output results of up-conversion routine can be objectively measured against input signal such as $PSNR(In, Out)$.

Subjective evaluation is another important measurement approach because objective metrics such as *PSNR* cannot fully substitute manual visual perception. During the subjective testing such visual characteristics as video stability, aliasing effect and overall impression were manually evaluated.

For quality testing, we use in this paper two typical image sequences. The first one, *Shields*, is a sequence with moving background containing many small details and texts and local motion on foreground. The second one, *Mobcal*, is a sequence containing global motion on background and fast motion on foreground. Both *HR* test image sequences are first down-scaled to *LR* by a factor of 2 in both vertical and horizontal directions. After *LR* sequence is reconstructed by different interpolation methods and concerned SR algorithm and compared with subjective and objective quality measurement criteria (see Figure 6).

Figures 6 and 7 show input *LR* frame and results of its bilinear, bicubic and Lanczos (window size a is 3) interpolations and SR transformation. Figure 7 also contains diagram of peak values of *PSNR* in comparison.

Quantitative *PSNR* comparisons between reconstructions for whole test sequences and distinctive selected regions in the images are in Table 1. According to quality measurements and visual comparison SR allows efficiently reconstruct low-resolution video to high quality and resolution video. As compared with different inter-

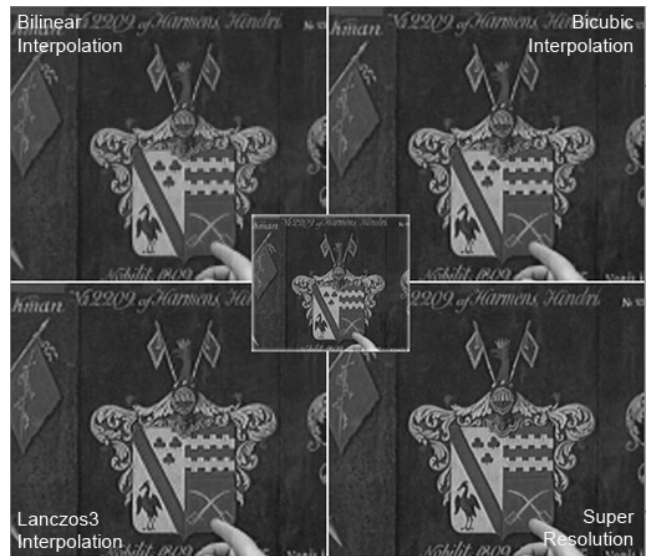


Figure 6: Visual quality comparison.

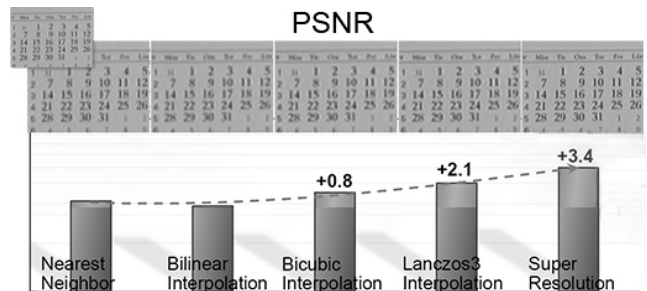


Figure 7: Objective PSNR measurement.

polation methods SR significantly increases resolution within small detailed and slow motion videos. Fast moving videos are reconstructed with the same quality and objective quality measurements as Lanczos interpolation.

5. CONCLUSION

The work demonstrates one possible approach for efficient implementation of high-quality up-conversion solution. The authors demonstrate how to resolve the problem of high computational complexity of every super-resolution solution without degradation of output visual quality. Proposed algorithms allow to build computational efficient solutions on various DSP or GPU platforms. At the same time the efficiency of computation does not affect visual quality of the proposed solution.

The results of objective and subjective quality comparison against well-known spatial domain based alternatives displays that method given exceeds the results of traditional algorithms in both subjective and objective fields.

A reasonable trade-off between quality of up-conversion and relatively low computational complexity of proposed method allows to design the real-time high-quality video up-conversion devices on various DSP or GPU platforms that will address the problem of efficient SD-to-HD video conversion.

	Bicubic Interpolation		Lanczos3 Interpolation		Super Resolution	
	Average PSNR	Max PSNR	Average PSNR	Max PSNR	Average PSNR	Max PSNR
Shields (moving background with small details)	30,69	31,03	31,22	31,97	32,13	32,94
Shields (moving background without small details)	33,48	33,71	33,99	34,22	34,24	34,57
Shields (local motion on foreground)	31,82	32,25	32,98	33,26	33,21	33,78
Shields	31,89	33,12	32,14	33,69	33,12	35,00
Mobcal (global motion on background)	32,24	32,96	32,68	33,09	33,56	33,97
Mobcal (moved text on foreground)	29,22	29,87	30,12	30,65	31,31	31,86
Mobcal (fast motion on foreground)	31,21	31,94	31,58	32,15	31,64	32,12
Mobcal	30,52	31,64	31,19	32,78	32,01	33,14

Table 1: PSNR comparison on different parts of images.

	Reconstruction of details with slow or without motion	Reconstruction of details with fast motion	Anti-aliasing effect	Clearness
<i>Shields</i>	Better	Same	Better	Better
<i>Mobcal</i>	Better	Same	Better	Better

Table 2: Subjective visual quality SR in comparison with Lanczos3 interpolation.

6. REFERENCES

- [1] S. Borman and R. Stevenson, "Super-resolution from image sequences - a review," *Circuits and Systems, 1998. Proceedings. 1998 Midwest Symposium on*, pp. 374-378, 1998.
- [2] K. Kim, M. Franz, and B. Scholkopf, "Kernel Hebbian Algorithm for Single-Frame Super-Resolution," *Statistical Learning in Computer Vision (SLCV 2004)*, pp. 135-149, 2004.
- [3] S. Borman and R. Stevenson, "Spatial Resolution Enhancement of Low-Resolution Image Sequences - A Comprehensive Review with Directions for Future Research," Department of Electrical Engineering, University of Notre Dame, 1998
- [4] Sina Farsiu, M. Dirk Robinson, Student Member, Michael Elad, and Peyman Milanfar, Senior Member, "Fast and Robust Multiframe Super Resolution," *IEEE Transactions on Image Processing*, vol. 13, no. 10, pp. 1327-1344, 2004.
- [5] Claude E. Duchon, "Lanczos Filtering in One and Two Dimensions," *Journal of Applied Meteorology*, no. 18(8), pp. 1016-1022, 1979.
- [6] Andrew S. Glassner, Ken Turkowski and Steve Gabriel, "Filters for Common Resampling Tasks," *Graphics Gems*, pp. 147-165, 1990.
- [7] D. Barreto, L. D. Alvarez and J. Abad, "Motion Estimation Techniques in Super-Resolution Image Reconstruction. A Performance Evaluation," *Virtual observatory. Plate content digitalization, archive mining and image sequence processing*, Sofia, Bulgaria, vol. 1, pp. 254-268, 2006.
- [8] I. Richardson, "H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia," Chichester: John Wiley & Sons Ltd., 2003.
- [9] Lisa Gottesfeld Brown, "A survey of image registration techniques," *ACM Computing Surveys (CSUR)*, vol. 24, iss. 4, pp. 325-376, 1992.
- [10] E. Aptoula, S. Lefevre and C. Ronse, "A hit-or-miss transform for multivariate images," *Pattern Recognition Letters*, vol. 30, iss. 8, pp. 760-764, 2009.
- [11] M. Khosravi and R. Schafer, "Template matching based on a grayscale hit-or-miss transform," *IEEE Transactions on Image Processing*, vol. 5, no. 5, pp. 1060-1066, 1996.
- [12] B. Perret, S. Lefevre and Ch. Collet, "A robust hit-or-miss transform for template matching applied to very noisy astronomical images," *Pattern Recognition*, vol. 42, no. 11, pp. 2470-2480, 2009.
- [13] A. M. Tekalp, M. K. Ozkan, and M. I. Sezan, "High-resolution Image Reconstruction from Lower-resolution Image Sequences and Space-varying Image Restoration," *Proceedings of the IEEE international conference on acoustics, speech, and signal process*, vol. 3, pp. 169-172, 1992.
- [14] T. Komatsu, T. Igarashi, K. Aizawa, and T. Saito, "Very high resolution imaging scheme with multiple different aperture cameras," *Signal Processing: Image Communication*, vol. 5, no. 5-6, pp. 511-526, 1993.
- [15] Barbara Zitova, Jan Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977-1000, 2003.

ABOUT THE AUTHOR

Vadim Vashkelis is a Ph.D. at St. Petersburg State Polytechnical University. His contact email is vashkelis@gmail.com.

Natalia Trukhina is a graduate at St. Petersburg State University of Aerospace Instrumentation. Her contact email is ntrukhina@gmail.com.

Ivan Chirkov is a graduate at St. Petersburg State Polytechnical University. His contact email is chirkov.ivan@gmail.com.

Robust Space/Spatial-Frequency based Filtering of Images in the presence of heavy tailed noise

Irena Orović, Nikola Žarić, Srdjan Stanković
 Faculty of Electrical Engineering
 University of Montenegro, 81000 Podgorica, Montenegro
 {irenao, zaric, srdjan}@ac.me

Abstract

A robust space/spatial-frequency representation for analysis of two-dimensional signals corrupted by heavy tailed noise is proposed. The L-estimate forms of robust short-time Fourier transform and spectrogram for 2D signals are introduced. They are used to define a non-stationary space-varying filtering procedure in the space/spatial-frequency domain. The efficiency of the proposed procedure is tested on the examples with interferogram images affected by a mixed Gaussian and impulse noise.

Keywords: *robust space/spatial-frequency distributions, L-estimate spectrogram, robust space-varying filtering.*

1. INTRODUCTION

In many real applications, such as communications, signals are often corrupted by the noise. Depending on its statistics, the noise is usually Gaussian, impulse, or the mixture of these two that is generally considered as a heavy tailed noise. The heavy tailed noise is present in images as film grain noise, photoelectronic noise, salt and pepper noise, etc.

Analysis and processing of stationary signals is performed either in the time/space or in the frequency domain. However, for signals that exhibit highly non-stationary characteristics, the joint time-frequency domain should be used [1],[2], or equivalently the space/spatial-frequency domain in the case of two-dimensional signals [3]. Thus, an efficient approach in image filtering could be obtained by using the concept of space-varying filtering based on the space/spatial-frequency representation [4]-[7]. However, the standard time-frequency and space/spatial-frequency representations produce poor results in the presence of heavy tailed noise. This kind of noise requires a special approach and formulation of the robust forms. Therefore, various robust forms of time-frequency distributions have been introduced for one-dimensional signals [8]-[9], for instance, the robust spectrogram and short-time Fourier transform. It has been shown that the L-estimation signal representation provides the best results for the mixture of Gaussian and impulse noises.

This paper represents an extension of robust approaches to the space/spatial-frequency representations used for non-stationary two-dimensional signals (e.g. images). The two-dimensional forms of L-estimate short-time Fourier transform and L-estimate spectrogram are introduced. Based on the robust representations of noisy images, a procedure for robust space-varying filtering is proposed. A suitable filter support function is defined by applying an energy floor to the L-estimate spectrogram. This approach provides successful image filtering, recovering the image quality.

The paper is organized as follows. The robust L-estimate forms of two-dimensional short-time Fourier transform are proposed in Section II. The robust space-varying filtering approach is proposed in Section III. The experimental results are presented in Section IV, while the concluding remarks are given in Section V.

2. TWO-DIMENSIONAL L-ESTIMATE FORMS OF THE SHORT-TIME FOURIER TRANSFORM AND THE SPECTROGRAM

Time-frequency distributions have been used for analysis of non-stationary signals, corrupted by noise. If the noise has a heavy tailed probability density function, the robust distribution forms should be used. In the sequel, the robust space/spatial-frequency distribution for two-dimensional signals is defined.

Consider the two-dimensional noisy signal:

$$x(n_1, n_2) = s(n_1, n_2) + n(n_1, n_2), \quad (1)$$

where $s(n_1, n_2)$ is a complex-valued two-dimensional signal corrupted with complex-valued noise $v(n_1, n_2)$. We may assume that the noise represents a mixture of Gaussian and impulse noise that usually appears in real applications. In order to provide an efficient space/spatial-frequency analysis, the L-estimate robust short-time Fourier transform (STFT) can be used [10]. Following the concepts introduced for one-dimensional signals, the two-dimensional L-estimate robust STFT can be defined as follows:

$$STFT_L(n_1, n_2, k_1, k_2) = \sum_{p=-N/2}^{N/2-1} \sum_{q=-N/2}^{N/2-1} a_p a_q (r_{p,q}(n_1, n_2, k_1, k_2) + j \cdot i_{p,q}(n_1, n_2, k_1, k_2)), \quad (2)$$

where,

$$\begin{aligned} r_{p,q}(n_1, n_2, k_1, k_2) &\in R(n_1, n_2, k_1, k_2), \\ R(n_1, n_2, k_1, k_2) &= \{ \text{Re}(x_{n,m}), m_1, m_2 \in [-N/2, N/2] \}, \\ i_{p,q}(n_1, n_2, k_1, k_2) &\in I(n_1, n_2, k_1, k_2), \\ I(n_1, n_2, k_1, k_2) &= \{ \text{Im}(x_{n,m}), m_1, m_2 \in [-N/2, N/2] \}, \end{aligned} \quad (3)$$

and

$$x_{n,m} = x(n_1 + m_1, n_2 + m_2) e^{-j2\pi(m_1 k_1 + m_2 k_2)/N}. \quad (4)$$

The elements: $r_{p,q}(n_1, n_2, k_1, k_2)$ and $i_{p,q}(n_1, n_2, k_1, k_2)$ are sorted in non-decreasing order as:

$$\begin{aligned} r_{p,q}(n_1, n_2, k_1, k_2) &\leq r_{p,q+1}(n_1, n_2, k_1, k_2) \\ r_{p,q}(n_1, n_2, k_1, k_2) &\leq r_{p+1,q}(n_1, n_2, k_1, k_2) \end{aligned}$$

and

$$\begin{aligned} i_{p,q}(n_1, n_2, k_1, k_2) &\leq i_{p,q+1}(n_1, n_2, k_1, k_2) \\ i_{p,q}(n_1, n_2, k_1, k_2) &\leq i_{p+1,q}(n_1, n_2, k_1, k_2) \end{aligned}$$

In order to efficiently remove impulses and preserve good resolution, the weighting coefficients a_p and a_q are designed in analogy with coefficients of α -trimmed filter used for image filtering in the space domain. For one-dimensional case these coefficients are given as [10]:

$$a_p = a_q = \begin{cases} \frac{1}{N(1-2\alpha)+4\alpha}, & \text{for } p, q \in [(N-2)\alpha, \alpha(2-N)+N-1] \\ 0, & \text{elsewhere,} \end{cases} \quad (5)$$

where N is even, while the parameter α takes values within the range $[0, 1/2]$. In order to clarify the influence of the parameter α value to the noise reduction let us consider the illustration in Figure 1. Namely, after the sorting operation the coefficients corrupted by impulse noise will be located at the beginning and at the end of the sorted sequence. The noise-free coefficients and coefficients corrupted by Gaussian noise will be in the middle part of the sorted sequence. In order to remove the impulse noise, the weighting coefficients a_p and a_q should be zero in the regions where impulse noise exist. Also, it is well known that standard average filter provide good result in the presence of Gaussian noise. Thus, to remove the Gaussian noise the average value of the middle part of the sorted sequence should be calculated.

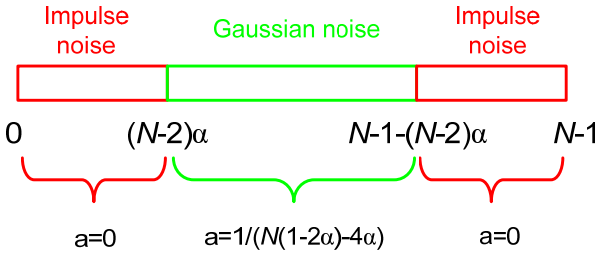


Figure 1. Influence of parameter α to the noise reduction

Higher α provides better reduction of impulse noise, while smaller α improves spectral characteristics. Thus, the value of the parameter α should be carefully chosen. As a special cases, for $\alpha=0$ and $\alpha=1/2$ the standard and median distribution forms follow, respectively.

Based on the real and imaginary parts of the L-estimate STFT, the two-dimensional L-estimate spectrogram is defined as:

$$\begin{aligned} SPEC_L(n_1, n_2, k_1, k_2) &= \\ &= \text{Re}\{STFT_L(n_1, n_2, k_1, k_2)\}^2 + \text{Im}\{STFT_L(n_1, n_2, k_1, k_2)\}^2, \end{aligned} \quad (6)$$

where $STFT_L$ is the L-estimate short-time Fourier transform.

Calculation of the 2D L-estimate STFT

The direct implementation of above equations may result in complex and computationally demanding procedure for calculation of the 2D L-estimate STFT. In order to define more appropriate calculation procedure, let us start with realization of the 2D Fourier transform. It is well known that 2D Fourier transform of a signal $x(n_1, n_2)$ can be calculated as:

$$FFT2(x(n_1, n_2)) = FFT(FFT(x(n_1, n_2))^T)$$

where T denotes transposition operator. Since the STFT represent the windowed version of the Fourier transform, the similar concept can be used for calculation of the 2D L-estimate STFT. The procedure for 2D L-estimate STFT calculation based on the 1D L-estimate STFT is presented in Figure 2. The first step is to multiply image rows with corresponding basis functions, which are the same as in the Fourier transform. In order to distinguish coefficient corrupted by impulse noise and coefficients corrupted by Gaussian noise, the obtained products are sorted. The sorted sequences are further multiplied with weighting coefficients a_q and summed, to obtain the noise free time-frequency representation of image rows. In order to obtain noise free space/spatial-frequency representation the same procedure is applied on the columns of time-frequency representation.

3. ROBUST SPACE/SPATIAL-FREQUENCY BASED FILTERING

The stationary filtering can be used in the case when signal and noise do not overlap in space and frequency domain. Otherwise, the space invariant filtering produces poor results. In these circumstances, the nonstationary space-varying filtering can be used [4]. Namely, the signal and the noise can be separated within the joint space/spatial-frequency domain, which is a basis for an efficient space/spatial-frequency filtering procedure. However, if the present noise is mixed Gaussian and impulse noise, the standard distributions cannot provide good representation [10]. Namely, by using standard distributions (as in [4]), the coefficient affected by impulse noise could be of higher strength than true signal coefficients. It may result in inappropriate filter design, and consequently in poor quality of filtered image. Thus, in the presence of mixed Gaussian and impulse noises, the robust form of space/spatial-frequency representation should be used to derive an optimal nonstationary filter.

In analogy with one-dimensional case, the nonstationary 2D space-varying filtering can be generally written in the form [4]:

$$(Hs)(x, y) = \int_a^T \int_b^T h(x + \frac{a}{2}, x - \frac{a}{2}, y + \frac{b}{2}, y - \frac{b}{2}) r(w(a, b)s(x + a, y + b)) da db, \quad (7)$$

where $h(x, y, a, b)$ represents the impulse response of the space-varying 2D filter, while s and w are signal and window function, respectively. The 2D window should be separable and symmetrical. The support function can be defined as follows:

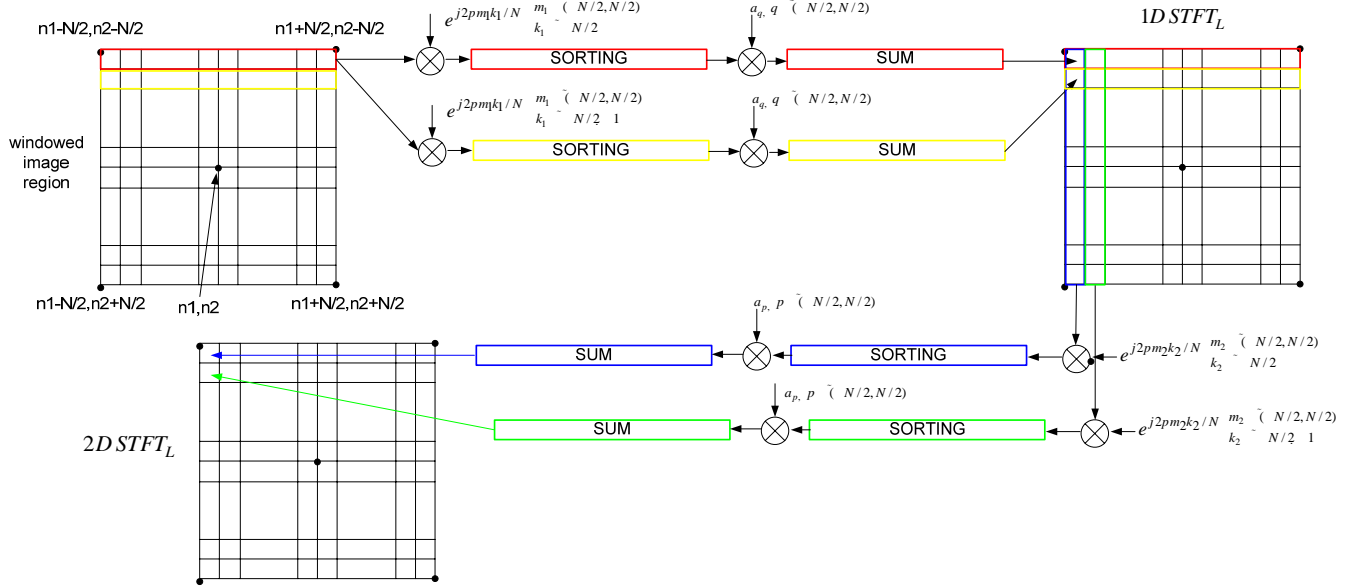


Figure 2. Algorithm for realization of 2D L-estimate STFT based on 1D L-estimate STFT

$$L_H(x, y, \omega_x, \omega_y) = \int_{\alpha} \int_{\beta} h\left(x + \frac{\alpha}{2}, x - \frac{\alpha}{2}, y + \frac{\beta}{2}, y - \frac{\beta}{2}\right) \times e^{-j(\alpha\omega_x + \beta\omega_y)} d\alpha d\beta. \quad (8)$$

By using the L-estimate STFT, the robust form of the space-varying filter can be obtained. Namely, based on the Parserval theorem, the output of the L-estimate space-varying filter is defined according to:

$$(H_s^L)(x, y) = \frac{1}{4\pi^2} \int_{\omega_x} \int_{\omega_y} L_H(x, y, \omega_x, \omega_y) STFT_L(x, y, \omega_x, \omega_y) d\omega_x d\omega_y. \quad (9)$$

The support function $L_H(x, y, \omega_x, \omega_y)$ is defined as Weyl symbol mapping to the space/spatial-frequency domain. Assuming that the signal components lie inside the two-dimensional region R_f , while the noise is outside this region, the support function $L_H(x, y, \omega_x, \omega_y)$ is usually defined as:

$$L_H(x, y, \omega_x, \omega_y) = \begin{cases} 1 & \text{for } (x, y, \omega_x, \omega_y) \in R_f \\ 0 & \text{for } (x, y, \omega_x, \omega_y) \notin R_f \end{cases}. \quad (10)$$

The discrete form of (10) that is suitable for practical realizations can be written as follows:

$$(H_s^L)(n_1, n_2) = \sum_{k_1} \sum_{k_2} L_H(n_1, n_2, k_1, k_2) STFT_L(n_1, n_2, k_1, k_2) \quad (11)$$

Therefore, to perform the space-varying filtering we should calculate the L-estimate 2D STFT and determine the support

function L_H . Obviously, a precise determination of L_H is related to the precise determination of the region R_f .

Hence, in order to define the support function that will allow efficient noise filtering, we consider the set of points D defined by using the robust L-estimate 2D spectrogram:

$$D = \{(n_1, n_2, k_1, k_2) \mid SPEC_L(n_1, n_2, k_1, k_2) > \xi\} \quad (12)$$

where ξ represents the energy floor. Therefore, D contains only the positions of relevant signal components within the space/spatial-frequency representation of noisy signal. The energy floor is defined as a portion of the maximal value of L-estimate 2D spectrogram:

$$\xi = \lambda \log \left(\max_{k_1, k_2} \{SPEC_L(n_1, n_2, k_1, k_2)\} \right) \quad (13)$$

Finally, the support function of the nonstationary space-varying filter can be defined in the form:

$$L_H(n_1, n_2, k_1, k_2) = \begin{cases} 1 & \text{for } (n_1, n_2, k_1, k_2) \in D \\ 0 & \text{for } (n_1, n_2, k_1, k_2) \notin D \end{cases}.$$

The proposed robust space-varying filtering of images can be summarized through the following steps:

1. Place a center of a two-dimensional window in the pixel on the position (n_1, n_2)
2. Calculate the L-estimate STFT ($STFT_L$) for windowed part of signal

3. Calculate the L-estimate spectrogram $SPEC_L$ based on the $STFT_L$
4. Choose λ and compute the energy floor ξ
5. Determine the support function L_H by using $SPEC_L$ and ξ
6. Calculate the filter output by using L_H and $STFT_L$
7. Repeat the procedure for each image pixel.

A simple block scheme of the filtering procedure is illustrated in Figure 3.

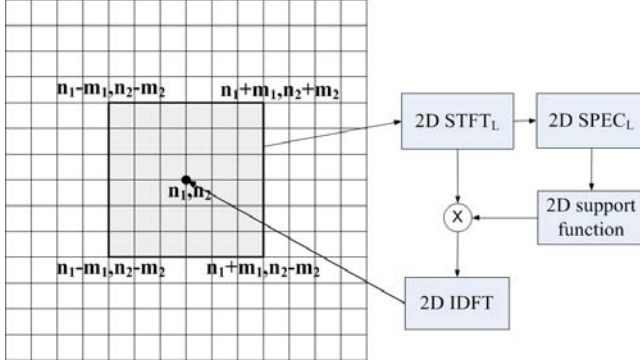


Figure 3. The scheme of the robust space-varying filtering

4. EXPERIMENTAL RESULTS

The proposed approach for the robust space/spatial-frequency based filtering is tested on the numerical example. The analytical image is given in the form:

$$s(n_1, n_2) = \cos\left(20\pi(n_1 - 0.75)^2 + 22\pi(n_2 - 0.75)^2\right) + 0.5e^{j(-100\cos(\pi n_1/2) - 100\cos(\pi n_2/2))}. \quad (14)$$

Note that the considered signal has a form of interferogram image that appears in the optics. The noise is defined as:

$$v(n_1, n_2) = 0.5v_1^3(n_1, n_2) + 0.5jv_2^3(n_1, n_2) \quad (15)$$

where $v_1(n_1, n_2)$ and $v_2(n_1, n_2)$ are mutually independent Gaussian noises (zero mean with variance equal to 1).

The original noise-free signal $s(n_1, n_2)$ is shown in Figure 4.a. The signal: $x(n_1, n_2) = s(n_1, n_2) + v(n_1, n_2)$ is illustrated in Figure 4.b. Note that the signal $x(n_1, n_2)$ is quite affected by the noise. Firstly, consider the filtering procedures in the spatial domain: the stationary median filtering and filter proposed in [11]. The mask 3×3 is used for both filters. As shown in Figure 4.c, the median filter produces poor results, and the image quality is even worse than before the filtering operation. The result obtained by filter

proposed in [11] (Figure 4.d) are better compared with median filter, but the quality of filtered image is not good especially in the regions with high non-stationarity. Further, we analyzed the procedures in the frequency domain: low pass filtering with cutoff frequency $f = f_{max}/2$ (provide reduction of Gaussian noise) and band-stop filtering with boundaries $f_h = 3f_{max}/4$ and $f_l = f_{max}/4$ (provide reduction of impulse noise [12]). The results are shown in Figure 4.e and Figure 4.f, respectively. Note that the noise remains present in both cases, and the quality of filtered image is not satisfactory. The best results are obtained by using the proposed robust space-varying filtering (Figure 4.g). Note that the robust space-varying filtering provides very successful results, providing good quality of filtered image that is close to the quality of original noise-free image.

The proposed robust space-varying filtering is performed by using the L-estimate STFT and spectrogram. The L-estimate forms are calculated by using parameter $\alpha = 3/8$. Namely, this value provides satisfying trade-off between noise reduction and distribution concentration. The window size is 64×64 . The energy floor is calculated by using $\lambda = 0.8$.

5. CONCLUSION

The L-estimation based space-spatial-frequency representation is proposed. Namely, for analysis of two-dimensional signals embedded in heavy tailed noise, the robust forms of multidimensional short-time Fourier transform and spectrogram are introduced. Based on these representations, the concept of non-stationary space-varying filtering is defined. Since it is based on the robust space/spatial-frequency analysis, the proposed procedure provides efficient filtering of a mixed Gaussian and impulse noise. Even for a high amount of noise, the filtered image retains good quality that is close to the original non-noisy image.

6. REFERENCES

- [1] B. Boashash, *Time-Frequency Analysis and Processing*, Elsevier, 2003, Amsterdam
- [2] B. Boashash, "Estimating and interpreting the instantaneous frequency of a signal-Part 1: Fundamentals", *Proc. of the IEEE*, vol. 80, no. 4, pp. 520-538, 1992.
- [3] S. Stankovic, L.J. Stankovic, Z. Uskokovic, "On the local Frequency, Group Shift and Cross Terms in Some Multidimensional Time-Frequency Distributions; A Method for Multidimensional Time-Frequency Analysis," *IEEE Transactions on Signal Processing*, vol. 43, no. 7, July 1995.
- [4] L.J. Stankovic, S. Stankovic, I. Djurovic, "Space/Spatial-Frequency Based Filtering," *IEEE Transaction on Signal Processing*, vol. 48, no. 8, Aug. 2000.
- [5] L.J. Stanković, "On the time-frequency analysis based filtering," *Annales des Telecommunications*, vol.55, No.5-6, May 2000.
- [6] H. Kirchauer, F. Hlawatsch, W. Kozek, "Time-frequency formulation and design of nonstationary Wiener filters," *Proc. Int. Conf. ASSP*, pp.1549-1552, 1995.
- [7] G. Matz, F. Hlawatsch, W. Kozek, "Generalized Evolutionary Spectral Analysis and the Weyl Spectrum of Nonstationary Random Processes," *IEEE Transaction on Signal Processing*, vol. 45, No. 6, pp.1520-1534, June 1997.

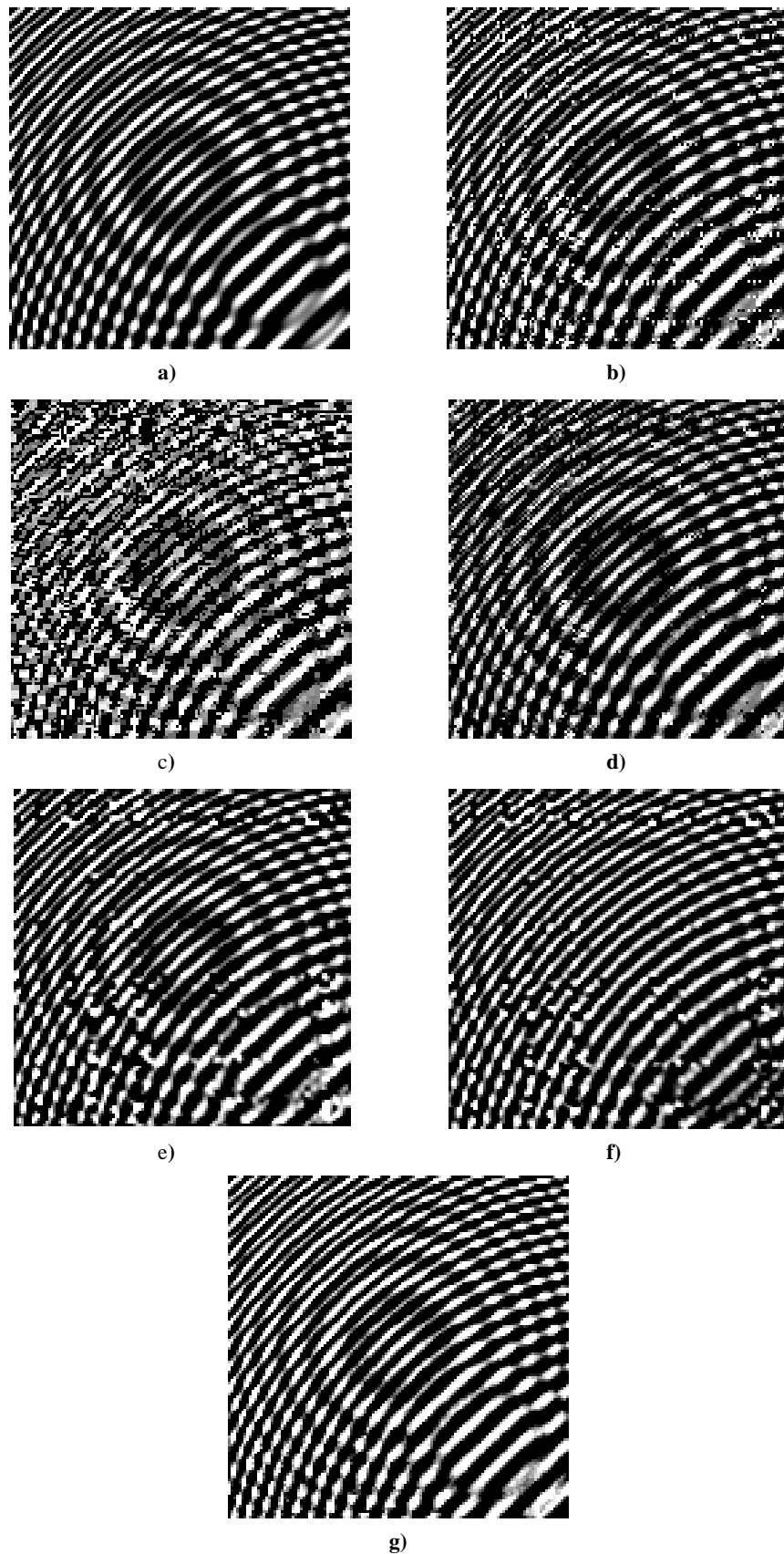


Figure 4: a) original image, b) image corrupted by the mixture of Gaussian and impulse noise, images filtered by: c) median filter, d) filter proposed in [11], e) low pass filter, f) band stop filter, g) space-varying filter based on the L-estimate spectrogram

- [8] V. Katkovnik, I. Djurović, LJ. Stanković, "Instantaneous frequency estimation using robust spectrogram with varying window length", *AEU Int. Journal Electron. Commun.*, vol. 54, no. 4, pp.193-202, 2000.
- [9] I. Djurović, LJ. Stanković, B. Barkat, "Robust Time-Frequency Distributions based on the robust short time Fourier transform", *Annales des Telecommunications*, vol. 60, no. 5-6, pp. 681-697, May-June 2005.
- [10] I. Djurović, LJ. Stanković, J. F. Böhme, "Robust L-estimation based forms of signal transforms and time-frequency representations", *IEEE Trans. Signal Processing*, vol. 51, no. 7, pp.1753-1761, July 2003.
- [11] N. Alajlan, M. Kamel, E. Jermigam, "Detail preserving impulsive noise removal," *Signal Processing: Image Communication*, vol. 19, pp. no. 10, 993-1003, Nov. 2004.
- [12] R. C. González, R. E. Woods, *Digital image processing*, 3th Edition, Prentice Hall, 2007.

About the authors

Srdjan Stanković is a Professor at University of Montenegro, Faculty of Electrical Engineering. His contact email is srdjan@ac.me.

Irena Orović, Ph.D, University of Montenegro, Faculty of Electrical Engineering. Her contact email is irenao@ac.me.

Nikola Žarić is a Ph.D. student at University of Montenegro, Faculty of Electrical Engineering. His contact email is zaric@ac.me.

A Novel Smart Bilateral Filter for Ringing Artifacts Removal in JPEG2000 Images

Vladimir Khryashchev, Ilya Apalkov, Lev Shmaglit
 Digital Circuits and Signals Laboratory
 Yaroslavl State University, Yaroslavl, Russia
 connect@piclab.ru

Abstract

Ringing artifacts arise near edges in highly compressed JPEG2000 images. This paper proposes original image ringing detection and suppression techniques. On the first stage edge detection and ringing estimation algorithms are applied to calculate the ringing map of a compressed input image. On the second stage this ringing map is utilized to adjust the smart bilateral filter for effectively protecting the image detail from blur effect. Experimental results show that the proposed smart bilateral filter allows to remove ringing artifacts while introducing no additional blurring and thus to increase visual quality of compressed JPEG2000 images.

Keywords: image compression artifacts, JPEG2000, ringing, blurring, bilateral filter, ringing map

1. INTRODUCTION

The work presented in this paper is motivated by the ringing distortion concealment for highly compressed JPEG2000 images. JPEG2000 is the latest international still image compression standard. It offers a strong compression performance with many practical features [1]. Considerable fidelity loss, namely compression ratio of about 40:1 or even 100:1, generally is acceptable in JPEG2000 compression. In this paper we propose a

post-processing technique that can efficiently improve visual quality for highly compressed JPEG2000 images.

Unlike the DCT-based compression methods, JPEG2000 doesn't produce blocking artifacts, but it introduces ringing and blurring artifacts into an image. Ringing effect (Gibbs phenomenon) is caused by the quantization or truncation of the high frequency coefficients. Moreover, it appears as distortion along sharp edges in an image. Ringing artifacts are, in general, more difficult to characterize and remove than block-transform compression artifacts. Blurring is another artifact resulting from the absence of high frequencies in highly compressed images. It appears around the sharp edges, and all image details become blurred. This effect is very similar to ringing artifact, and sometimes it is hard to distinguish between them. The difference between these two effects is that they appear on different sides: horizontal or vertical (Fig. 1).

One of the main problems of ringing artifacts removal is to detect the presence of ringing effect and to estimate the necessary ringing suppression level. A few algorithms which estimate ringing and blurring level in compressed images have been proposed recently [2-6]. In [2], ringing effect is modeled by ideal low-pass filter which truncates high frequency data. Image ringing level is estimated using the analysis of ringing level of 1D normal cross-sections of image edges obtained by an edge detection algorithm. A full- and no-reference blur metric and a full-reference ringing metric are proposed in [3].

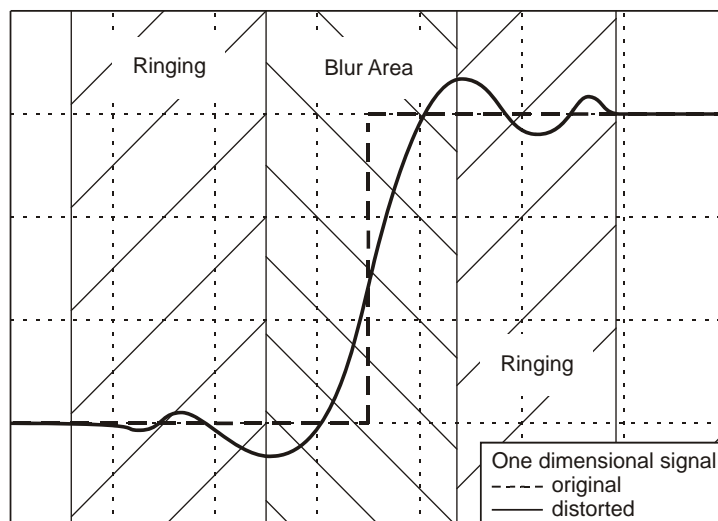


Fig. 1. Ringing and edge blur in one dimensional signal

The proposed metrics are defined in the spatial domain and are based on the analysis of the edges in an image. The ringing metric is defined as maximum of the differences between pixel values of the reference image and the processed image in the edge neighborhood. In [4], wavelet decomposition is used and ringing effect is measured for JPEG2000 compression as a difference between correlations of neighbor coefficients of different wavelet subbands. The work [5] does not introduce a ringing estimation method, but it presents an algorithm to find regions where the ringing effect is the most visible. It is based on luminance masking and texture masking as typical for the human visual system. In [6], a no-reference ringing detection method using Gabor filtering was suggested. It shows good results but it fails if image contains periodic structures like fence, geometrical textures, etc.

Several wavelet-based de-ringing algorithms have been proposed in [7-10]. Fan and Cham [7] perform edge reconstruction on wavelet-based compression images using a model-based method. However it cannot achieve image enhancement when the image carries less edge distortion or deadly distorted edges. In [8], a new edge directed filter for wavelet-based images is proposed. It adopts a Partial Difference Equations based (or PDE-based) iterative algorithm. The method can get image quality enhancement, although the computation cost is high. Nosratinia [9] employs further compression to the shifted versions of the received image and selectively cuts off the high frequency signal of them. In [10], adaptive nonlinear filtering method to reduce ringing artifacts in the received image is proposed. The coefficients around edge areas are selectively processed in spatial domain and each decomposition level. The shift variance character of decimated wavelet transform in JPEG2000 is utilized. Several shift versions of the received images are used to estimate the variance of the compression noise.

The problem of image de-ringing after JPEG2000 compression is also considered in [11-15]. The de-ringing algorithm proposed in [11] utilizes a quad-tree partitioning scheme for post-processing the reconstructed image in a spatially varying manner. Voting strategy is used to determine a set of morphological filters for reducing the ringing artifacts. In [12], each pixel value is replaced with a function of the values of neighboring pixels that are within a specified window. To avoid conflict with the above goal, that is smooth shade regions and sharp edges, the de-ringing algorithm uses a number of adaptive noise reduction algorithms. In [13], binary morphological operators are used to isolate the regions of an image where the ringing artifact is most prominent to the human visual system. Then a gray-level morphological nonlinear smoothing filter is applied to the unmasked regions of the image under the filtering mask to eliminate ringing within the constraint region. An adaptive nonlinear diffusion method to reduce ringing artifacts is proposed in [14]. The diffusion coefficient is defined as the linear combination of three membership functions that correspond to the different gradient scales. The parameters are tuned using a stochastic optimization technique so that the diffusion coefficient can adapt to the given image data to achieve better restoration performance. In [15], a maximum likelihood approach to the ringing artifact removal problem is presented. It employs a parameter estimation method based on the k-means algorithm with the number of clusters determined by a cluster separation measure.

Despite the fact that essential progress has been reached, the problem of ringing artifact removal still remains actual. None of

the proposed no-reference ringing estimation metrics can effectively detect ringing artifacts and measure their level in general case. Most of the proposed ringing reduction schemes introduce additional blurring into processed images. Others have too high computational cost. That's why we propose ringing detection and ringing removal algorithms which are deprived of the described above lacks. Our scheme achieves ringing reduction while preserving the same level of blurring artifacts.

The rest of the paper is organized as follows. In section 2, we introduce a blurring estimation metric. In section 3, we introduce a method of ringing level estimation in compressed JPEG2000 images. In section 4, we propose the smart bilateral filter for ringing artifacts removal. In section 5, we present experimental results of applying of the proposed filter. Section 6 concludes the paper.

2. BLURRING METRIC CALCULATION

JPEG2000 encoder uses separable wavelet transform with biorthogonal Daubechies 9/7 wavelet-basis [1] for image compression. This transform is applied consistently first to rows and then to columns of an image. That's why it is quite enough to consider the influence of JPEG2000 compression algorithm only on image rows.

Let's consider an image row $\{x_1, x_2, \dots, x_N\}$, where x_i – image pixel luminance value. To detect edges on this row we apply Sobel filter. The result of this operation is a set of values $\{p_1, p_2, \dots, p_K\}$, where $K < N$, which indicates the position of edges in the row.

For each point p_k we find the location of the local luminance extrema closest to the left – l_k , and the local luminance extrema closest to the right – r_k . The local blur metric for one edge we define as $w_k = r_k - l_k$. Edge width on an image is numerically equal to w_k .

The Blurring Metric (BM) of an image we then define as the average value of local blur metrics calculated for each of M rows:

$$BM = \frac{\sum_{j=1}^M \sum_{k=1}^K w_{k,j}}{\sum_{j=1}^M K_j} \quad (1)$$

On the basis of experimental results the top threshold value of the blurring metric, which is not exceeded for natural compressed images, has been calculated. Taking this threshold value into account, normalized blurring measure has the following form:

$$0 \leq \frac{BM}{20} \leq 1. \quad (2)$$

The proposed algorithm lends itself to both a full-reference and a no-reference implementation. The difference between them is that a set of values $\{p_1, p_2, \dots, p_K\}$, indicating the position of edges, is calculated for original images in the case of full-reference metric, and for decoded images in the case of no-reference metric.

3. RINGING METRIC CALCULATION

The results acquired on the stage of blurring level estimation are used to construct the ringing metric. Let's define the local ringing metric for the edge with position p_k as:

$$(w_{ring})_k = \left| \max(x_i - \tilde{x}_i) - \min(x_i - \tilde{x}_i) \right| \times |w_f - l_k| + \left| \max(x_j - \tilde{x}_j) - \min(x_j - \tilde{x}_j) \right| \times |w_f - r_k|, \quad (3)$$

$$i \in [p_k - w_f, l_k],$$

$$j \in [r_k, p_k + w_f]$$

where x_i – original image pixel luminance value, \tilde{x}_i – decoded image pixel luminance value, w_f – the fixed ringing width which was acquired during the analysis of distortion factors influence on the ideal signal. On the contrary to the blurring metric's calculation, the point p_k must be located strictly in the middle of the edge, i.e. $|l_k - p_k| = |r_k - p_k|$.

To estimate the total ringing level we calculate the average value of received local ringing metrics. The sum is taken only over those p_k for which local ringing exists. In this case, the resulting Ringing Metric (RM) will show the amplitude of ringing artifacts, taking into account the size of a region where they appear, and will not depend on other types of distortions:

$$RM = \frac{\sum_{j=1}^M \sum_{k=1}^K (w_{ring})_{k,j}}{\sum_{p_k} 1}. \quad (4)$$

On the basis of experimental results the top threshold value of the ringing metric, which is not exceeded for JPEG2000 compressed images, has been calculated:

$$0 \leq \frac{RM}{512} \leq 1. \quad (5)$$

As the ringing estimation algorithm uses the original image, it supposes only full-reference implementation. It is necessary to

note that the ringing metric includes all operations for the blurring metric's calculation.

4. SMART BILATERAL FILTER FOR RINGING REDUCTION

Let's consider the influence of bilateral, median and linear low-pass filters on ringing artifacts. We define the level of ringing artifacts reduction as the percentage difference between the ringing metric value before filtration and the ringing metric value after filtration. By averaging the level of ringing artifacts reduction on the whole test set (10 images); we acquire the average level of ringing artifacts reduction. Results of this test for each filter are presented in Table 1. Here CR stands for Compression Ratio of test images.

Results of filter comparison show that the best reduction of ringing artifacts can be acquired by the use of bilateral filter. However, the visual and numerical analysis of the processed images shows that bilateral filter introduces the greatest amount of blurring into an image. For elimination of this lack we will try to modify the scheme of bilateral filter, and to process not the whole image and even not all edges but only those fragments of an image where ringing is really visible. Actually it is necessary to construct a ringing map for an image and then to filter only those pixels which belong to this map.

On the first stage we detect all edges on the decoded image and find segments $[p_k - w_f/2, l_k) \cup (r_k, p_k + w_f/2]$ for each location of p_k . Each segment should be plotted on a map – an empty image with the same size as the input image.

As the ringing metric considers only vertical edges, we apply the same operations to horizontal edges. For this purpose we apply horizontal Sobel filter and find p_k for each column of input image. Then we consider the transposed column of the image, find segments for each p_k and plot them on the corresponding columns of the ringing map. It is necessary to note that acquired ringing map will contain vicinities of edges which are subject to blurring. Filtration of these areas will inevitably cause blurring increase. Hence we should delete vicinities of edges from the ringing map to preserve the level of blurring in the input image. To carry it out we unite all areas on which the blurring metric is calculated and subtract the received set of points from the ringing map.

Table 1. Ringing level reduction using low-pass filters

Filter	The average level of ringing artifact reduction (%)				
	CR=25	CR=35	CR=50	CR=75	CR=100
Bilateral	6.83	11.49	13.76	14.01	15.69
Median	5.37	8.52	9.88	9.84	10.70
Linear	3.29	6.17	7.90	8.15	8.76

Thus, the ringing map will contain pixels subject to ringing and will not contain pixels subject to blurring. The ringing map, containing areas of horizontally and vertically directed ringing, is imposed on an image. Examples of the ringing map calculated for several test images are presented in Fig. 2.

On the next stage of the proposed ringing reduction scheme we perform image filtration by bilateral filter. Filter mask is applied only to those pixels which belong to the ringing map. Let's define this algorithm as smart bilateral filter. The described scheme of image processing is presented in Fig. 3.



Fig. 2. Ringing maps for compressed test images (CR=17) “Barbara” (left), Scarlett (center), Peppers (right)

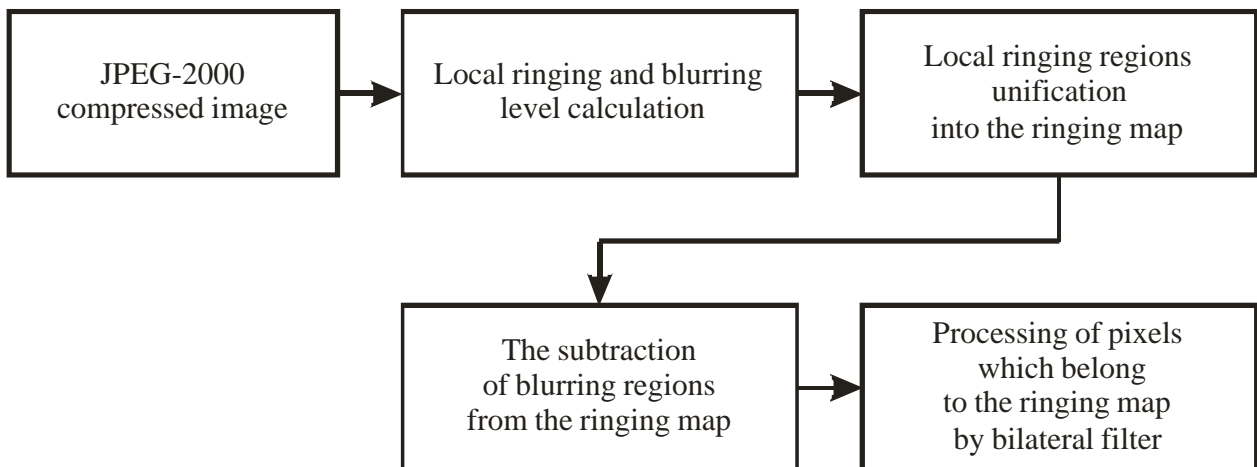


Fig. 3. Ringing detection and reduction algorithm

Smart bilateral filter has same parameters as classic bilateral filter [16, 17]: the mask size w , spatial mean square displacement σ_s , and luminance mean square displacement σ_l . For the purpose of testing, these parameters have been fixed to the following values: $w=5$, $\sigma_s=3$, $\sigma_l=0.5$. Pixel luminance values have been preliminary normalized to be distributed inside a segment [0, 1].

5. EXPERIMENTAL RESULTS

Comparative results of bilateral and smart bilateral filtering for test images “Lenna” and “Baboon”, concerning the proposed ringing and blurring metrics, are presented in Fig. 4. The received diagrams show that in the case of low compression (CR<20), the ringing reduction procedure leads to the opposite effect – natural fluctuations of luminance are blurred by bilateral filter. The growth of the ringing metric indicates that the algorithm’s application is incorrect in the case of low compression. Filtration of highly compressed images by smart bilateral filter gives high

level of ringing suppression with preservation of the same level of blurring. Smart bilateral filter efficiency increases for highly detailed images (such as “Baboon”). At the same time filtration by classic bilateral filter leads to the substantial growth of blurring in images and a slightly higher level of ringing suppression.

Evaluation of the proposed algorithm by standard quality metrics such as peak signal-to-noise ratio (PSNR) shows that classic bilateral filter reduces image quality compared to smart bilateral filter (Fig. 5). This is explained by the fact that classic bilateral filter removes ringing artifacts not by smoothing fluctuations of luminance but by blurring image edges.

Experimental results show that computing complexity of smart bilateral filter compared to that of bilateral filter decreases with the increase of image size (Fig. 6). For an image with 768 rows smart bilateral filter works 1.5 times faster compared to classic bilateral filter and for an image with 1024 rows – 2 times faster.

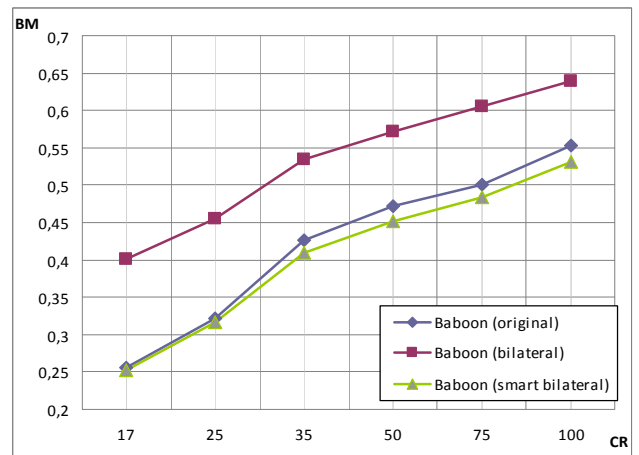
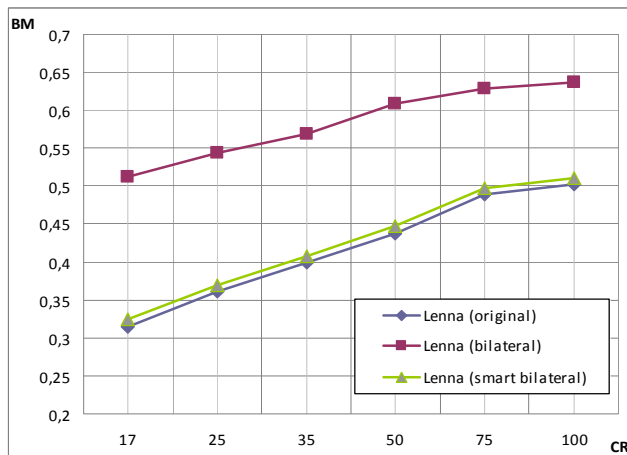
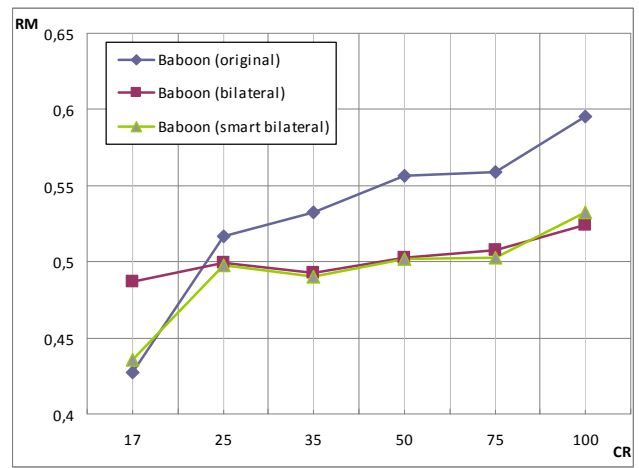
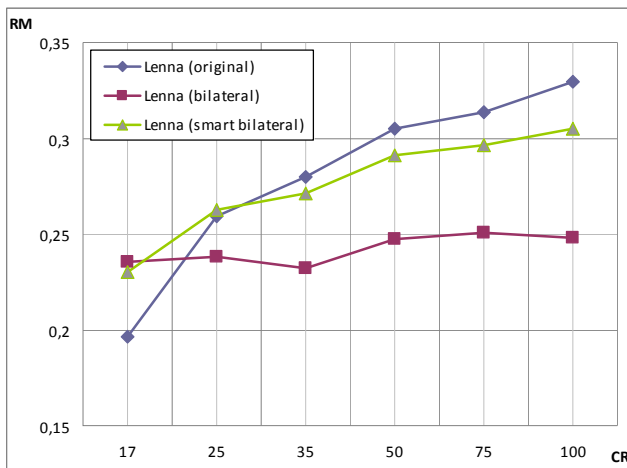


Fig. 4. Bilateral and smart bilateral filtration comparative results for “Lenna” and “Baboon” compressed images

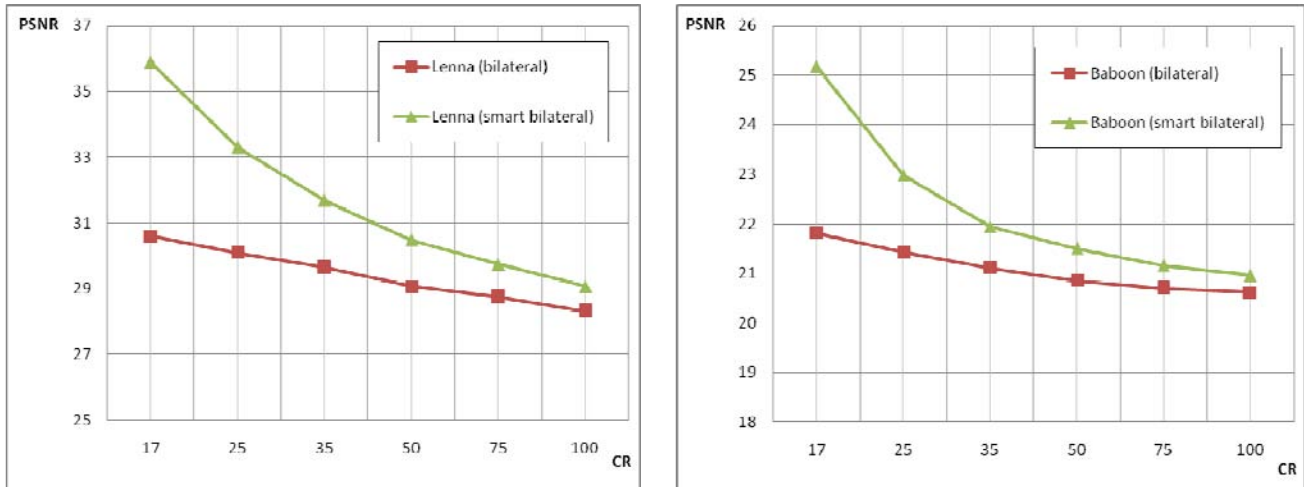


Fig. 5. Peak signal-to-noise ratio for “Lenna” and “Baboon” images processed by bilateral and smart bilateral filters

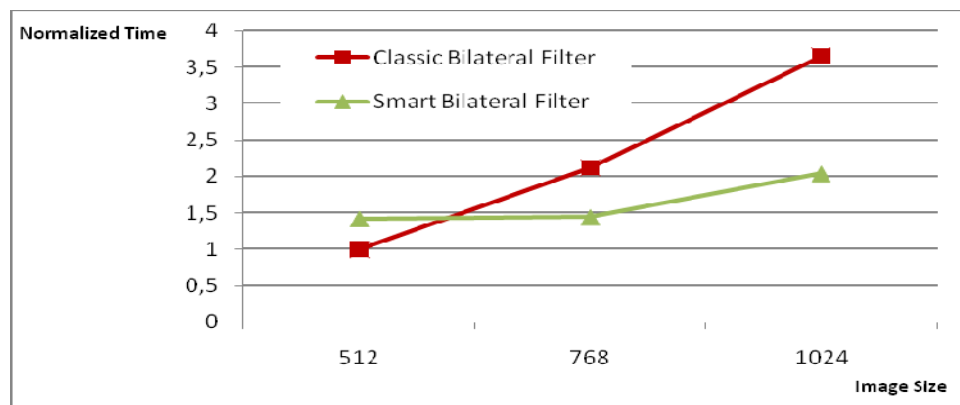


Fig. 6. Computing complexity comparison of bilateral and smart bilateral filtration

6. CONCLUSIONS

Blurring and ringing distortions have strong nonlinear dependence from each other. Suppression of one of these types of distortions by standard filters can lead to the growth of another type. Therefore there is an important problem of ringing artifacts reduction with the preservation of the level of compressed image blurring.

The proposed smart bilateral filter gives the solution of this problem in application to JPEG2000 images. The application of classic bilateral filter to the whole image requires much more time than the application of smart bilateral filter to image fragment belonging to the ringing map. Therefore computing complexity of smart bilateral filter is lower even taking into account calculations we perform to acquire the ringing map.

7. ACKNOWLEDGMENT

This work is executed under financial support of Russia Foundation of Basic Research (grant 10-08-01186-a).

8. REFERENCES

- [1] D. Taubman, M. Marcellin, “*JPEG2000: Image Compression Fundamentals, Standards, and Practice*”. Norwell, MA: Kluwer, 2001.
- [2] A. Nasonov, A. Krylov, “Scale-space method of image ringing estimation,” *Proceedings of IEEE International Conference on Image Processing (ICIP'09)*, pp. 2794-2797, 2009.
- [3] P. Marziliano, F. Dufaux, S. Winkler, T. Ebrahimi, “Perceptual blur and ringing metrics: application to JPEG2000,” *Signal Processing: Image Communication*, vol. 19, pp. 163-172, 2004.
- [4] M. Khambete, M. Joshi, “Blur and ringing artifact measurement in image compression using wavelet transform,” *Proceedings of World Academy of Science, Engineering and Technology*, vol. 20, pp. 183-186, 2007.
- [5] H. Liu, N. Klomp, I. Heynderickx, “Perceptually relevant ringing region detection method,” *Proceedings of 16th European Signal Processing Conference (EUSIPCO'08)*, 2008.

- [6] B. Zuo, "Perceptual ringing metric to evaluate the quality of images restored using blind deconvolution algorithms," *Optical Engineering*, vol. 48, № 3, pp. 037004-037004-9, 2009.
- [7] G. Fan, W. Cham, "Model-based Edge Reconstruction for Low Bit-rate Wavelet-Compressed Images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 120–132, 2000.
- [8] S. Ye, Q. Sun, E. Chang, "Edge Directed Filter based Error Concealment for Waveletbased Images," *Proceedings of IEEE International Conference on Image Processing (ICIP'04)*, vol. 2, pp. 809–812, 2004.
- [9] A. Nosratinia, "Postprocessing of JPEG2000 Images to Remove Compression Artifacts," *IEEE Signal Processing Letters*, vol. 10, pp. 296-299, 2003.
- [10] J. Fang, J. Sun, "Ringing Artifact Reduction for JPEG2000 Images," *Proceedings of International Conference on Intelligent Computing in Communication (ICIC'07)*, vol. 3, pp. 1026-1034, 2007.
- [11] Y. Chen, Y. Chang, W. Yen, "Design of a De-Ringing Filter for Wavelet-Based Compressed Image," *Proceedings of the 23rd International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC'08)*, pp. 1265-1268, 2008.
- [12] M. Shen, C. Kuo, "Artifact reduction in low bit rate wavelet coding with robust nonlinear filtering," *IEEE Second Workshop on Multimedia Signal Processing*, pp. 480-485, 1998.
- [13] S. Oguz, Y. Hu, T. Nguyen, "Image coding ringing artifact reduction using morphological post-filtering," *IEEE Second Workshop on Multimedia Signal Processing*, pp. 628-633, 1998.
- [14] S. Yao, W. Lin, Z. Lu, E. Ong, X. Yang, "Adaptive nonlinear diffusion processes for ringing artifacts removal on JPEG2000 images," *IEEE International Conference on Multimedia and Expo (ICME'04)*, vol. 1, pp. 691-694, 2004.
- [15] S. Yang, Y. Hu, D. Tull, T. Nguyen, "Maximum likelihood parameter estimation for image ringing artifact removal," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, № 8, pp. 963-973, 2001.
- [16] C. Tomasi, R. Manduchi, "Bilateral filtering for gray and color images," *Sixth International Conference on computer vision*, pp. 839 – 846, 1998.
- [17] M. Elad, "On the origin of the bilateral filter and ways to improve it," *IEEE Transactions on image processing*, vol. 11, № 10, pp. 1141-1151. 2002.

About the author

Vladimir Khryashchev is a docent at Yaroslavl State University, Laboratory of Digital Circuits and Signals. His contact email is vladimir@piclab.ru.

Ilya Apalkov is a docent at Yaroslavl State University, Laboratory of Digital Circuits and Signals. His contact email is apalkoff@piclab.ru.

Lev Shmaglit is a PhD student at Yaroslavl State University, Laboratory of Digital Circuits and Signals. His contact email is lev_shmaglit@yahoo.com.

Edge Detection Using Reaction-Diffusion Equation with Variable Diffusion Coefficient

Alexander M. Denisov, Andrey S. Krylov, Victoria Yu. Medvedeva
Faculty of Computational Mathematics and Cybernetics
Lomonosov Moscow State University
 den@cs.msu.ru , kryl@cs.msu.ru, medvedevav_88@mail.ru

Abstract

New edge detection method using FitzHugh-Nagumo reaction-diffusion system with image-dependent variable diffusion coefficient is suggested. It avoids false edges arising for noisy images and images with ringing in the case of commonly used constant diffusion coefficient.

Keywords: edge detection, FitzHugh-Nagumo model, variable diffusion coefficient

1. INTRODUCTION

Diffusion filtering methods are an effective tool for edge detection. These methods usually use as a criterion the zero-crossing of the second partial derivatives. As an example, Marr-Hildreth operator is based on the zero-crossing analysis of image convolution with the Laplacian of the Gaussian function (LoG). An alternative edge detection method uses the difference of two Gaussian filters with different diffusion coefficients (DoG, also introduced by Marr and Hildreth) (see review in [1]).

The Gaussian filter is the solution of the diffusion equation. Thus the computation of the difference of two Gaussian filters can be replaced by a numerical solution of the system of two diffusion equations with different diffusion coefficients. This replacement idea is a basis of many filtering algorithms like anisotropic diffusion.

The idea to use the FitzHugh-Nagumo reaction-diffusion system of equations for binary image edge detection was suggested in [2] by Nomura et al. Later they suggested a modification of this algorithm for greyscale images: local threshold value method [3]. Nevertheless, these methods produce false edges for noisy images and images with ringing artifact. In this paper, to overcome this problem we propose to use the FitzHugh-Nagumo reaction-diffusion system with image-dependent variable diffusion coefficient.

2. FITZHUGH-NAGUMO MODEL

The FitzHugh-Nagumo system can be represented as follows:

$$\begin{aligned}\frac{\partial u}{\partial t} &= D_u \Delta u + f(u, v), \Omega \times [0, T], \\ \frac{\partial v}{\partial t} &= D_v \Delta v + g(u, v), \Omega \times [0, T], \\ f(u, v) &= \frac{1}{\varepsilon} (u(1-u)(u-a) - v), \\ g(u, v) &= u - bv, \\ u(x, t, t=0) &= I(x, y), \Omega,\end{aligned}$$

$$\begin{aligned}v(x, y, t=0) &= 0, \Omega, \\ \frac{\partial u}{\partial n} \Big|_{\partial \Omega} &= \frac{\partial v}{\partial n} \Big|_{\partial \Omega} = 0,\end{aligned}$$

where Ω is rectangle $[0, x_0] \times [0, y_0]$, ε is a small positive constant, D_u and D_v are positive constants $D_u < D_v$.

If we want to understand the time behaviour of u and v function, we can consider a simplified system:

$$\begin{aligned}\frac{\partial u}{\partial t} &= f(u, v), \Omega \times [0, T], \\ \frac{\partial v}{\partial t} &= g(u, v), \Omega \times [0, T], \\ f(u, v) &= \frac{1}{\varepsilon} (u(1-u)(u-a) - v), \\ g(u, v) &= u - bv.\end{aligned}$$

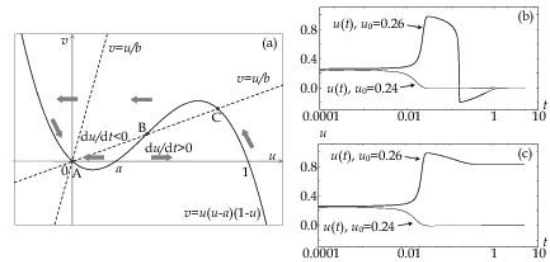


Figure 1. FitzHugh-Nagumo model behavior. (a) Phase plot. (b) monostable system ($a=0.25, b=1$). (c) bistable system ($a=0.25, b=10$). $\varepsilon = 10^{-3}$ [2]

Picture 1(a) shows that function u increases in the area below the $v = u(u-1)(1-u)$ curve, and function v increases in the area below $u = bv$ line. Thus, if our starting point is $(u, 0)$, $u \in (a, 1)$ and the curves intersect, the system will achieve an equilibrium in the C point. If we start from $(u, 0)$, $u \in (0, a)$ point, the system will achieve an equilibrium in the A point. If lines do not intersect, the system will achieve an equilibrium in the A point in both cases. Thus, the behaviour of the system depends on ratio of the a and b parameters. It can be monostable or bistable.

Full equation system causes appearance of the pulses in the edge points. FitzHugh-Nagumo system can detect edges of the binary image, but it is impossible for it to detect edges of greyscale image, because the FitzHugh-Nagumo system defines edge using global threshold value. Local threshold value was introduced for the purpose of greyscale images edge detection [3].

Problem with local threshold value

$$a = a(x, y, t),$$

can be represented as follows:

$$\begin{aligned}\frac{\partial u}{\partial t} &= D_u \Delta u + f(u, v), \Omega \times [0, T], \\ \frac{\partial v}{\partial t} &= D_v \Delta v + g(u, v), \Omega \times [0, T], \\ \frac{\partial a}{\partial t} &= D_a \Delta a, \Omega \times [0, T], \\ f(u, v) &= \frac{1}{\varepsilon} (u(1-u)(u-a) - v), \\ g(u, v) &= u - bv, \\ u(x, t, t=0) &= a(x, t, t=0) = a_0 * I(x, y), \Omega, \\ v(x, y, t=0) &= 0, \Omega, \\ \frac{\partial u}{\partial n} |_{\partial \Omega} &= \frac{\partial v}{\partial n} |_{\partial \Omega} = \frac{\partial a}{\partial n} |_{\partial \Omega} = 0.\end{aligned}$$

Diffusion coefficient D_a should satisfy $D_a > D_v$ condition and $0 < a_0 < 1$.

3. FITZHUGH-NAGUMO MODEL WITH VARIABLE DIFFUSION COEFFICIENT

FitzHugh-Nagumo system with local threshold value allows us to make an effective edge emphasis for "clear" images. But false edges appear in the case of noise or ringing artefact (Gibbs effect). To solve this problem we introduce the variable diffusion coefficient, as in Perona-Malik approach[4]:

$$h(x, y) = \frac{1}{1 + \lambda^2 |\nabla I(x, y)|^2}.$$

The suggested diffusion coefficient is small in the edge areas, and diffusion in smooth areas is accelerated. These areas will be de-noised more intensively. Thus we do not have false edges that can appear in the case of constant diffusion coefficient.

Problem with variable diffusion coefficient can be represented as follows:

$$\begin{aligned}\frac{\partial u}{\partial t} &= \text{div}(h(x, y) \nabla u) + f(u, v), \Omega \times [0, T], \\ \frac{\partial v}{\partial t} &= D_v \Delta v + g(u, v), \Omega \times [0, T], \\ \frac{\partial a}{\partial t} &= D_a \Delta a, \Omega \times [0, T], \\ f(u, v) &= \frac{1}{\varepsilon} (u(1-u)(u-a) - v), \\ g(u, v) &= u - bv, \\ u(x, t, t=0) &= a(x, t, t=0) = a_0 * I(x, y), \Omega, \\ v(x, y, t=0) &= 0, \Omega, \\ \frac{\partial u}{\partial n} |_{\partial \Omega} &= \frac{\partial v}{\partial n} |_{\partial \Omega} = \frac{\partial a}{\partial n} |_{\partial \Omega} = 0.\end{aligned}$$

4. INFLUENCE OF VARIABLE DIFFUSION COEFFICIENT ON THE QUALITY OF EDGE DETECTION

The results of the application of edge detection algorithm with variable diffusion coefficient are represented in figures 2-9. The algorithm has been applied to noisy images and images with ringing. For comparison the results of constant diffusion coefficient algorithm with $D_u = 1.5$ are given.

The following difference scheme was chosen for the computation:

$$\begin{aligned}\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\tau} &= h(i, j) \left(\frac{u_{i-1,j}^n - 2u_{i,j}^n + u_{i+1,j}^n}{(\Delta x)^2} + \right. \\ &+ \frac{u_{i,j-1}^n - 2u_{i,j}^n + u_{i,j+1}^n}{(\Delta y)^2} \left. \right) + \frac{(u_{i,j}^n - u_{i-1,j}^n)(h(i,j) - h(i-1,j))}{(\Delta x)^2} + \\ &+ \frac{(u_{i,j}^n - u_{i,j-1}^n)(h(i,j) - h(i,j-1))}{(\Delta y)^2} + f_{i,j}^n, \\ \frac{a_{i,j}^{n+1} - a_{i,j}^n}{\tau} &= \\ &= D_a \left(\frac{a_{i-1,j}^n - 2a_{i,j}^n + a_{i+1,j}^n}{(\Delta x)^2} + \frac{a_{i,j-1}^n - 2a_{i,j}^n + a_{i,j+1}^n}{(\Delta y)^2} \right), \\ \frac{v_{i,j}^{n+1} - v_{i,j}^n}{\tau} &= \\ &= D_v \left(\frac{v_{i-1,j}^n - 2v_{i,j}^n + v_{i+1,j}^n}{(\Delta x)^2} + \frac{v_{i,j-1}^n - 2v_{i,j}^n + v_{i,j+1}^n}{(\Delta y)^2} \right) + g_{i,j}^n, \\ f_{i,j}^n &= \frac{1}{\varepsilon} (u_{i,j}^n (1 - u_{i,j}^n) (u_{i,j}^n - a_{i,j}^n) - v_{i,j}^n), \\ g_{i,j}^n &= u_{i,j}^n - bv_{i,j}^n, \\ h(i, j) &= \frac{1}{1 + \lambda^2 \left(\left(\frac{u_{i,j}^n - u_{i-1,j}^n}{\Delta x} \right)^2 + \left(\frac{u_{i,j}^n - u_{i,j-1}^n}{\Delta y} \right)^2 \right)}.\end{aligned}$$

The parameters are: $a_0 = 0.25$; $b = 5$; $D_v = 5$; $D_a = 10$; $\varepsilon = 0.001$; $\lambda = 1$.

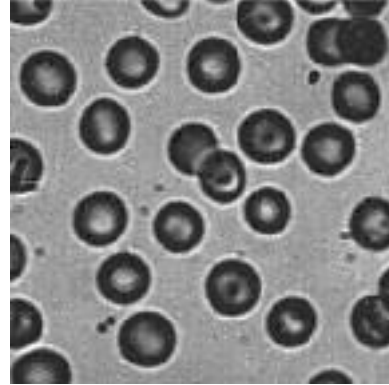


Figure 2. Source noisy image

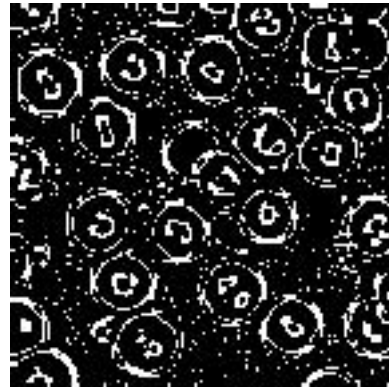


Figure 3. Result of constant diffusion coefficient FitzHugh-Nagumo edge detector

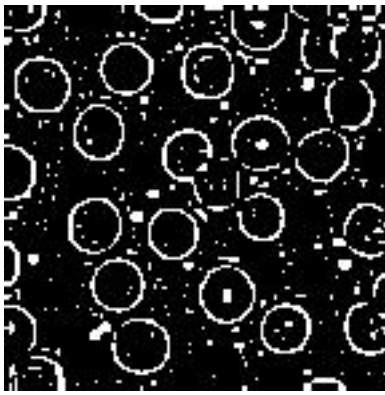


Figure 4. Result of FitzHugh-Nagumo edge detector with variable diffusion coefficient

As we see in the figures 2 - 4, for noisy images FitzHugh-Nagumo model with variable diffusion coefficient detects edges more effectively. It is also effective for images with ringing and noise. The results with the same parameters of computation for images with ringing are shown in the figures 5 - 7, for images with noise and ringing - in figures 8 - 10.

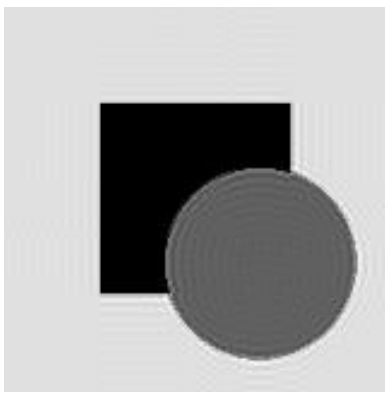


Figure 5. Source image with ringing

5. CONCLUSION

It was shown that the introduction of the image dependent variable coefficient makes edge detection more effective and allows to avoid false edges, caused by noise and ringing. The work was supported by federal target program Scientific and scientific-pedagogical personnel of innovative Russia in 2009-2013 and RFBR grant 08-01-00314.

6. REFERENCES

- [1] A. Pinho, L. Almeida, "A review on edge detection based on filtering and differentiation", *Revista do DETUA*, Vol. 2, No 1, pp. 113-126, 1997.
- [2] A. Nomura, M. Ichikawa, R.H. Sianipar, H. Miike, "Reaction-Diffusion Algorithm for Vision Systems", *Vision Systems:*

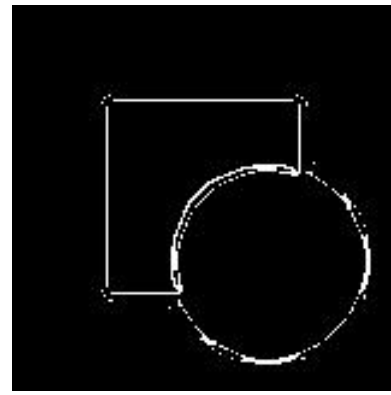


Figure 6. Result of constant diffusion coefficient FitzHugh-Nagumo edge detector

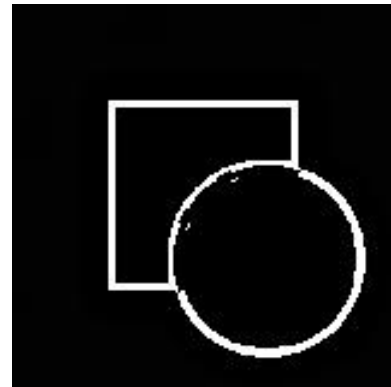


Figure 7. Result of FitzHugh-Nagumo edge detector with variable diffusion coefficient

Segmentation and Pattern Recognition, ed. G. Obinata and A. Dutta, *i-Tech*, Vienna, p. 61-80, 2007.

- [3] A. Nomura, M. Ichikawa, R.H. Sianipar, H. Miike, "Edge Detection with Reaction-Diffusion Equations Having a Local Average Threshold", *Pattern Recognition and Image Analysis*, Vol. 18, No 2, pp. 289-299, 2008.
- [4] P. Perona, J. Malik, "Scale-space and Edge Detection Using Anisotropic Diffusion", *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 629-639, 1990.
- [5] J. Weickert, "A review of nonlinear diffusion filtering", *Scale-Space Theory in Computer Vision*, Springer Berlin / Heidelberg, pp. 1-28, 1997.

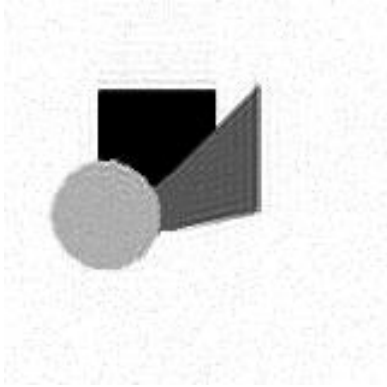


Figure 8. Source image with noise and ringing

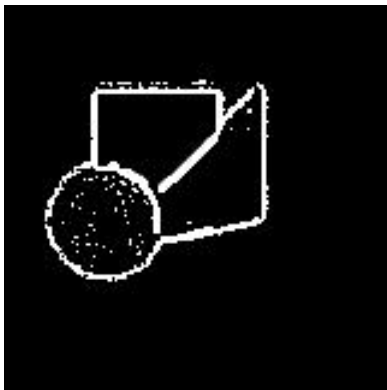


Figure 9. Result of FitzHugh-Nagumo edge detector with $D_u = 1.5$

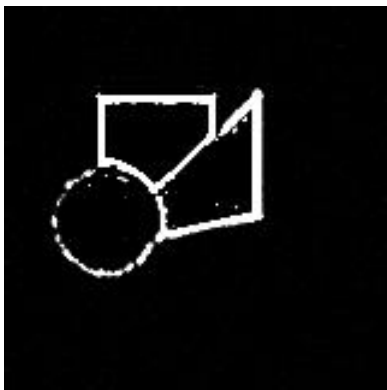


Figure 10. Result of FitzHugh-Nagumo edge detector with variable diffusion coefficient $D_u(x, y) = \frac{1}{1 + \lambda^2 |\nabla I(x, y)|^2}$

Scale Space Edge Curvature Estimation and Its Application to Straight Lines Detection

Ekaterina V. Semeikina, Dmitry V. Yurin

Laboratory of Mathematical Methods of Image Processing
Faculty of Computational Mathematics and Cybernetic
Moscow Lomonosov State University, Moscow, Russia
esemeikina@graphics.cs.msu.ru, yurin@cs.msu.su

Abstract

The differential geometry based formula for edge curvature estimation at a fixed scale is derived in the paper. It is shown that having lines of various curvatures in the same image, one is not able to estimate them correctly using a fixed scale. So we propose scale space algorithm which allows to choose proper scale adaptively in every point of interest. Relationship between curvature and scale required to estimation was derived analytically, constant was found and the relationship was proved via numerical modeling. The scale space algorithm for edge curvature estimation can be used in particularly for straight line detection. It allows to select only points of interest (points with low curvature). This speeds up detection algorithms and decreases probability of missing lines for complex scenes.

Keywords: *curvature estimation, edge detection, scale space, differential geometry, straight line detection, circle detection.*

1. INTRODUCTION

The idea of scale-space was suggested by Witkin [1] and Koenderink [2] and then was developed by Lindeberg [3]. Scale space approach is used in the theory of differential invariants [8] which have been successfully applied to feature detection, describing of detected features [4,5,6] and edge detection [7]. Informativeness of such differential invariants as gradient absolute value and Laplacian has been proved by effectiveness of their application to edge and feature detection tasks. However, informativeness of high order invariants is unclear from the tasks of matching or learning [12]. In this paper it is first time suggested to use 4-th order invariant – edge curvature – for measurements instead of characterization. It is shown in the paper that edge curvature calculated in scale space is stable enough to classify edge points.

Typical straight line detection scheme consists of two steps: edge detection and straight segment finding. The last step are generally realized via Hough transform (HT) [10,11] or some randomized algorithm. If we remove after the first step all points where edges have high curvature a lot of curve lines will be removed. Thus maxima in accumulator array will be sharper for HT and probability of successful choice will be larger for randomized algorithm.

2. CURVATURE ESTIMATION AT A FIXED SCALE

We will deal with images smoothed via convolution with Gaussian kernel, i.e. we use scale-space $L(x, y, t)$ parameterized with $t = \sigma^2$, where σ is the Gaussian's standard deviation.

Let us consider edge point P on the plane (x, y) . We will use so called *gauge coordinates* [8] (v, w) in the point P, where v-axis is tangent to the curve in the point P and w-axis is normal to the curve. Origin of coordinates is in the point P, so $v=0$, $w=0$. Curvature in the point (0,0) is $\kappa = w'' = d^2w(v)/dv^2$.

Let us consider a curve equation $f(x, y) = const$ and obtain the curvature formula for it in a manner similar to [8]. After taking first and second derivatives with respect to v we obtain:

$$f_v + f_w w' = 0 \quad (1)$$

$$\text{and } f_{vv} + f_{vw} w' + f_{ww} w'^2 + f_w w'' = 0 \quad (2)$$

As v -axis is tangent to curve in point P, $f_v(0,0) = 0$ and (1) results in $w' = 0$. Thus (2) can be reduced to:

$$f_{vv}(0,0) + f_w(0,0) w''(0) = 0, \quad (3)$$

$$\kappa = w''(0) = -f_{vv}(0,0)/f_w(0,0) \quad (4)$$

Expression (4) is the equation for curvature of the line defined by the implicit equation $f(x, y) = const$.

In the gauge coordinates edge can be described as points where absolute value of intensity derivative in the w-axis direction $\partial L / \partial w$ have a maxima, i.e. $L_{ww} = 0$.

For edge defined by $f(x, y) = L_{ww}$, the expression $f_v(0,0) = 0$ is satisfied and hence (4) can be used to obtain curvature equation:

$$\kappa_{edge} = w''(0) = -L_{wvv}(0,0)/L_{wvw}(0,0) \quad (5)$$

Equation (5) in coordinates (x, y) can be rewritten explicitly:

$$\begin{aligned} \kappa_{edge} &= -A/B, \quad \text{where} \\ A &= (L_{xxx} + L_{yyy})L_x^2 L_y^2 + L_{xxy} (L_x^4 + L_y^4 - 4L_x^2 L_y^2) + \\ &\quad + 2L_x L_y (L_x^2 - L_y^2) (L_{xxy} - L_{xyy}) \\ B &= \sqrt{L_x^2 + L_y^2} (L_{xxx} L_x^3 + L_{yyy} L_y^3 + 3L_x L_y (L_{xxy} L_x + L_{xyy} L_y)) \end{aligned} \quad (6)$$

Figure 1 demonstrates curvature estimation with various values of σ for image 1a of size 512x512 containing the circles with radii 10, 30, 50, 100, 150, 200 pixels. It is obvious from (6) that curvature estimation can be carried out only near edges where gradient

absolute value is not near to zero. Other regions are marked with white in Figure 1c-e.

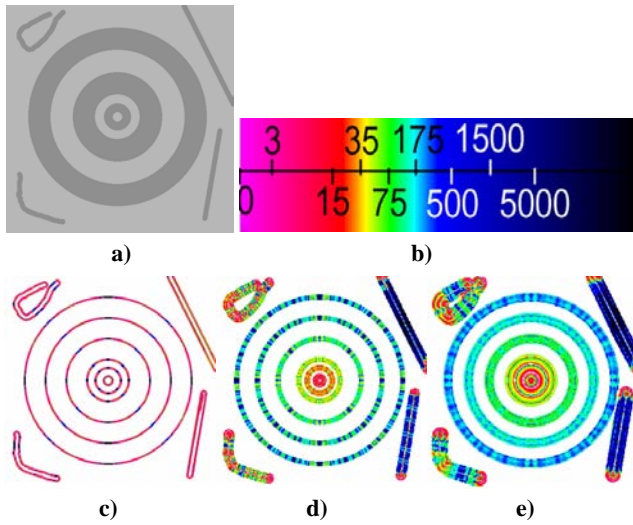


Figure 1: a) the original image, b) correspondence between color and curvature radius $R=1/\kappa$ in pixels, c-e) curvature estimation using edge curvature formula (6): c) $\sigma=1$, d) $\sigma=3.17$, e) $\sigma=6.35$.

Nonconstant curvature along circle is caused by pixel structure of the image. It is seen that this radial effect decreases with increasing scale σ . Derivatives in (6) are computed via convolutions with derivatives of Gaussian function, i.e. via integration in neighbourhood with radius of order of σ . So, an acceptable accuracy of curvature estimation can be reached if the filled area in Figure 2 contains large enough number of pixels:

$$\Delta \cdot \sigma > const .$$

It can be easily seen from Figure 2 that

$$\begin{aligned} \Delta &= R - R \cdot \cos(\arcsin(\sigma/R)) = \\ &= R \cdot \left(1 - \sqrt{1 - \sin^2(\arcsin(\sigma/R))}\right) = \\ &= R \cdot \left(1 - \sqrt{1 - (\sigma/R)^2}\right) \approx 1/2 \cdot R(\sigma/R)^2 = \sigma^2/(2R) \end{aligned}$$

So, substituting this result into previous formula we obtain

$$\sigma^3/R > c, \text{ where } c \text{ is constant} \quad (7)$$

When such $\sigma = \sigma_{req}$ is reached we can see the uniformly colored circles in Figure 1d (red circle $R=10$, $\sigma=3.17$) and 1e (green circle $R=100$, $\sigma=6.35$). While $\sigma < \sigma_{req}$, curvature estimation demonstrated radial structure caused by pixel structure of a curve.

In Figure 1e the typical ring-like artifacts of various colors appeared near circles of small radii. These artifacts are caused by influence of neighbour edges which are closer to each other than distance proportional to σ . In such cases integrals value are result of some ‘‘edge mixture’’ and curvature estimation is bad.

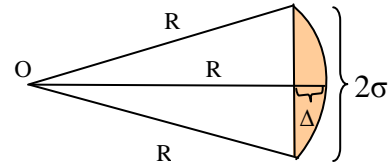


Figure 2: Circle arc.

Any edge in real image is likely to be situated near other edges. Since distance to the closest edge is unknown we use the smallest size of sigma which satisfies (7) to decrease possibility of neighbour edges influence:

$$\sigma_{req}(R) = \sqrt[3]{cR} \quad (8)$$

So, having edges with different curvatures in the same image it is impossible estimate their curvature using the same σ . It is the reason to develop a scale-space algorithm proposed in section 5.

3. PRECISION OF CURVATURE ESTIMATION AND DETERMINATION OF THE PROPER SCALE

Estimation (8) is confirmed by the experiments with model images each of which contains a single circle of the known radius $R_0 \in [10, 900]$. For these images we calculate curvature in edge points using the same σ for the whole image. Then we build histogram of the obtained curvature values.

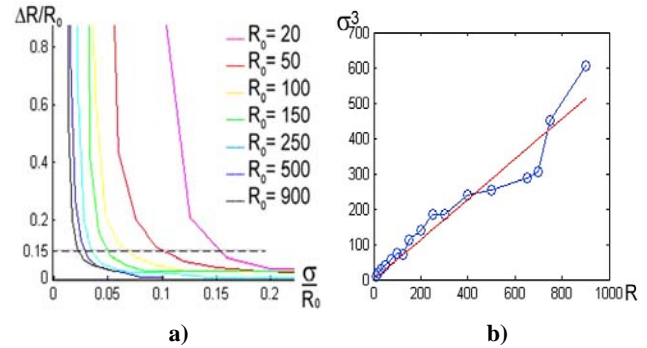


Figure 3: a) the dependence of curvature estimation error $\Delta R/R_0$ and σ/R_0 . ΔR is histogram dispersion, R_0 is

known radius, b) – minimal value of σ^3 allowing estimation with precision 15%. – approximation of the experimental line.

Curvatures in the points of ideal circle are the same, but we deal with circle at a finite pixel grid and there is radial effect. For curvature estimation with $\sigma < \sigma_{req}(R)$ there is spread of obtained curvature values and its histogram has width $\Delta R = \left\langle (R - \langle R \rangle)^2 \right\rangle^{1/2}$ other than 1. So we need to analyze histogram width (see Figure 3a). Fixing the desired precision ($\Delta R/R_0 = 15\%$ was used) we can determine $\sigma_{req}(R)$ for which chosen precision reaches. Inclination angle of approximation line in Figure 3b gives the coefficient for (7), (8) $c = 1.4$.

4. SCALE SPACE CURVATURE ESTIMATION

In this section we propose an algorithm for scale space curvature estimation. In our tests scale-space was built using σ grid: $\sigma_0 = 1.0$, $\sigma_{i+1} = s \cdot \sigma_i$, $s = \sqrt[3]{2}$, and the constants $\alpha = \beta = 0.2$ were used.

Algorithm 1.

for each scale level σ_i :

for each pixel where curvature has not been calculated yet:

1. Get curvature value κ_i using σ_i from (6).
2. Calculate average $\langle \kappa_i \rangle$ and d mean deviation of κ_i in the edge-neighbourhood of the current point
3. Calculate $R_i := 1/\langle \kappa_i \rangle$ and $\sigma_{req}(R_i)$ using (8)
4. **If** ($\sigma_{req}(R_i) \leq \sigma_i$ or $d \leq \alpha|\kappa_i|$ or $|\kappa_i - \kappa_{i-1}| \leq \beta|\kappa_i|$) **then** $\langle \kappa_i \rangle$ is correct, do not calculate value in this point with further scales.

Edge-neighbourhood means only edge points which are situated within neighbourhood of the current point. Following the logic of scale space approach we choose neighbourhood radius dependent on scale and equal to $\max(2, \sigma_i/2)$.

On step 5 of the algorithm we check stability and correctness of the curvature κ_i calculated using σ_i . If curvature deviation within edge-neighbourhood is high: $d > \alpha|\kappa_i|$ then it is caused by noise and radial effect and greater scale should be chosen to correct estimation. Also if curvature value changes sharply from previous scale to the current: $|\kappa_i - \kappa_{i-1}| > \beta|\kappa_i|$, then current scale σ_i is not large enough.

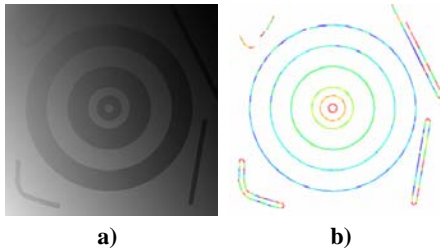


Figure 4: b) Scale space curvature estimation for image 4a. Curvature is shown only in edge points (see 1b for color notation).

It can be seen from Figure 4 that the proposed multiscale algorithm allows correct curvature estimation for images containing lines with various curvatures. Application of statistical differencing [9, p. 288] to images before curvature estimation allows to enhance image quality and to obtain more precise estimation.

5. CURVATURE FILTERING FOR LINE DETECTION

Straight line detection is an example of the area where the proposed curvature algorithm can be applied. A common scene usu-

ally contains not only straight lines and selection of points with low curvature allows straight line detector to work only with points of interest and to exclude from consideration impossible line positions (some kind of noise). We demonstrate this benefit on example of Hough transform [10].

For each image from the test base (containing synthetic and real images) we have compared HT sinogram [11] (accumulator array) mapped from all edge points and sinogram mapped from points with low curvature. Typical examples of such sinograms are presented in Figures 5d, f.

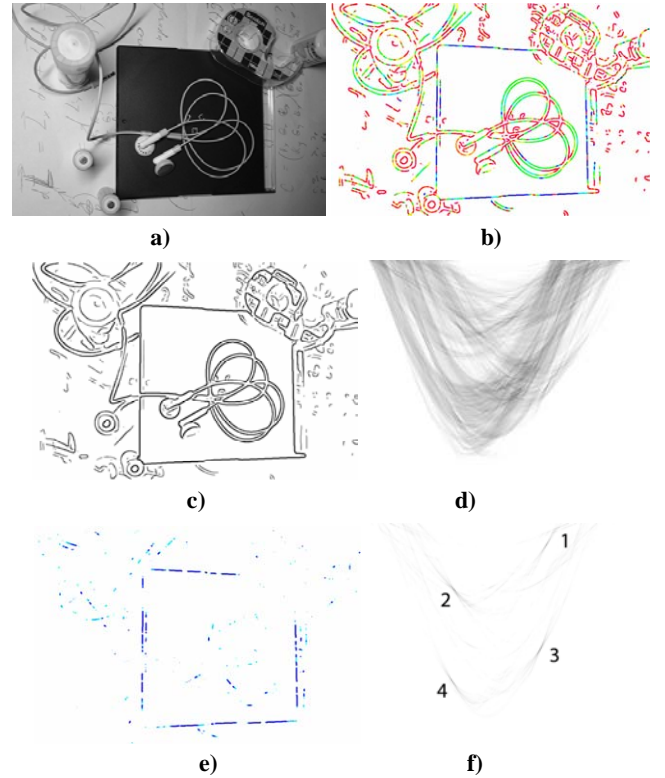


Figure 5: b) curvature estimation for image 6a, c) all edges, d) sinogram built for all edge points, e) edge points with high curvature radius $R > 150$, f) sinogram built for 6e).

In order to compare this sinograms we introduce measure of contrast of sinogram local maxima $T = I_V/I$, where I is intensity in local maxima, I_V is average intensity in a ring neighbourhood of maxima. This comparison has shown that contrast of sinogram maxima corresponding to real straight lines increased significantly (see Table 1), while sinogram maxima due to false line detections were suppressed.

Table 1: Measure of contrast in sinogram local maximum $T = I_V/I$

	Sinogram built for:		
	gradient absolute value	all edges	filtered edges
Point 1	0.5	0.32	0.034
Point 2	0.4	0.13	0.03
Point 3	0.37	0.17	0.029
Point 4	0.52	0.18	0.022
Average over test base	0.4	0.15	0.03

Some examples of high curvature edges suppression are shown in the Figure 6.

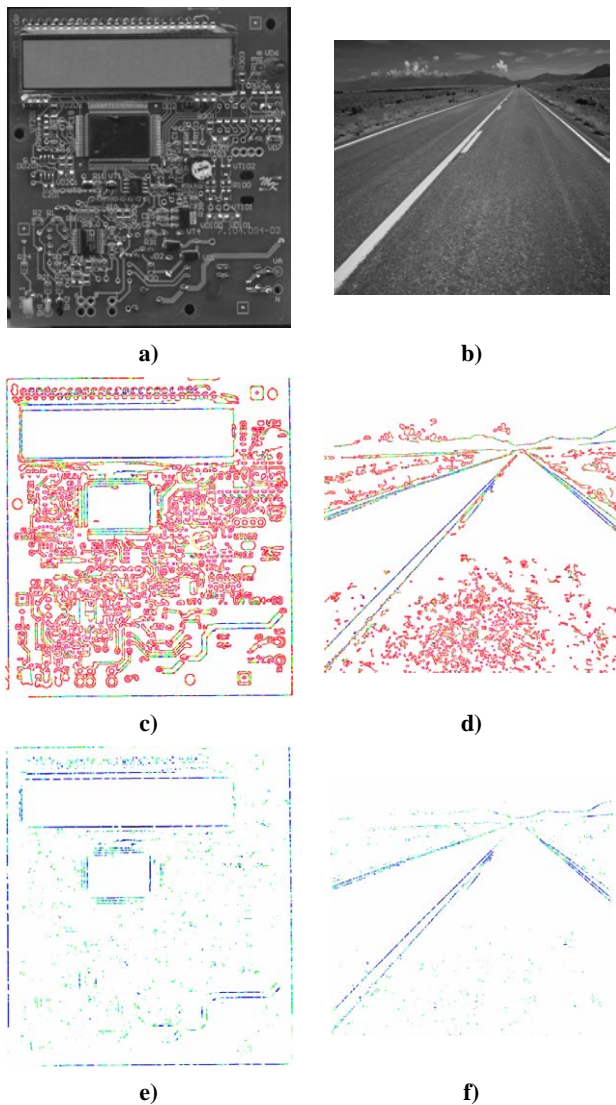


Figure 6: c), d) edges curvature estimation, e), f) edges with low curvature. Colors correspond to curvatures according to Figure 1b

6. CONCLUSION

The differential geometry based formula (6) for edge curvature at a fixed scale has been derived in the paper. It has been shown that curvature estimation at a fixed scale suffers of lack of accuracy so a scale space algorithm for edge curvature estimation has been proposed.

We have analyzed curvature estimation accuracy in dependence on used scale and have obtained a requirement for minimal scale selection; the coefficient in this formula has been obtained with numerical modeling.

The tests performed on synthetic and natural images show that the proposed algorithm allows correct curvature estimation for most of edge points even for images where lines curvature changes in range from small circles to straight lines.

An application of curvature analysis for straight line detection has been suggested. It has been shown that suppression of edge points with high curvature results in increasing sinogram quality and better contrast of its pikes correspondent to straight lines.

7. ACKNOWLEDGMENTS

The research is done under support of RFBR, grants 09-07-92000-HHC_a, 09-01-92470-MHKC_a and federal target program “Scientific and scientific-pedagogical personnel of innovate Russia in 2009-2013”.

8. REFERENCES

- [1] A.P.Witkin. *Scale-Space Filtering*. 8th International Joint Conference on Artificial Intelligence, 1019 - 1022, August 1983.
- [2] J. J. Koenderink, A. J. van Doorn. *The Structure of Images*. Biological Cybernetics, 50: 363 - 370, 1984.
- [3] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, Dordrecht, 1994.
- [4] H. Bay, T. Tuytelaars, L. V. Gool. *SURF: Speeded Up Robust Features*. 9th ECCV, 3951(1): 404 - 417, 2006.
- [5] D G. Lowe. *Object Recognition from Local Scale-Invariant Features*. ICCV, 1150 - 1157, September 1999.
- [6] E. Tola, V. Lepetit, P. Fua. *A Fast Local Descriptor for Dense Matching*. IEEE CVPR, 1 - 8, 2008.
- [7] T. Lindeberg. *Edge Detection and Ridge Detection with Automatic Scale Selection*. CVPR, 30(2): 117 - 156, 1998.
- [8] L. Florack, B. M. H. Romeny, J. J. Koenderink, M. A. Viergever. *Scale and the Differential Structure of Images*. Image and Vision Computing, 10: 376 - 388, 1992.
- [9] W. K. Pratt. *Digital Image Processing: PIKS Scientific inside. - 4th ed.* Wiley-Interscience, 2007.
- [10] R. Duda, P. Hart. *Use of the Hough Transformation to Detect Lines and Curves in Pictures*. Comm. ACM 15: 1 - 15, 1972.
- [11] C.G.Ho, R.C.D.Young, C.D.Bradfield, C.R.Chatwin. *A Fast Hough Transform for the Parameterization of Straight Lines using Fourier Methods*. Real-Time Imaging, 6(2): 113-127, 2000
- [12] C. Schmid, R. Mohr. *Local grayvalue invariants for image retrieval*. PAMI, 19(5):530-534, 1997.

About the authors



Ekaterina V. Semeikina is PhD student at Laboratory of Mathematical Methods of Image Processing, Faculty of Computational Mathematics and Cybernetics, Moscow Lomonosov State University <http://imaging.cs.msu.su>. Her contact email is esemeikina@graphics.cs.msu.ru



Dmitry V. Yurin, PhD, is a senior scientist at Laboratory of Mathematical Methods of Image Processing, Faculty of Computational Mathematics and Cybernetics, Moscow Lomonosov State University <http://imaging.cs.msu.su>. His contact email is yurin@cs.msu.su.

Short Reference Image Quality Estimation Using Modified Angular Edge Coherence

Dmitry V. Sorokin, Andrey S. Krylov
 Faculty of Computational Mathematics and Cybernetics
 Moscow Lomonosov State University, Moscow, Russia
 {dsorokin, kryl}@cs.msu.ru

Abstract

A new short reference image quality metric is introduced in this paper. It is based on the idea that a high quality image has a high value of angular edge coherence in the basic edges points and a low level of angular edge coherence in the basic edges neighborhood. The angular edge coherence is measured using the coefficients of image expansion into the Gauss-Laguerre Circular Harmonic Functions. The practical usability is illustrated with several test images.

Keywords: Gauss-Laguerre circular harmonic functions, image quality metric, angular edge coherence, basic edges.

1. INTRODUCTION

The problem of results quality estimation in image enhancement or restoration algorithms does not have an universal solution. The automatic image quality metrics often do not correlate well enough with the perceptual image quality.

There are many approaches which are performed by direct pixel-by-pixel calculation of conventional signal metrics (such as MSE, PSNR, RMSE etc.) of the obtained and the original images. These techniques are referred in the literature as full reference image quality assessment metrics. There are also more sophisticated metrics taking into account perceptual features of human visual system [1].

But for the majority of image processing tasks we cannot use the full reference metrics due to the absence of the reference image. For this reason no-reference (or “blind”) methods and short reference methods are designed. No-reference approaches assess the quality of image without any information about original image. Short reference approaches operate with partial information about original image. Some of these methods are introduced in [2-4]. The methods based on the edge coherence approach are introduced in [5,6]. They are based on the local expansion of the observed image into the system of so-called Gauss-Laguerre orthonormal family.

In this paper a new approach using modified angular edge coherence is proposed. The modification is based on Gauss-Laguerre expansions analysis. The importance for the human image perception of the areas near the main image edges has also been taken into consideration.

2. MULTIMODAL ANGULAR EDGE COHERENCE

Let us consider a family of complex orthonormal and polar separable family of functions:

$$g_n^\alpha(r, \gamma; \sigma) = \psi_n^{|\alpha|}(r^2 / \sigma) e^{i\alpha\gamma}.$$

Their radial profiles are Laguerre functions:

$$\psi_n^\alpha(x) = \frac{1}{\sqrt{n! \Gamma(n + \alpha + 1)}} x^{\alpha/2} e^{-x/2} L_n^\alpha(x),$$

where $n = 0, 1, \dots; \alpha = 0, \pm 1, \pm 2, \dots$ and $L_n^\alpha(x)$ are Laguerre polynomials:

$$L_n^\alpha(x) = (-1)^n x^{-\alpha} e^x \frac{d}{dx^n} (x^{n+\alpha} e^{-x}).$$

The Laguerre function $\psi_n^\alpha(x)$ can be calculated using the following recurrence relations:

$$\psi_{n+1}^\alpha(x) = \frac{(x - \alpha - 2n - 1)}{\sqrt{(n+1)(n+\alpha+1)}} \psi_n^\alpha(x) - \sqrt{\frac{n(n+\alpha)}{(n+1)(n+\alpha+1)}} \psi_{n-1}^\alpha(x), \quad n = 0, 1, \dots,$$

$$\psi_0^\alpha(x) = \frac{1}{\sqrt{\Gamma(\alpha+1)}} x^{\alpha/2} e^{-x/2}, \quad \psi_{-1}^\alpha(x) \equiv 0$$

These functions g_n^α , called Laguerre Gauss circular harmonic (LG-CH) functions, are referenced by integers n (referred by radial order) and α (referred by angular order). The real and imaginary part of g_0^1 , g_0^3 and g_0^5 are illustrated in Figure 1.



Figure 1: The real parts of g_0^1 , g_0^3 and g_0^5 (upper row) and the imaginary parts of g_0^1 , g_0^3 and g_0^5 (lower row).

The LG-CH functions are self-steerable, i.e. they can be rotated by the angle φ using multiplication by the factor $e^{i\alpha\varphi}$. They also keep their shape invariant under Fourier transformation. They are suitable for multiscale and multicomponent image analysis [7].

Let us consider an observed image $I(x, y)$ defined on the real plane R^2 . Due to the orthogonality of g_n^α family the image $I(x, y)$ can be expanded in the analysis point x_0, y_0 for fixed σ in Cartesian system as follows:

$$I(x_0, y_0) = \sum_{\alpha=-\infty}^{\infty} \sum_{n=0}^{\infty} c_{\alpha,n} g_n^\alpha(x, y),$$

where

$$c_{\alpha,n} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(x_0, y_0) (g_n^\alpha(x, y))^* dx dy,$$

and

$$x = x_0 + r \cos \gamma, \quad y = y_0 + r \sin \gamma.$$

These coefficients can be calculated at any image point x_0, y_0 by convolution:

$$c_{\alpha,n}(x_0, y_0) = I(x, y) * (g_n^\alpha(-x, -y))^*.$$

Let us consider the convolution results $c_{1,0}(x, y)$, $c_{3,0}(x, y)$ and $c_{5,0}(x, y)$ for LG-CH functions of index orders 1,0 3,0 and 5,0. As it has been shown in [5, 6] the high level of magnitude of $c_{1,0}(x, y)$ corresponds to the edges of the image, while $c_{3,0}(x, y)$ and $c_{5,0}(x, y)$ corresponds to different types of crosses, corners (these values are smaller in the edge points). At the same time it is shown in [6] that for the ideal edge patterns, i.e. unitary step edges passing through x_0, y_0 , the arguments of $c_{1,0}(x, y)$ and $c_{3,0}(x, y)$ satisfy the following relation:

$$\arg(c_{3,0}(x_0, y_0)) = 3 \arg(c_{1,0}(x_0, y_0)) + \pi.$$

It can be shown that the arguments of $c_{1,0}(x, y)$ and $c_{5,0}(x, y)$ satisfy the analogous relation:

$$\arg(c_{5,0}(x_0, y_0)) = 5 \arg(c_{1,0}(x_0, y_0)).$$

This relationship is analogous to the relationship of the odd harmonics of the Fourier expansion for a periodic square waveform (see Figure 2). In our approach we take 3 first odd modes (the work [6] took 2 first odd modes).

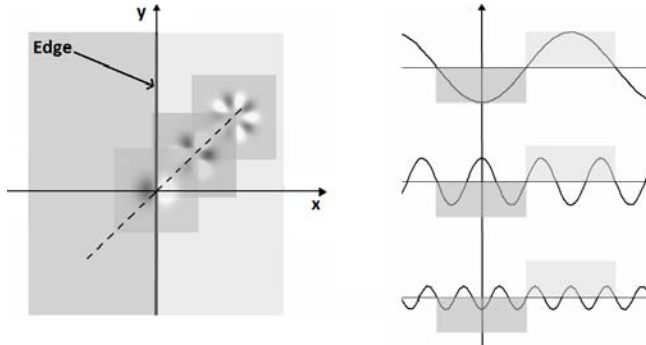


Figure 2: Phase relationship and its analogue with the odd harmonics of the Fourier expansion for a periodic square waveform.

The modified angular edge coherence filter based on these relationships is defined as:

$$MAEC(x_0, y_0) = |c_{1,0}(x_0, y_0)| \cdot |\cos(8 \arg(c_{1,0}(x_0, y_0)) - \arg(c_{3,0}(x_0, y_0)) - \arg(c_{5,0}(x_0, y_0))) \cdot (|c_{3,0}(x_0, y_0)| \cos(3 \arg(c_{1,0}(x_0, y_0)) - \arg(c_{3,0}(x_0, y_0))) + |c_{5,0}(x_0, y_0)| \cos(5 \arg(c_{1,0}(x_0, y_0)) - \arg(c_{5,0}(x_0, y_0))))|$$

This approach is more sensitive to the weak edges than the AEC [5] (PEC [6]). The AEC metric (with contrast normalization) is defined in [6] as:

$$AEC(x_0, y_0) = -|c_{1,0}(x_0, y_0)| \cdot |c_{3,0}(x_0, y_0)| \cdot \cos(3 \arg(c_{1,0}(x_0, y_0)) - \arg(c_{3,0}(x_0, y_0)))$$

It is also important to mention that the value of MAEC is positive while the values of AEC (PEC) can be negative as well. This is crucial for the calculation of MAEC for an image region R:

$$MAEC_R(I) = \frac{\sum_{(x,y) \in R} MAEC(x, y)}{\|R\|},$$

where $\|R\|$ is the number of points in R and the value MAEC(x,y) is normalized to lie within [0,1] interval.

3. IMAGE QUALITY METRIC

Standard metrics based on whole image square error calculation like MSE, PSNR do not correlate well with the perceptual image quality. As an example, ringing effect in textured areas is not noticeable while ringing effect near sharp isolated edges is annoying.

In [8] the authors introduced an image quality metrics in edge and edge neighborhood regions for image restoration methods like image interpolation or image deringing. The authors introduced Basic Edge Points (BEP) regions and Basic Edge Neighborhood (BEN) regions in images (see Figure 3). They estimated the quality using RMSE (root of mean square error) of the reference image and the restored image within these areas.



a) Reference image b) Its BEP (white color) and BEN (grey color) regions

Figure 3: BEP and BEN regions illustration.

In this paper a short reference approach is introduced. This approach is inspired by the fact that MAEC value should be reasonably high in BEP regions, indicating the good quality of edge restoration. Meanwhile, the values of MAEC should be reasonably low in BEN regions, indicating the absence of ringing effect. Thus we suggest the following basic edge quality metric:

$$BEQ(I) = \frac{MAEC_{BEP}(I)}{MAEC_{BEN}(I)}$$

Obviously, the specific BEQ values correspond to the specific image. For this reason, the relative modification of BEQ can be defined as follows:

$$RBEQ(I) = \frac{BEQ(I)}{BEQ(\tilde{I})}$$

where $I(x, y)$ is observed image and $\tilde{I}(x, y)$ is the reference image. The $RBEQ$ metric is a short reference metric. It analyzes only areas in the region of isolated strong edges. The value of $RBEQ(I)$ is less than 1 in case of image quality degradation and greater than 1 in case of image quality improvement.

4. EXPERIMENTAL RESULTS

To illustrate the sufficiency of the proposed approach it was tested with different images. The suggested metric has been compared with $RTAEC$ metric (the relative short reference metric based on angular edge coherence function for the whole image) [5]. $RTAEC$ metric shows good correlation with DMOS results [5]. These results are also close to the results of MSSIM metric.



Figure 4: Comparison of $RBEQ$ and $RTAEC$ metrics.

In Figure 4 (“Lena”) and Figure 6 (“Peppers”) the reference images, their BEP and BEN map images and corrupted versions of the reference images with different kinds of artifacts are presented. The values of $RBEQ$ and $RTAEC$ for these images are given. The $RBEQ$ metric is more sensitive than $RTAEC$ to the image corruption with Gaussian blur, white Gaussian noise and unsharp mask artifacts (see Figures 6d, 6e and 6f). But in the case

of pixelized image the $RBEQ$ and $RTAEC$ values do not reflect properly the existing artifact (see Figure 6c).

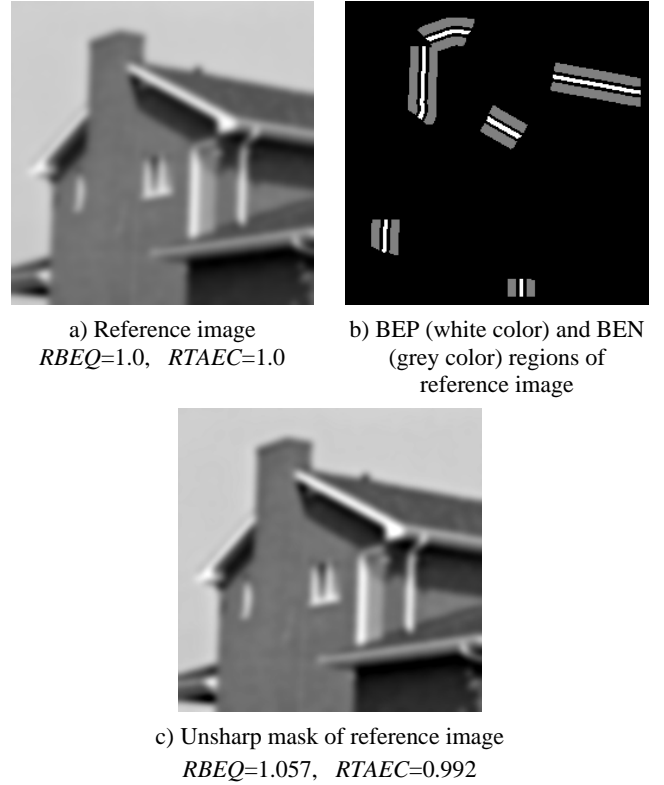


Figure 5: Comparison of $RBEQ$ and $RTAEC$ values for the unsharp masking result

In Figure 5 the result of unsharp mask of blurred “House” image is given. This illustrates the different behavior of $RBEQ$ and $RTAEC$ metrics. Despite the fact that unsharp mask is enhancing algorithm for blurred images $RTAEC < 1$ indicates the degradation of image quality. On the contrary, $RBEQ > 1$ indicates the improvement in image quality that corresponds to human perception.

It can be seen from the given examples that the $RBEQ$ and $RTAEC$ metrics both find the blurring of the image but the $RBEQ$ is much more sensitive to the presence of noise and it estimates better the unsharp masking results.

In Table 1 the $RBEQ$ and $RTAEC$ metrics are compared with other types of edge coherence metrics. Where $RBEQ_1$ is defined as:

$$RBEQ_1(I) = \frac{BEQ_1(I)}{BEQ_1(\tilde{I})}, \quad BEQ_1(I) = \frac{AEC_{BEP}(I)}{AEC_{BEN}(I)},$$

and $RTAEC_1$ is defined as:

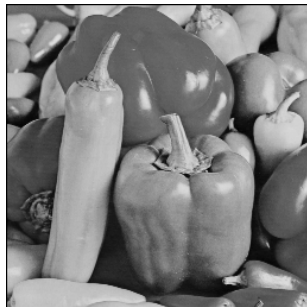
$$RTAEC_1(I) = \frac{MAEC(I)}{MAEC(\tilde{I})}.$$

It is illustrated that $RTAEC_1$ and $RBEQ_1$ metrics are not appropriate. $RTAEC_1 > 1$ in case of image quality degradation and $RTAEC_1 < 1$ in case of image quality improvement. The $RBEQ_1$ metric has the same drawback for images corrupted with white Gaussian noise and it is also less sensitive to the artifacts even in comparison with $RTAEC$ metric.

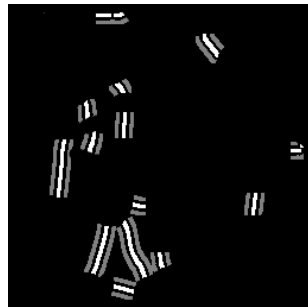
The given results demonstrate the sufficiency of using $MAEC$ metric in BEP and BEN regions to construct $RBEQ$ image quality metric.

Image	$RBEQ$	$RTAEC$	$RBEQ_1$	$RTAEC_1$
Figure 4c	0,530	0,982	1,002	1,289
Figure 4d	0,486	0,947	0,917	2,707
Figure 6c	0,643	0,748	0,981	1,638
Figure 6d	0,340	0,898	0,950	2,527
Figure 6e	0,241	0,999	1,005	1,088
Figure 6f	0,157	0,951	0,888	2,776
Figure 5c	1,057	0,992	1,011	0,955

Table 1: Comparison of $RBEQ$, $RTAEC$, $RBEQ_1$ and $RTAEC_1$ metrics.



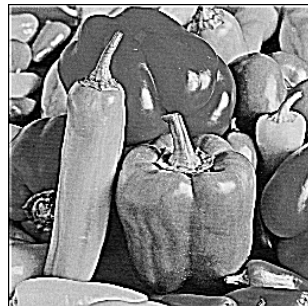
a) Reference image
 $RBEQ=1.0$, $RTAEC=1.0$



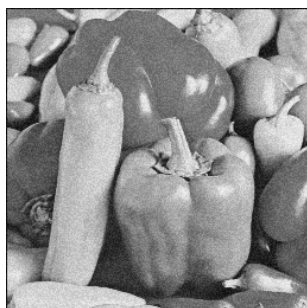
b) BEP (white color) and BEN (grey color) regions of reference image



c) Pixelized reference image
 $RBEQ=0.643$, $RTAEC=0.748$



d) An unsharp mask result for the reference image
 $RBEQ=0.340$, $RTAEC=0.898$



e) Reference image with white Gaussian noise
 $RBEQ=0.241$, $RTAEC=0.999$



f) Gaussian blurred reference image
 $RBEQ=0.157$, $RTAEC=0.951$

Figure 6: Comparison of $RBEQ$ and $RTAEC$ metrics.

5. CONCLUSION

A new short reference image quality metric using a modification of the angular edge coherence approach has been developed. The preliminary testing results look promising. The future work will include a detailed statistical analysis of the results for big image bases and a more detailed analysis of specific image artifacts like pixelization.

The work was supported by federal target program "Scientific and scientific-pedagogical personnel of innovative Russia in 2009-2013" and RFBR grant No 10-01-99535.

6. REFERENCES

- [1] M. P. Eckert and A. P. Bradley, "Perceptual quality metrics applied to still image compression" // *Signal Processing*, v. 70, pp. 177-200, Nov. 1998.
- [2] Z. Whang, E. Simoncelli "Reduced-Reference Image Quality Assessment Using A Wavelet-Domain natural Image Statistic Model" // *Human Vision and Electronic Imaging X, Proc. SPIE 2005*, vol. 5666.
- [3] M. Carnec, P. Le Callet, D. Barba, "Visual Features for Image Quality Assessment with Reduced Reference" // *Proc. of the IEEE ICIP 2005*, vol. 1, pp. 421-424.
- [4] P. Marziliano, F. Dufaux, S. Winkler, T. Ebrahimi, "Perceptual blur and ringing metrics: application to JPEG2000" // *Signal Processing: Image Communication* vol. 19, 2004, pp. 163-172.
- [5] G. L. Capodiferro, E. Di Claudio, G. Jacovitti, "Short Reference Image Quality Rating Based on Angular Edge Coherence" // *Proceedings of 14th European Signal Processing Conference, EUSIPCO 2006, Florence, Italy*.
- [6] V. Baroncini, L. Capodiferro, E. D. Di Claudio, G. Jacovitti, "The polar edge coherence: a quasi blind metric for video quality assessment" // *EUSIPCO 2009, Glasgow, 24-28 Aug. 2009*, pp.564-568.
- [7] Jacovitti, A. Neri, "Multiresolution circular harmonic decomposition" // *IEEE Trans. on Signal Processing*, vol. 48, no. 11, 2000, pp. 3242-3247.
- [8] A. Nasonov, A. Krylov "Adaptive Image Deringing" // *Proc. of GraphiCon'2009, Moscow, Russia, October 2009*, pp. 151-154.

About authors

Dmitry V. Sorokin is postgraduate of the Faculty of Computational Mathematics and Cybernetics, Moscow Lomonosov State University.
Email: dsorokin@cs.msu.ru

Andrey S. Krylov is professor, head of the Laboratory of Mathematical Methods of Image Processing, Faculty of Computational Mathematics and Cybernetics, Moscow Lomonosov State University.
Email: kryl@cs.msu.ru

Russian Section

Эффективный метод расчета освещения для сцен с большой глубиной зеркальных отражений и преломлений

Константин Востряков
Институт Прикладной Математики им. М.В.Келдыша
vostryakov@gin.keldysh.ru

Аннотация

Разработана модификация алгоритма стохастических прогрессивных фотонных карт, которая позволяет эффективно рассчитывать освещенность в сценах имеющих большую сложность зеркальных отражений и преломлений. В таких сценах большинство лучей света попадают в камеру претерпев десятки и даже сотни переотражений. Данный тип сцен редко встречается в компьютерных играх, производстве визуальных спецэффектов, но широко распространен в светотехнике и оптике, например, светопроводящих системах.

1 ВВЕДЕНИЕ

Светопроводящие системы появились еще в 19 веке. В них использовались зеркала внутри полой трубки, обеспечивая пропускание света по изогнутому туннелю благодаря зеркальным переотражениям от его стенок. Использование традиционных зеркал приводит к значительным потерям световой энергии из-за поглощения, поскольку наилучшие зеркала отражают лишь 95% падающего света. Таким образом после 10 переотражений остается лишь 60% первоначальной энергии света, а после 100 переотражений уже менее 1%. Как известно, световоды, изготовленные из стекла или пластика, могут обеспечивать передачу света более эффективно благодаря эффекту полного внутреннего отражения. Свет, многократно отражаясь от внутренних стенок световода, достигает противоположного конца проводника с очень маленьким поглощением.

Хотя световоды очень практичны при малых размерах и малых энергиях, они становятся тяжелыми и дорогими, если необходимо создать большую по размерам светопроводящую систему. Не так давно, в 1981 году, была предложена призматическая светопроводящая система. Пустая внутри, как и самые первые системы 19 века, но использующая вместо зеркал тонкие призматические пластинки. Эти пластинки имеют плоскую внутреннюю сторону и призматическую внешнюю. Поскольку эта система работает на принципе полного внутреннего отражения, то она практически также эффективна, как и световоды. Благодаря своей полой конструкции система имеет более малый вес и низкую цену в сравнении со световодами. Призматическая светопроводящая система может быть изготовлена достаточно больших размеров и служить для пропускания солнечного света внутрь зданий и сооружений, сокращая потребление электроэнергии. Она получила название светового колодца (light tube or light pipe).

Сегодня существует повсеместное распространение бытовой электроники. Вся она имеет пользовательский интер-

фейс со световой индикацией. Передача света от светодиода обеспечивается светопроводящей системой на основе обычных или призматических световодов, это позволяет более оптимально расположить все электронные компоненты внутри корпуса, на печатной плате и т.п.

Светопроводящие системы могут применяться для создания источников света сложной формы. От источника параболическим рефлектором свет направляется в призматический световод. Многократно отражаясь от стенок световода, неравномерности излучения источника теряются, свет относительно равномерно распределяется по всей длине световода. Через многочисленные отверстия свет покидает световод. Размер многочисленных отверстий позволяет создать желаемое распределение света вокруг оптической системы. Такие светопроводящие системы позволяют создать источники света сложной формы, что очень привлекательно для светового дизайна. А использование светодиодов позволяет добиться компактности и энергоэффективности.

2 СПЕКУЛЯРНАЯ СЛОЖНОСТЬ

Для автоматизации проектирования оптических систем, примеры которых приведены выше, требуется моделирование распространения света. Но такие объекты имеют большую, так называемую, спекулярную сложность. Это значит, что большинство лучей света попадают в камеру претерпев десятки и даже сотни зеркальных отражений и преломлений. Спекулярная сложность значительно увеличивает время расчета, путь света становится очень извилистым, лишь ничтожная часть всех траекторий световых путей соединяет источники света и камеру. Расчет глобальной освещенности становится затруднительным как для прямого, так и обратного метода Монте-Карло трассировки лучей.

В алгоритме трассировки лучей Whitted'a [1], после столкновения луча с поверхностью, отраженный и преломленный лучи трассируются последовательно, что приводит к экспоненциальному увеличению числа возможных траекторий. Монте-Карло трассировка путей (path tracing) [2] позволяет смягчить проблему, случайно выбирая отражение, пропускание или поглощение, на основе коэффициентов спекулярного отражения и пропускания. Так выбираются более значимые траектории. Но коэффициенты отражения и пропускания дают доступ лишь к локальной информации о значимости данной траектории. При большой спекулярной сложности это оказывается недостаточно. То же самое можно сказать про Марковские методы Монте-Карло [2], которые используют предыдущую траекторию для построения следующей. Трудно найти важные траектории в таких сценах на основе локальной информации.

Как было показано в работе [3] метод фотонных карт более эффективен для расчета каустик. При этом он может выполняться в прогрессивной манере, обеспечивая сходимости к точному решению при наличии конечной памяти, т.е. оценка световой плотности статистически состоятельна. Суть прогрессивного стохастического метода фотонных карт состоит в последовательном выполнении следующих действий на каждой итерации:

- прямая трассировка световых траекторий с сохранением точек столкновения с поверхностями сцены (фотонов);
- обратная трассировка световых траекторий с сохранением точек столкновения с поверхностями сцены;
- оценка световой плотности в точках обратной трассировки, используя точки прямой трассировки из окрестности.

Уменьшение окрестности поиска после каждой итерации обеспечивает постоянное сокращение ошибки дискретизации, т.е. состоятельность оценки. К сожалению, данный метод позволяет эффективно рассчитывать яркость только для неспекулярных функций отражения, которая наблюдается лишь через небольшое число спекулярных переотражений.

3 МОДИФИКАЦИЯ МЕТОДА ПРОГРЕССИВНЫХ ФОТОННЫХ КАРТ

Предлагаемый алгоритм, основываясь на методе прогрессивных фотонных карт, позволяет эффективно рассчитать сцены большой сложности. Ограничив глубину обратной трассировки лучей удалось получить, пусть и смещенную, но все-таки состоятельную Монте-Карловскую оценку световой плотности, то есть сходящуюся к точному решению.

Оценка яркости света в классическом методе фотонных карт может быть получена, используя уравнение визуализации, следующим образом.

$$L(x, \omega) = \int_{\Omega} p(x, \omega, \omega'_p) \frac{d^2\Phi(x, \omega'_p)}{dA} \quad (1)$$

$$= \sum_{p=1}^n p(x, \omega, \omega'_p) \frac{\Phi_p(x, \omega'_p)}{\Delta A}, \quad (2)$$

где рассеянное поверхностью излучение $L(x, \omega)$ в точке x в направлении ω равно интегралу по всем направлениям на полусфере Ω по произведению функции отражения и пропускания $p(x, \omega, \omega')$, падающего дифференциального потока $\Phi(x, \omega')$ по площади поверхности и телесному углу, косинуса угла между нормалью к поверхности и направлением ω' . В уравнении (2) имеем сумму произведений $p(x, \omega, \omega')$ на мощности ближайших фотонов деленные на площадь окрестности.

В модифицированном алгоритме при достижении фиксированной максимальной глубины траектории идущей из камеры берется оценка световой плотности, подобно уравнению (2), но уже используя фотоны из окрестности зеркального луча (рис. 1). Полученная световая яркость умножается на спекулярный коэффициент поверхности

и коэффициент передачи траектории от камеры до этой точки:

$$L(x, \omega) = \sum_{p=1}^n p_s(x, \omega, \omega'_p) \frac{\Phi_p(x, \omega'_p)}{\Delta A \Delta \Omega}, \quad (3)$$

где p_s значение коэффициента зеркального отражения (преломления), $\Delta A = \pi r^2$, $\Delta \Omega = 2\pi(1 - \cos\theta_{max})$, а r и θ_{max} — это текущие радиус и угол дискретизации оценки.

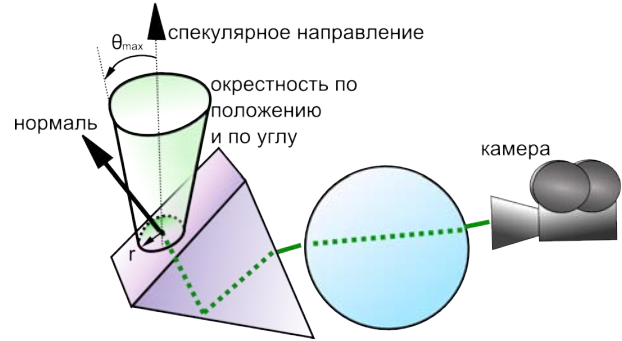


Рис. 1: Достигнута фиксированная максимальная глубина траектории. В окрестности спекулярного луча выполняется поиск фотонов.

4 ЗАКЛЮЧЕНИЕ

Разработана модификация алгоритма стохастических прогрессивных фотонных карт, которая позволяет эффективно рассчитывать освещенность в сценах имеющих большую глубину зеркальных отражений и преломлений. Данный тип сцен редко встречается в компьютерных играх, производстве визуальных спецэффектов, но широко распространен в светотехнике и оптике, например, светопроводящих системах. Ограничив глубину обратной трассировки лучей удалось получить, пусть и смещенную, но все-таки состоятельную Монте-Карловскую оценку световой плотности, то есть сходящуюся к точному решению. Эффективное моделирование таких сцен позволяет создать более совершенные системы автоматизированного оптического проектирования.

Работа поддержана грантами Президента РФ НИИ-8129.2010.9, РФФИ № № 09-01-00472, 10-01-00302, а также компанией Integra Inc.

Список литературы

- [1] Turner Whitted. An improved illumination model for shaded display. *Commun. ACM*, 23(6):343–349, 1980. 2
- [2] Eric Veach and Leonidas J. Guibas. Metropolis light transport. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 65–76, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co. 2
- [3] Toshiya Hachisuka and Henrik Wann Jensen. Stochastic progressive photon mapping. *ACM Trans. Graph.*, 28(5):1–8, 2009. 2

Построение спектрального представления из RGB данных в задачах спектрального моделирования

Жданов Дмитрий Дмитриевич

Государственный Оптический Институт им. С.И. Вавилова, Санкт-Петербург

Потемин Игорь Станиславович

Институт Прикладной Математики им. М.В.Келдыша РАН, Москва

Аннотация

Данная статья посвящена физически корректному компьютерному моделированию сцен, содержащих объекты, оптические свойства которых определены в различных пространствах, например, в спектральном и RGB пространствах. Для уменьшения ошибок цветопередачи предлагается перевод всех данных в единое спектральное пространство. В работе предложен метод физически корректного преобразования RGB данных источников света и оптических свойств поверхностей в спектральное представление. В отличие от большинства предлагаемых методов, разработанное решение обеспечивает точное преобразование RGB цвета как для источников света, так и для свойств оптических поверхностей.

Ключевые слова: Цветопередача, RGB, XYZ, CIE, рендеринг, спектральные оптические свойства, спектральная плотность, спектральная трассировка лучей, фотореалистичность.

1. ВВЕДЕНИЕ

Построение фотореалистичных изображений требует физически корректного представления оптических свойств сцены. Для источников света и оптических свойств поверхностей единственным корректным представлением является спектральное представление данных. В сложных сценах достаточно трудно добиться однородного спектрального представления данных. Как правило, в «спектральных» сценах ряд оптических свойств остается определенным в RGB пространстве [1, 2]. Отсутствие спектральных данных объясняется трудоемкостью спектральных измерений и наличием у ряда стандартных библиотечных объектов, например, текстур или HDR1 источников света, исключительно RGB представления.

Для сцен со смешанными оптическими свойствами трассировка лучей в двух пространствах (RGB и спектральном) одновременно задача алгоритмически сложная и не всегда достаточно точная. В процессе трассировки возможно формирование отрицательных RGB цветов (как результат конвертации спектральных данных луча в локальное представление RGB поверхности). Естественно, отрицательные значения исключаются, но точность цветопередачи снижается. Поэтому аккуратным решением будет преобразование всех RGB данных в спектральное представление.

Известно, что преобразование спектральных данных в RGB представление выполняется однозначным образом, однако

обратное преобразование RGB данных в спектральное представление неоднозначно. Не имея никакой информации об исходном спектре излучения, задача не может быть однозначно решена. Поэтому, как правило, для решения данной задачи используется ряд допущений. Во-первых, предполагается, что спектральные свойства должны быть представлены как можно в более широкой области спектра и поэтому оказывать наименьшее влияние на истинно спектральные данные. Например, если спектрограмма луча имеет особенность, как то ярко выраженный максимум, то взаимодействие луча с поверхностью, заданной в RGB пространстве, должно по возможности сохранять особенность спектрограммы луча. Это означает, что спектрограмма, полученная из RGB данных, должна быть максимально широкой и равномерной. Во-вторых, это касается спектральных свойств поверхностей, интегральные спектральные коэффициенты отражения / пропускания которых не должны превышать 1.

Для решения задачи конвертации RGB данных в спектральное представление был предложен ряд подходов и алгоритмов [3, 4, 5, 6, 7], базирующихся на выборе базовых функций (спектрограмм), взвешивая которые R, G и B цветами, можно получить спектрограмму требуемого RGB цвета.

Основными недостатками этих подходов являются либо неточность конвертации, то есть конвертация RGB в спектральное представление и обратно в RGB не дает исходного RGB цвета; либо неоднородность спектрограммы, то есть спектрограмма не плавная функция, а набор острых пиков; либо формирование спектрограмм с отрицательными значениями; либо превышение спектрального отражения / пропускания поверхности 1. Основным преимуществом данных подходов является эффективность преобразования цвета, что делает возможным осуществлять переход от RGB к спектральному представлению в процессе трассировки лучей.

Для формирования «реалистичных» спектрограмм из RGB данных в работе [10] был предложен алгоритм использования ограниченного набора спектральных точек сложного цветного объекта, например, спектрограмм для ограниченного набора углов двунаправленной функции отражения. Основным ограничением данного подхода является наличие спектральных данных, которые не всегда доступны.

Данная работа предлагает точное решение преобразования RGB данных в спектральное представление. Хотя эффективность данного решения не так высока, как у подходов, приведенных выше, это не является столь критичным, поскольку преобразование цветов можно

выполнять в момент загрузки сцены и, хотя загрузка сцены станет несколько длительнее, трассировка лучей будет эффективнее, поскольку сцена будет определена в однородном спектральном пространстве.

2. АЛГОРИТМ ПРЕОБРАЗОВАНИЯ RGB ДАННЫХ В СПЕКТРАЛЬНОЕ ПРЕДСТАВЛЕНИЕ

Предлагаемое решение основывается на решении интегральных уравнений преобразования спектральных данных в RGB представление. RGB цвет вычисляется в соответствии с известными CIE формулами [8]:

$$\left. \begin{aligned} x &= \int_{380}^{780} \bar{x}_{\lambda} \varphi(\lambda) d\lambda \\ y &= \int_{380}^{780} \bar{y}_{\lambda} \varphi(\lambda) d\lambda \\ z &= \int_{380}^{780} \bar{z}_{\lambda} \varphi(\lambda) d\lambda \end{aligned} \right\} \Rightarrow rgb = M \cdot xyz \quad (1)$$

где:

$\bar{x}_{\lambda}, \bar{y}_{\lambda}, \bar{z}_{\lambda}$ – функции чувствительности XYZ для стандартного наблюдателя,

$\varphi(\lambda)$ – функция плотности распределения яркости (интенсивности) или другой световой величины,

x, y, z – координаты цвета в XYZ пространстве,

M – матрица преобразования координат цвета из XYZ в RGB пространство.

Поскольку преобразование RGB в XYZ однозначно, то интегральные уравнения можно решать относительно x, y и z координат цвета. Для решения заменим интегралы суммами с постоянным шагом по длине волны и получим систему линейных уравнений от значений спектральной плотности (искомой спектрограммы RGB цвета):

$$\left. \begin{aligned} x &= \Delta\lambda \sum_{i=1}^N \bar{x}_{\lambda_i} \varphi(\lambda_i) \\ y &= \Delta\lambda \sum_{i=1}^N \bar{y}_{\lambda_i} \varphi(\lambda_i) \\ z &= \Delta\lambda \sum_{i=1}^N \bar{z}_{\lambda_i} \varphi(\lambda_i) \end{aligned} \right\} \quad (2)$$

Очевидно, что при $N > 3$ система имеет множество решений. Решение системы (2) может быть найдено как линейная комбинация частных решений. Алгоритм решения следующий:

Шаг 1. Циклически, для всех длин волн спектрального интервала выбирается 3 новых, несовпадающих длины волны ($\lambda_i, \lambda_j, \lambda_k$).

Шаг 2. Находится частное l -ое решение уравнения для этих волн ($\lambda_i, \lambda_j, \lambda_k$):

$$\left. \begin{aligned} x &= \Delta\lambda(\bar{x}_{\lambda_i} \varphi_i(\lambda_i) + \bar{x}_{\lambda_j} \varphi_i(\lambda_j) + \bar{x}_{\lambda_k} \varphi_i(\lambda_k)) \\ y &= \Delta\lambda(\bar{y}_{\lambda_i} \varphi_i(\lambda_i) + \bar{y}_{\lambda_j} \varphi_i(\lambda_j) + \bar{y}_{\lambda_k} \varphi_i(\lambda_k)) \\ z &= \Delta\lambda(\bar{z}_{\lambda_i} \varphi_i(\lambda_i) + \bar{z}_{\lambda_j} \varphi_i(\lambda_j) + \bar{z}_{\lambda_k} \varphi_i(\lambda_k)) \end{aligned} \right\} \quad (3)$$

Шаг 3. Если решение системы линейных уравнений (3) существует ($\varphi_i(\lambda_i), \varphi_i(\lambda_j), \varphi_i(\lambda_k)$) и все спектральные плотности являются неотрицательными числами, то данное решение есть один из элементов общего решения, и оно добавляется к общему решению:

$$\left. \begin{aligned} \varphi(\lambda_i) &= \varphi(\lambda_i) + \varphi_i(\lambda_i) \\ \varphi(\lambda_j) &= \varphi(\lambda_j) + \varphi_i(\lambda_j) \\ \varphi(\lambda_k) &= \varphi(\lambda_k) + \varphi_i(\lambda_k) \end{aligned} \right\} \quad (4)$$

Шаг 4. Нормировка спектрограммы. После окончания цикла (шаг 1) мы имеем ненормированную спектрограмму $\varphi(\lambda_i)$.

Очевидно, что для данного решения нормировочный множитель спектрограммы это общее количество найденных допустимых решений L .

Приводя результат работы алгоритма к исходной системе линейных уравнений (2) можно записать:

$$\left. \begin{aligned} x &= \Delta\lambda \sum_{i=1}^N \bar{x}_{\lambda_i} \frac{1}{L} \sum_{l=1}^{L_i} \varphi_l(\lambda_i) \\ y &= \Delta\lambda \sum_{i=1}^N \bar{y}_{\lambda_i} \frac{1}{L} \sum_{l=1}^{L_i} \varphi_l(\lambda_i) \\ z &= \Delta\lambda \sum_{i=1}^N \bar{z}_{\lambda_i} \frac{1}{L} \sum_{l=1}^{L_i} \varphi_l(\lambda_i) \end{aligned} \right\} \quad (5)$$

где:

L_i – общее количество найденных допустимых решений для длины волны λ_i ,

$\varphi_l(\lambda_i)$ – найденное частное решение l -ой функции спектральной плотности для длины волны λ_i .

Очевидно, что полученная спектрограмма удовлетворяет условию максимальной представительности, поскольку в ней присутствуют все возможные решения для всего видимого спектра.

Для вычисления спектральных коэффициентов отражения (пропускания) поверхности выражения (1) – (5) непосредственно не применимы. Необходимо помнить, что поверхность не обладает собственным цветом, а под цветом поверхности, как правило, понимают нормированный отраженный (преломленный или рассеянный) цвет от стандартного белого источника света, например, D65. Поэтому аналогом выражений (1) для цвета поверхности является следующий набор выражения:

$$\left. \begin{aligned} \left| \begin{aligned} x_{D65} &= \int_{380}^{780} \bar{x}_{\lambda} D65(\lambda) d\lambda \\ y_{D65} &= \int_{380}^{780} \bar{y}_{\lambda} D65(\lambda) d\lambda \\ z_{D65} &= \int_{380}^{780} \bar{z}_{\lambda} D65(\lambda) d\lambda \end{aligned} \right| \left| \begin{aligned} x_{\rho} &= \int_{380}^{780} \bar{x}_{\lambda} \rho(\lambda) D65(\lambda) d\lambda \\ y_{\rho} &= \int_{380}^{780} \bar{y}_{\lambda} \rho(\lambda) D65(\lambda) d\lambda \\ z_{\rho} &= \int_{380}^{780} \bar{z}_{\lambda} \rho(\lambda) D65(\lambda) d\lambda \end{aligned} \right| \quad (6) \\ rgb_{D65} &= M \cdot xyz_{D65} & rgb_{\rho} &= M \cdot xyz_{\rho} \\ r &= \frac{r_{\rho}}{r_{D65}}; & g &= \frac{g_{\rho}}{g_{D65}}; & b &= \frac{b_{\rho}}{b_{D65}}; \end{aligned}$$

где:

$D65(\lambda)$ – спектрограмма белого источника D65, освещающего поверхность,

$\rho(\lambda)$ – спектральные свойства отражения (пропускания) поверхности.

Поскольку спектрограмма опорного источника D65 определена, для вычисления спектрограммы отражения (пропускания) поверхности $\rho(\lambda)$ применим алгоритм (2) – (5), где вместо спектральной излучательности $\varphi_i(\lambda_i)$ вычисляется $\rho_i(\lambda_i)$. Дополнительным ограничением для вычисления спектрограммы поверхности является условие $\frac{1}{L} \sum_{i=1}^L \varphi_i(\lambda_i) \leq 1$.

3. ПРИМЕНЕНИЕ РАЗРАБОТАННОГО АЛГОРИТМА ДЛЯ ПОСТРОЕНИЯ ФОТОРЕАЛИСТИЧНЫХ ИЗОБРАЖЕНИЙ

Разработанный алгоритм был интегрирован в программный комплекс построения фотореалистичных изображений SPECTER [9]. Все данные оптических свойств поверхностей и источников сцены, определенные в RGB пространстве, переводятся в спектральное представление в процессе обработки сцены перед началом рендеринга. Для преобразования оптических свойств поверхностей сцены из RGB в спектральное представление использовался опорный источник D65. Спектральные данные, преобразованные из RGB цветов, дают неотличимый результат от исходного RGB представления. Ниже приводятся изображения, выполненные в исходном RGB и конвертированном спектральном пространствах.

Для удобства сравнения результатов моделирования использовалась зеркально симметричная сцена (сфера, освещенная точечным источником света, и наблюдателем, находящимся на линии от источника света к центру сферы).

В качестве первого примера была рассмотрена конвертация источника света с цветами RGB = (6000, 5000, 1000)лм в спектральное представление. На рисунке 1 показана спектрограмма данного источника света, полученная в результате конвертации.

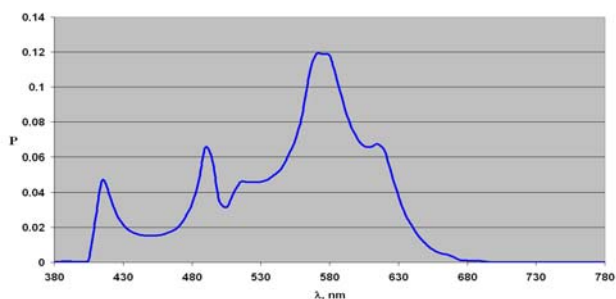


Рисунок 1: Спектрограмма источника света для RGB = (6000, 5000, 1000)лм.

На рисунке 2 показано изображение белой сферы, освещенной цветным источником света (RGB = (6000, 5000, 1000)лм). Изображение левой половины сферы было выполнено в RGB пространстве, а изображение правой половины – в спектральном пространстве со спектрограммой источника света, представленной на рисунке 1. Изображения половинок неразличимы, что говорит о том, что ошибка данного метода визуально неопределима.

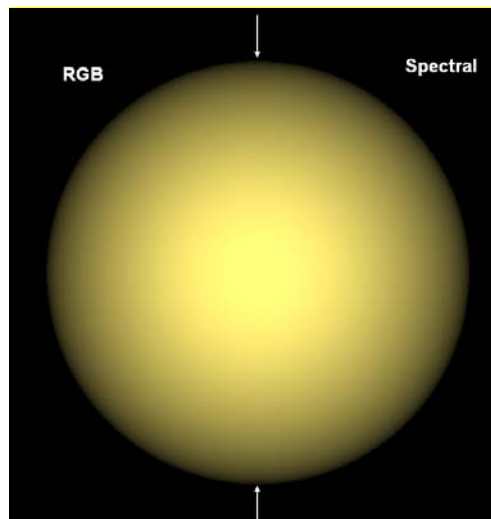


Рисунок 2: Изображения сферы. Левая половина построена в RGB пространстве, правая половина в спектральном пространстве.

Необходимо отметить, что из-за слабой чувствительности глаза на краях спектрального диапазона (фиолетовой и красной областях) решение уравнения (5) может приводить к резким пикам в данных краевых зонах. Значения на краях спектрального интервала могут на порядок превышать среднее значение спектрограммы. Хотя данные пики не оказывают практически никакого влияния на цвет, они выглядят неестественным образом и должны быть устранены. Для фильтрации спектрограмм источников света используется простой алгоритм, идея которого заключается в следующем. Частные решения (3) сортируются в порядке возрастания максимального значения одной из спектральных плотностей $\varphi_i(\lambda_i), \varphi_j(\lambda_j), \varphi_k(\lambda_k)$, и K последних решений, которые приводят к появлению краевых пиков, исключаются из решения (5).

В качестве второго примера была рассмотрена конвертация цветного коэффициента яркости диффузной поверхности с цветами RGB = (0.21, 0.72, 0.15) в спектральное представление. На рисунке 3 показан спектральный коэффициент яркости данной поверхности, полученный в результате конвертации.

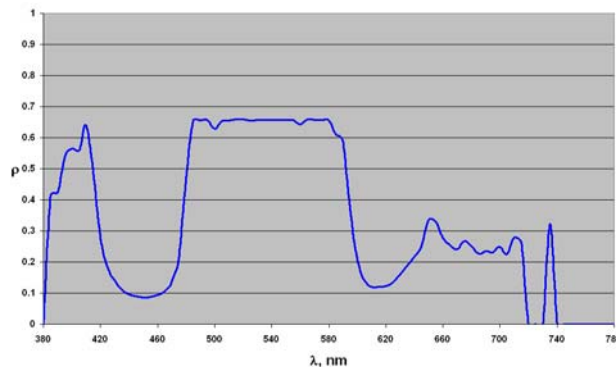


Рисунок 3: Спектральный коэффициент яркости диффузной поверхности для RGB = (0.21, 0.72, 0.15)лм.

На рисунке 4 показано изображение зеленой сферы ($RGB = (0.21, 0.72, 0.15)$), освещенной белым источником света D65. Изображение левой половины сферы было выполнено в RGB пространстве, а изображение правой половины – в спектральном пространстве со спектральными свойствами поверхности, представленными на рисунке 3. Как и для цвета поверхности, изображения половинок неразличимы, что говорит о том, что ошибка данного метода визуально неопределима.

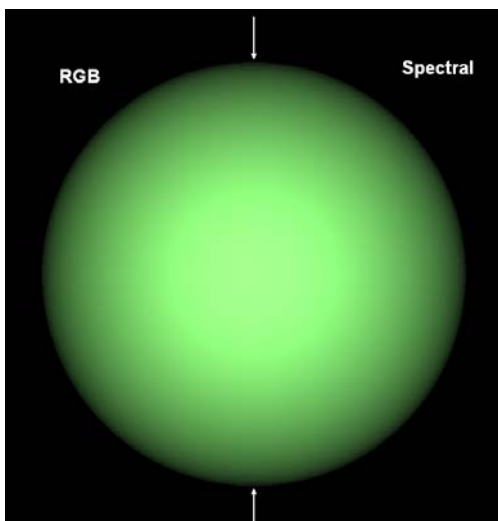


Рисунок 4: Изображения сферы. Левая половина построена в RGB пространстве, правая половина в спектральном пространстве.

При конвертации RGB данных в спектральные свойства поверхности использовался дополнительный критерий: спектральное отражение (пропускание) не должно превышать 1. В сочетании с фильтрацией, используемой для источников света, результирующая спектрограмма становится, как правило, более плоской, что видно на рисунке 3.

4. ЗАКЛЮЧЕНИЕ

Хотя эффективность предлагаемого метода невысока, он обеспечивает высокую точность конвертации и представительность спектральных данных, т.е. определенность в широкой области спектра. Данный метод применим в случаях, когда требуется построение полной спектральной сцены, и не предполагается быстрая (хотя и не совсем точная) конвертация RGB данных в спектральное представление в процессе трассировки луча.

Разработанные алгоритмические решения были интегрированы в программный комплекс СПЕКТЕР, что позволило выполнять фотореалистичный спектральный рендеринг сцен, включающих, наряду со спектральными оптическими свойствами, свойства, заданные в RGB пространстве.

5. БЛАГОДАРНОСТИ

Работа поддержана грантами Президента РФ НШ-8129.2010.9, РФФИ № 10-01-00302, а также компанией Integra Inc.

Авторы выражают свою признательность В.Г. Соколову и А.А. Гарбулю за предоставленные примеры.

6. ABSTRACT

The article is devoted to physically correct light simulations of the scenes with mixture of optical properties defined both in RGB and spectral spaces. To eliminate inaccuracy of the color reproductions the article proposes a reduction of all optical properties to single spectral space and an accurate method of the reduction. As opposed to most of methods the solution proposes the precise solution which is applied both for light source color conversions and conversion of surface reflectances and transmittances.

Keywords: Color reproduction, RGB, XYZ, rendering, spectral optical properties, spectral density, spectral ray tracing, photorealism.

7. REFERENCES

- [1] Roman Durikovic, Ryou Kimura. Spectrum-based rendering using programmable graphics hardware / SCCG 2005: Proceedings of the 21st spring conference on Computer graphics.
- [2] Greg Ward, Elena Eydelberg-Vileshin. Picture perfect RGB rendering using spectral prefiltering and sharp color primaries / EGRW 2002: Proceedings of the 13th Eurographics workshop on Rendering.
- [3] Hiroyuki Fukuda, Junko Kishimoto, Masahiro Yamaguchi, Hideaki Haneishi, Nagaaki Ohyama. Spectral-based image-editing system / SIGGRAPH 2005.
- [4] Brian Smits. An RGB to Spectrum Conversion for Reflectances / University of Utah, January 21, 2000.
- [5] Glassner, A. S. How to derive a spectrum from an RGB triplet / IEEE Computer Graphics and Applications 9, 4 (July 1989), 95–99.
- [6] Glassner, A. S. Principles of Digital Image Synthesis // Morgan-Kaufman, San Francisco, 1995.
- [7] Sun, Y., Fracchia, F. D., Calvert, T. W., and Drew, M. S. Deriving spectra from colors and rendering light interference / IEEE Computer Graphics and Applications 19, 4 (July/August 1999).
- [8] Д.С. Волосов, М.В. Цивкин. Теория и расчет светоптических систем. // «Искусство», Москва – 1960.
- [9] <http://www.integra.jp/en/index.html>
- [10] Martin Rump, Reinhard Klein. Spectralization: Reconstructing Spectra from Sparse Data / In proceedings of SR '10 Rendering Techniques, pages 1347-1354, Eurographics Association, June 2010

СПЕКТРАЛЬНАЯ ТРАССИРОВКА ЛУЧЕЙ В ЗАДАЧАХ ПОСТРОЕНИЯ ФОТОРЕАЛИСТИЧНЫХ ИЗОБРАЖЕНИЙ

Б.Х. Барладян¹, К.А. Востряков¹, В.А. Галактионов¹, Д.Д. Жданов², И.С. Потемин¹, Л.З. Шапиро¹

¹Институт Прикладной Математики им. М.В.Келдыша РАН, Москва

²Государственный Оптический Институт им. С.И. Вавилова, Санкт-Петербург

Аннотация

Данная статья посвящена анализу ошибок цветопередачи, возникающих при компьютерном построении изображения сцен со сложными оптическими свойствами. В работе произведен анализ причин возникновения ошибок цветопередачи при построении изображений в RGB пространстве, и предложен эффективный и аккуратный алгоритм решения задач построения спектрального изображения. Предложенные решения были реализованы для программ прямой стохастической и обратной детерминистической трассировки лучей. Кроме того, для сложных сцен, содержащих смесь оптических свойств, определенных в различных моделях, включая RGB, было предложено эффективное и аккуратное решение для приведения всех оптических свойств к единой спектральной модели.

Ключевые слова: Цветопередача, RGB, XYZ, рендеринг, спектральные оптические свойства, спектральная плотность, спектральная трассировка лучей, фотореалистичность.

1. Введение

Построение фотореалистичных изображений сложных сцен, содержащих объекты со специальными свойствами пропускания, отражения и рассеивания, требует применения физически аккуратных моделей расчета яркости, формируемой данными объектами. Как правило, программы компьютерного построения изображений используют RGB представление для описания оптических свойств геометрических объектов сцены и источников света. Такое приближение используется не только в «аппаратных» системах [1], основанных на OpenGL, но и в большинстве программ, декларирующих физическую аккуратность и фотореалистичность формируемого изображения [2, 3].

Как известно, RGB пространство ограничено, и представление положительных цветов не охватывает весь видимый спектральный диапазон. В данном контексте необходимо различать два момента, связанных с клиппированием и искажением представления цвета.

Во-первых, это искажение цвета, возникающее при расчете видимой яркости объектов сцены. Яркость в RGB представлении пропорциональна произведению компонент R, G и B освещенности объекта на соответствующие компоненты коэффициента яркости объекта в направлении наблюдения. Данное соотношение не вызывает искажения цвета только в том случае, если коэффициент яркости был изначально определен для цвета заданного источника света. Очевидно, что для сцен, в которых наблюдение осуществляется сквозь цветные стекла или зеркала, или для

сцен, содержащих разноцветные источники излучения, исключить искажения цвета практически невозможно [4].

Во-вторых, это клиппирование цвета на приемниках излучения. Большинство современных приемников излучения не могут воспроизвести видимую яркость или освещенность без искажения цвета. Такие приемники излучения как CCD матрицы трансформируют спектральное распределение освещенности в RGB изображение. Естественно, такая трансформация теряет отрицательную составляющую цвета, и это невозможно исправить. Поэтому, когда мы говорим о формировании фотореалистичного изображения, мы должны понимать, что «фотореалистичность» может уже содержать искажение цвета, и задача построения фотореалистичного изображения должна включать соответствующую модель приемника излучения [5].

Очевидно, задача построения фотореалистичных изображений содержит две независимые модели. Во-первых, построение истинного распределения освещенности или яркости на приемнике излучения, и, во-вторых, преобразование входного излучения в RGB представление. Данная работа посвящена исключительно первому аспекту, а именно, формированию распределения освещенности и яркости без искажения цвета. Единственная модель, которая гарантирует правильную цветопередачу, – спектральная модель, а единственный метод расчета освещенности и яркости – это спектральная трассировка лучей. Поэтому основной акцент данной работы – это физически точная трассировка спектральных лучей в сценах, содержащих объекты со сложными оптическими свойствами.

2. Базовые ограничения RGB модели и ошибки вычисления цвета

Преобладающее большинство компьютерных систем построения изображений используют аддитивное RGB представление цвета в описаниях оптических свойств объектов сцены. Объекты сцены, обладающие цветными свойствами, можно разделить на две основные группы. Во-первых, это первичные источники, излучающие свет, и, во-вторых, это вторичные источники, отражающие, преломляющие и рассеивающие свет.

В случае первичного источника света RGB цвет вычисляется в соответствии с известными CIE формулами [6]:

$$\left. \begin{aligned} x &= \int_{380}^{780} \bar{x}_{\lambda} \varphi(\lambda) d\lambda \\ y &= \int_{380}^{780} \bar{y}_{\lambda} \varphi(\lambda) d\lambda \\ z &= \int_{380}^{780} \bar{z}_{\lambda} \varphi(\lambda) d\lambda \end{aligned} \right\} \Rightarrow rgb = M \cdot xyz \quad (1)$$

где:

$\bar{x}_\lambda, \bar{y}_\lambda, \bar{z}_\lambda$ – функции чувствительности XYZ для стандартного наблюдателя,

$\rho(\lambda)$ – функция плотности распределения яркости (или другой фотометрической величины),

x, y, z – координаты цвета в XYZ пространстве,

M – матрица преобразования координат цвета из XYZ в RGB пространство [12].

Известно, что хотя представление цвета в виде RGB каналов охватывает весь видимый спектральный диапазон, составляющие цвета могут быть отрицательными числами.

В компьютерных вычислениях, как и в системах отображения цифрового изображения, используются положительные значения для отображения RGB цвета. Очевидно, что такое ограничение цветового представления приводит к ошибкам отображения цвета. Для оценки ошибки цветоощущения, как правило, используется выражение цвета в Lab пространстве. Международным комитетом CIE был определен ряд стандартов цветового различия: CIE76 [7], CIE94 и CIEDE2000 [8]. Используя формулы цветового различия, можно оценить ошибку цветоощущения (ΔE_{ab}^*), вызванную клиппированием отрицательных значений RGB пространства.

В случае вторичного источника света более корректно говорить о цвете поверхности как о реакции поверхности на падающий свет. Поскольку поверхность не обладает собственным цветом, то в RGB пространстве под цветом поверхности, как правило, понимают нормированный отраженный (преломленный или рассеянный) цвет от стандартного белого источника света, например, D65. Общий алгоритм вычисления RGB цвета поверхности выглядит следующим образом:

$$\begin{cases} x_{D65} = \int_{380}^{780} \bar{x}_\lambda D65(\lambda) d\lambda \\ y_{D65} = \int_{380}^{780} \bar{y}_\lambda D65(\lambda) d\lambda \\ z_{D65} = \int_{380}^{780} \bar{z}_\lambda D65(\lambda) d\lambda \end{cases} \quad \begin{cases} x_\rho = \int_{380}^{780} \bar{x}_\lambda \rho(\lambda) D65(\lambda) d\lambda \\ y_\rho = \int_{380}^{780} \bar{y}_\lambda \rho(\lambda) D65(\lambda) d\lambda \\ z_\rho = \int_{380}^{780} \bar{z}_\lambda \rho(\lambda) D65(\lambda) d\lambda \end{cases} \quad (2)$$

$$rgb_{D65} = M \cdot xyz_{D65} \quad rgb_\rho = M \cdot xyz_\rho$$

$$r = \frac{r_\rho}{r_{D65}}; \quad g = \frac{g_\rho}{g_{D65}}; \quad b = \frac{b_\rho}{b_{D65}};$$

где:

$\rho(\lambda)$ – спектральные свойства отражения (пропускания) поверхности.

В реальных сценах, как правило, используется сложное освещение, в котором присутствуют источники света различного цвета, и, кроме того, зеркальные и преломляющие цветные объекты добавляют виртуальные цветные источники освещения. В результате в реальной сцене не существует такого источника света, чей цвет можно было бы использовать как базовый цвет для расчета цветов объектов сцены. Поэтому для расчета цветов объектов сцены, в качестве базового цвета, как правило, используется белый цвет, например, источник D65. Очевидно, что такое решение может привести к искажению цвета объекта в реальной сцене.

3. Проблемы цветопередачи при решении уравнения рендеринга в RGB пространстве

В предыдущей главе рассматривались основные проблемы цветопередачи при выполнении компьютерного моделирования в RGB пространстве. В данной главе будут рассмотрены проблемы решения уравнения рендеринга в RGB пространстве. Уравнение рендеринга определяет яркость в направлении наблюдения как сумму собственной яркости наблюдаемого объекта и яркости, рассеянной данным объектом в направлении наблюдения [3, 9]. Для статических сцен уравнение рендеринга, определяющее яркость объекта в точке \vec{p} , направлении \vec{v} и для цветовой компоненты c , можно записать следующим образом:

$$L(\vec{p}, \vec{v}, c) = \tau(\vec{p}, \vec{v}, c) \left(L_0(\vec{p}, \vec{v}, c) + \frac{1}{\pi} \int_{4\pi} BPDF(\vec{p}, \vec{v}, \vec{v}', c) E_o(\vec{p}, \vec{v}', c) d\omega \right) \quad (3)$$

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,

$BPDF(\vec{p}, \vec{v}, \vec{v}', c)$ – функция двунаправленного рассеивания от источника освещения в направлении \vec{v}' на наблюдателя,

$E_o(\vec{p}, \vec{v}', c)$ – локальная освещенность объекта в точке наблюдения по направлению \vec{v}' , созданная источником света в телесном угле $d\omega$.

Основным и аккуратным методом решения уравнения рендеринга является метод трассировки лучей. В лучевом подходе интегрирование по сфере выполняется посредством трассировки лучей в направлении всех возможных источников излучения, и уравнение рендеринга принимает следующий вид:

$$L(\vec{p}, \vec{v}, c) = \tau(\vec{p}, \vec{v}, c) \left(L_0(\vec{p}, \vec{v}, c) + \frac{1}{\pi} \sum_{\text{for all rays}} BPDF(\vec{p}, \vec{v}, \vec{v}', c) E_i(\vec{p}, \vec{v}', c) \right) \quad (4)$$

где:

$E_i(\vec{p}, \vec{v}', c)$ – локальная освещенность объекта в точке наблюдения по направлению \vec{v}' , созданная i -м лучом.

В случае моделирования в RGB пространстве цвета поверхностей ($BPDF(\vec{p}, \vec{v}, \vec{v}', RGB)$ и $\tau(\vec{p}, \vec{v}, RGB)$) приводятся к одним условиям освещения, как правило, к белому источнику D65. Данное упрощение приводит к ошибке цветопередачи, вызванной неоднозначной зависимостью спектрального состава излучения от RGB цвета. На примере яркости света, рассеянного поверхностью, неоднозначность выглядит следующим образом:

$$BPDF(\vec{p}, \vec{v}, \vec{v}', RGB) E_i(\vec{p}, \vec{v}', RGB) \neq RGB[BPDF(\vec{p}, \vec{v}, \vec{v}', \lambda) E_i(\vec{p}, \vec{v}', \lambda)] \quad (5)$$

То есть использование RGB пространства для моделирования не гарантирует правильной цветопередачи. Следующие два примера визуально демонстрируют искажение цвета, вызванное вычислениями в RGB пространстве. На данных

простых примерах результаты рендеринга в RGB пространстве сравниваются с результатами спектрального рендеринга.

В первом примере красный источник света, заданный в видимой области спектра (спектрограмма источника света показана на рисунке 1а), освещает сферическую зеленую поверхность (спектральный коэффициент отражения показан на рисунке 1б). RGB цвет зеленой поверхности рассчитан при условии освещения поверхности белым источником D65.

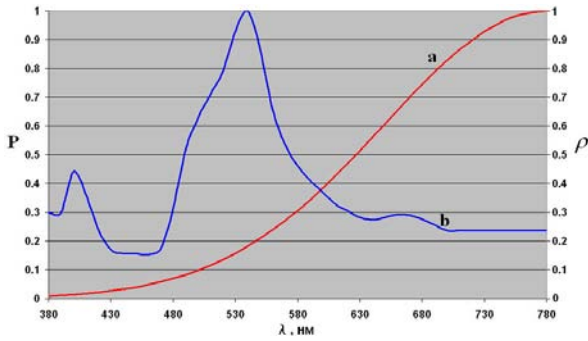


Рисунок 1: (а) Спектрограмма источника света, (б) спектральный коэффициент отражения поверхности.

Изображение сферы, освещенной красным источником света, показано на рисунке 2. Поскольку изображение зеркально симметрично, то для удобства сравнения результатов левая половина изображения была рассчитана в RGB пространстве, а правая половина – в спектральном пространстве.

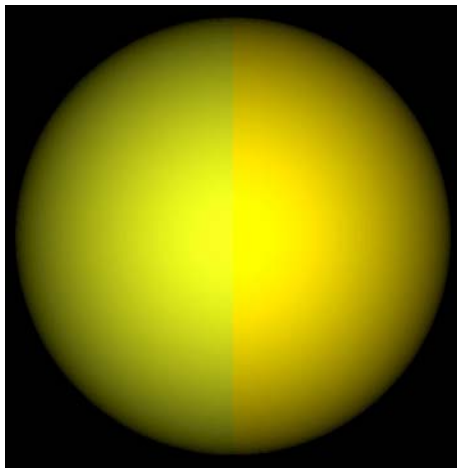


Рисунок 2: Изображения сферы. Левая половина в RGB пространстве, правая половина в спектральном пространстве.

В данном примере разница цветов обусловлена тем, что цвет источника света в сцене отличается от опорного цвета, используемого для вычисления цвета поверхности. Однако, разница цветов может возникнуть при освещении источником света того же цвета. На следующем примере белый свет (спектрограмма источника света показана на рисунке 3а) освещает зеленую сферическую поверхность (спектральный коэффициент отражения показан на рисунке 3б).

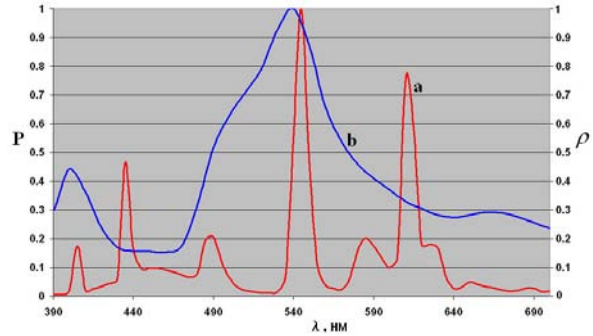


Рисунок 3: (а) Спектрограмма источника света, (б) спектральный коэффициент отражения поверхности.

Поскольку спектрограмма белого света отличается от базового источника D65, хотя их RGB практически совпадают, результирующие цвета в RGB и спектральной модели отличаются друг от друга. Аналогично рисунку 2 левая половина изображения на рисунке 4 была рассчитана в RGB пространстве, а правая половина – в спектральном пространстве.

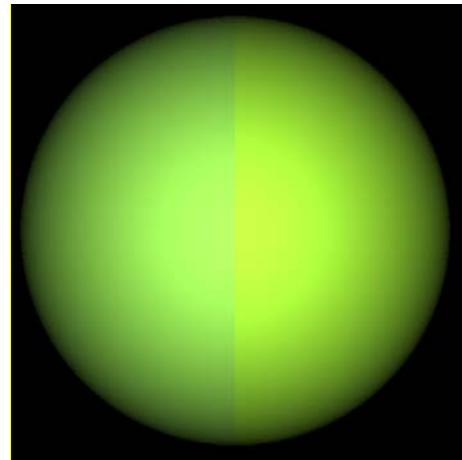


Рисунок 4: Изображения сферы. Левая половина в RGB пространстве, правая половина в спектральном пространстве.

Решение уравнения рендеринга в RGB пространстве имеет принципиальное ограничение. Этим ограничением является невозможность моделирования дисперсионных эффектов, то есть зависимости направления распространения света от длины волны света. В случае дисперсионного рассеивания яркость в выбранном направлении равна нулю, поскольку рассеивание осуществляется в бесконечно узком спектральном интервале. Поэтому в случае дисперсии имеет смысл говорить о спектральном коэффициенте пропускания (отражения) для заданной длины волны $\tau(\vec{p}, \vec{v}, \lambda)$, то есть об отношении плотности потока прошедшего (отраженного) излучения к плотности потока падающего излучения.

Физически аккуратное построение изображений сцен, содержащих диспергирующие элементы, возможно, если при моделировании спектральная область разбивается на большое количество узких спектральных областей. Рисунок 5 демонстрирует изображение дисперсионного пятна рассеивания, созданного стеклянной призмой, освещенной белым коническим источником D65.

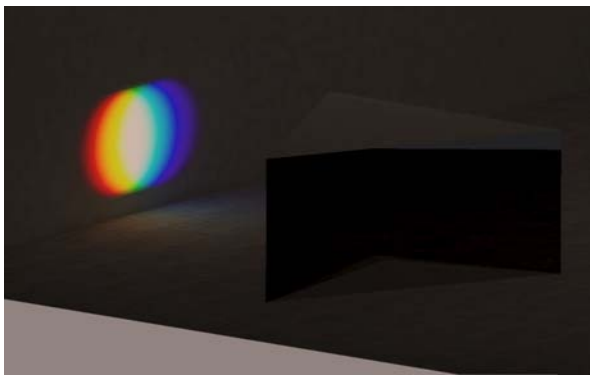


Рисунок 5: Изображение дисперсионного пятна, сформированного стеклянной призмой. Моделирование производилось в видимом диапазоне на 81 длине волны.

Аналогичный подход мог бы быть применен и в RGB пространстве, например, каждому цветному каналу сопоставить свою длину волны ($R \approx 620$ нм, $G \approx 540$ нм, $B \approx 460$ нм). Однако малое количество цветовых каналов (R, G и B) приводит к дискретности изображения, в результате чего оно выглядит неестественным. Рисунок 6 демонстрирует аналогичную модель дисперсионного рассеивания стеклянной призмы, выполненную в RGB пространстве.

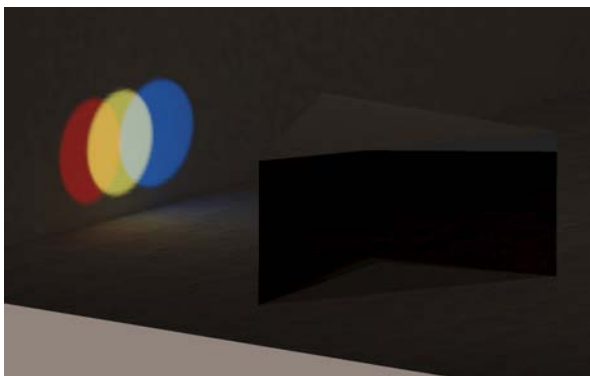


Рисунок 6: Изображение дисперсионного пятна, сформированного стеклянной призмой. Моделирование производилось в RGB пространстве.

4. Решение уравнения рендеринга в спектральном пространстве

Физически аккуратное решение уравнения рендеринга может быть найдено только в спектральном пространстве. Технически возможны три основных способа решения уравнения рендеринга (3). Во-первых, трассировка лучей может быть выполнена для ряда «монохроматических» сцен, оптические свойства каждой из которых определены для одной длины волны. Финальный результат – это сумма отдельных «монохроматических» расчетов. Во-вторых, для сцены, оптические свойства которой определены в широкой области спектра, можно трассировать отдельные «монохроматические» лучи, то есть лучи, чьи свойства определены для одной длины волны. Естественно, «монохроматические» лучи должны покрывать весь спектральный интервал и вносить соответствующую часть спектральной энергии в финальный результат. В-третьих, для сцены, определенной в широкой области спектра, возможна трассировка «полихроматических» лучей, то есть лучей, чьи

спектральные свойства определены во всем спектральном интервале. Финальный результат в данном случае – это есть собственно результат «полихроматического» расчета.

Рассмотрим более подробно преимущества и недостатки каждого из решений. Технически, первое решение выглядит наиболее простым. Формируется N независимых сцен (по числу длин волн, используемых в вычислениях) с оптическими свойствами, определенными для выбранных длин волн. Оптические свойства источников света задаются в виде спектрального распределения плотности потока излучения. Такой вид задания наиболее удобен для любого из видов спектрального моделирования, поскольку исключает ширину спектрального интервала, и спектральные данные могут быть применены к любому набору длин волн. В результате моделирования формируется N «монохроматических» изображений, которые могут быть объединены в изображение, содержащее спектральную плотность яркости для данного набора длин волн. Применяя выражения (1), формируется изображение в RGB пространстве, которое не содержит ошибок преобразования цвета в процессе трассировки лучей. Данное («монохроматическое») решение было реализовано как расширение программного комплекса SPECTER [10], работавшего до сих пор исключительно в RGB пространстве. Для автоматического формирования спектральных сцен был разработан генератор сцен, преобразующий набор спектральных данных в соответствующий набор независимых сцен. По окончании расчета отдельные изображения складывались в единое RGB изображение.

Данное решение оказалось недостаточно эффективным, поэтому единственным его преимуществом являлась простота его реализации (оно может быть применено практически к любому рендереру). Следующее возможное решение задачи спектрального моделирования – это трассировка «монохроматических» лучей в спектральной сцене. «Монохроматические» лучи испускаются для всего набора длин волн, определенных в сцене, однако в отличие от предыдущего способа спектрального моделирования, энергия лучей определяется не спектральной плотностью потока, а собственно спектральным потоком. Необходимость денормализации от плотности потока к собственно величине потока объясняется возможной непостоянностью интервала между соседними длинами волн, заданных в сцене. Выражение (6) демонстрирует простейший способ денормализации спектральной плотности потока.

$$P(\lambda_i) = \varphi(\lambda_i) \frac{\lambda_{i+1} - \lambda_{i-1}}{2} \quad (6)$$

где:

$$\varphi(\lambda_i) - \text{спектральная плотность потока } P(\lambda_i).$$

По окончании расчета поток, переносимый лучом, обратно нормируется, что позволяет получить результат в виде спектральной плотности заданной световой величины и применить выражения (1) для представления результата в RGB пространстве. В зависимости от способа спектрального моделирования (детерминистический или стохастический, прямой или обратный) луч может переносить как собственно спектральную энергию, так и единичную энергию, при этом вероятность его испускания будет пропорциональна энергии в спектральном интервале.

Очевидно, единственное преимущество данного решения – это удобство работы. Все спектральные данные инкапсулированы в одной сцене, и изображение строится в течение одного спектрального расчета. Однако эффективность спектральных вычислений не превышает эффективности предыдущего решения.

В данной работе было разработано эффективное решение для спектральной трассировки лучей в сценах, определенных во всей спектральной области. Это решение предполагает трассировку «полихроматического» луча. Идея метода заключается в следующем. Источник света или камера испускает луч, который определен во всем спектральном интервале. Данный луч распространяется в сцене, и на всех не диспергирующих поверхностях преобразование луча осуществляется для всего спектрального диапазона. Данное преобразование аналогично цветному преобразованию в RGB пространстве, когда вместо R, G и B компонент выступает денормализованная спектральная плотность. В случае детерминистического преобразования лучей (расчет видимой яркости, например) – это простое произведение денормализованных спектральных световых характеристик луча на соответствующие значения спектральных коэффициентов поверхности.

В случае стохастической прямой трассировки луча ситуация сложнее. Во-первых, вероятностным образом, в соответствии со спектральными мощностями K источников света, выбирается источник i , который испускает луч:

$$\frac{\sum_{j=1}^{i-1} P_j(\lambda)}{\sum_{j=0}^K P_j(\lambda)} \leq \xi < \frac{\sum_{j=1}^i P_j(\lambda)}{\sum_{j=0}^K P_j(\lambda)} \quad (7)$$

где:

ξ – случайное число, равномерно распределенное между 0 и 1.

Аналогичным вероятностным образом выбираются точка начала, направление и спектральный состав луча. Затем спектральная энергия луча нормируется к единичной энергии:

$$p(\lambda_n) = \frac{P_i(\lambda_n)}{\sum_{i=1}^N P_i(\lambda_i)} \quad (8)$$

При отсутствии дисперсии, луч сохраняет весь спектральный диапазон длин волн при любой трансформации на объекте. В этом случае спектральное преобразование аналогично преобразованию луча в RGB пространстве. То есть в соответствии со спектральными характеристиками рассеивающего объекта и спектрограммой луча выбирается направление нового луча и его спектральный состав. Спектральная мощность луча нормируется к единичной мощности (8).

Более интересен случай спектральной трансформации луча на диспергирующей поверхности. Для данных поверхностей не существует направления рассеивания, в котором представлен весь спектральный состав луча. Поэтому единственное решение – это преобразование «полихроматического» луча в «монохроматические» лучи. В случае детерминистической трассировки в точке дисперсии формируется пучок «монохроматических» лучей, количество которых равно количеству длин волн в спектральном составе луча. Каждый

луч имеет собственное направление и рекурсивно трассируется из данной точки. По окончании рекурсивной трассировки последнего луча, в точке дисперсии собирается полный спектральный состав луча.

Дисперсия в случае стохастической трассировки обрабатывается вероятностным образом. При использовании алгоритма «русской рулетки» трассируется только один луч и не допускается никакой рекурсии. Поэтому из «полихроматического» луча, представляющего широкий спектр, с вероятностью, пропорциональной денормализованной энергии луча на длинах волн, выбирается один n -ый «монохроматический» луч, который продолжает стохастическое распространение в сцене:

$$\sum_{l=1}^{n-1} p(\lambda_l) \leq \xi < \sum_{l=1}^n p(\lambda_l) \quad (9)$$

В случае использования метода «русской рулетки» энергия луча (как «полихроматического», так и «монохроматического») нормируется к единице. Поэтому энергия, переносимая единичным лучом, не несет достоверной информации о спектральном составе излучения. Истинный спектральный состав формируется на приемниках излучения как вероятность выбора луча определенной длины волны. Нормализация спектральных характеристик к плотности этих характеристик (6) производится на приемниках излучения.

Эффективность приведенного решения для сцен со сложной геометрией очевидна. В сложных сценах основная часть времени (до 90 %) расходуется на трассировку луча, а именно поиск точки пересечения луча с геометрическими объектами сцены. Поэтому, хотя преобразования в широкой области спектра значительно замедляют эффективность рендеринга, общая эффективность остается значительно выше, чем эффективность рендеринга, основанная на независимой трассировке лучей, определенных на одной длине волны. Дисперсионные эффекты, разбивающие луч, определенный в широкой области спектра, на индивидуальные «монохроматические» лучи, как правило, незначительно снижают общую эффективность работы программы. Во-первых, для большинства сцен доля диспергирующих элементов незначительна в общем объеме сцены, а, во-вторых, вероятность возникновения дисперсии на первом же элементе трассы луча тоже недостаточно велика.

Данное решение было реализовано в программном комплексе INSPIRER2 для режимов прямой стохастической и обратной детерминистической трассировки лучей. Это позволило осуществлять физически аккуратный и эффективный рендеринг спектральных сцен с учетом непрямого освещения.

В рамках данной работы был реализован метод спектрального моделирования, основанный на трассировке «полихроматического» луча в спектральной сцене (расширение программного комплекса INSPIRER2). Было произведено сравнение разработанного метода с методом, основанным на расчете набора независимых «монохроматических» сцен (используемым в программном комплексе SPECTER).

Изображения, формируемые этими методами, неотличимы, однако эффективность их вычислений различна. Ниже показаны два типичных изображения спектральных сцен. Рисунок 7 демонстрирует сцену, в которой отсутствуют дисперсионные эффекты, а рисунок 8 – сцену, содержащую

дисперсионные элементы (как правило, дисперсионные элементы занимают меньшую часть объема сцены). Спектральные расчеты проводились на 41 длине волны.



Рисунок 7: Изображение сцены со спектральными свойствами в отсутствии дисперсии.



Рисунок 8: Изображение сцены, содержащей диспергирующие оптические элементы.

Таблица 1 содержит значения времени рендеринга, включающего как прямую, так и обратную трассировку лучей, выполненного на компьютере Intel Core2 Duo T9400 2.53 GHz.

Сцена	Рисунок 7	Рисунок 8
«Монохроматическая» трассировка отдельных сцен	52 .3мин.	19.4 мин.
«Полихроматическая» трассировка лучей в единой спектральной сцене	16.9 мин.	6.8 мин.

Видно, что решение, используемое для «полихроматической» трассировки лучей, значительно эффективнее, чем выполнение расчетов на отдельных длинах волн. Кроме того, замедление, вызванное эффектом дисперсии, незначительно влияет на общее время рендеринга.

5. Взаимодействие спектрального луча с RGB объектами сцены

В большинстве случаев при работе со сценами, содержащими сложные объекты окружения, такие как обычные или HDR1 текстуры, не удается полностью определить оптические свойства сцены в спектральном представлении. В данной ситуации возникает вопрос об оптимальном решении для аккуратного рендеринга, хотя говорить о полной физической аккуратности невозможно.

Существуют два основных решения. Во-первых, одновременная трассировка в двух пространствах (спектральном и RGB). То есть, все, что можно трассировать в спектральном пространстве, трассируется в спектральном пространстве. Все остальное трассируется в RGB пространстве. В результате формируются два изображения – спектральное и RGB. Финальное изображение в RGB пространстве – это сумма двух изображений. Второе решение – это перевод всех оптических свойств, определенных в RGB пространстве, в спектральное представление и выполнение трассировки лучей исключительно в однородном спектральном пространстве.

В данной задаче нами было выбрано второе решение. Основные причины: во-первых, оно технически проще, а во-вторых, оно несколько точнее, поскольку исключает промежуточное клипирование цвета, которое может возникнуть при переходе от спектрального к RGB пространству.

Очевидно (1) обратное преобразование из RGB в спектр неоднзначно, и существует бесконечное количество спектральных кривых, которые могут сформировать один и тот же RGB цвет. Поэтому основными критериями трансформации RGB данных в спектральные данные являются максимальная равномерность спектральных данных и, для оптических свойств поверхностей, не превышение единичного коэффициента спектрального отражения (пропускания).

Для преобразования RGB данных в спектральное представление предлагалось большое количество решений. Например, [11] предлагает формировать спектральную кривую как сумму спектральных кривых основных цветов (красного, зеленого, голубого, желтого, цианового и магенты). Хотя спектральная кривая получается достаточно равномерной, недостатком этого и ряда других методов являлась неточность преобразования цвета поверхностей, связанная с выполнением условия максимально допустимого спектрального коэффициента отражения (пропускания).

В данной работе было предложено эффективное и аккуратное решение, позволяющее формировать спектральные кривые по RGB данным. Поскольку преобразование спектральных данных в RGB представление линейно, то спектральное распределение можно выразить следующим образом:

$$P(\lambda_i) = w_i^R \cdot r + w_i^G \cdot g + w_i^B \cdot b \quad (10)$$

где коэффициенты w_i^R, w_i^G, w_i^B определяются из условий минимальной вариации спектральной кривой и обратимости.

При преобразовании RGB данных в спектральное представление используются два основных решения. Для простых данных, как то коэффициенты отражения или пропускания, конвертация выполняется в процессе загрузки

сцены. Поэтому рендеринг работает исключительно со спектральными данными. Для сложных объектов, как то текстура или HDRI источник излучения, преобразование в спектральное представление откладывается на момент трассировки луча. Поэтому при попадании луча на такой объект RGB данные конвертируются в спектральное представление. Поскольку выражение (10) не является трудоемким, никакого замедления практически не происходит. Кроме того, отложенная конвертация цвета позволяет существенно уменьшить требования к требуемому высококачественной памяти, что важно для сцен, содержащих высококачественные текстуры или HDRI источники.

В качестве примера рендеринга, выполненного в пространстве, содержащем смесь RGB и спектральных данных, рисунок 9 демонстрирует изображение автомобиля, вычисленное в спектральном пространстве на 41 длине волны. Автомобиль покрыт краской, для которой была определена двунаправленная спектральная функция отражения. Автомобиль освещался источником света, определенным в виде HDRI в RGB пространстве.



Рисунок 9: Пример рендеринга, выполненного в пространстве, содержащем смесь RGB и спектральных данных.

6. Заключение

Искажение цветопередачи делает в большинстве случаев невозможным использования RGB пространства для задач построения фотореалистичных изображений. Поэтому единственное возможное решение для обеспечения физически аккуратного рендеринга – это спектральная трассировка лучей.

В рамках программного комплекса INSPIRER2 был реализован эффективный и физически аккуратный алгоритм спектральной трассировки обратного детерминистического и прямого стохастического лучей, что позволило выполнять спектральный рендеринг в сценах с многократным диффузным рассеиванием и дисперсией. Кроме того, были реализованы алгоритмы эффективного и аккуратного преобразования оптических свойств сцены, определенных в RGB пространстве, в однородное спектральное представление. Это позволило осуществлять спектральный рендеринг в сценах со сложным окружением, например, освещенных HDRI источником света.

7. Благодарности

Работа поддержана грантами Президента РФ НШ-8129.2010.9, РФФИ № 10-01-00302, а также компанией Integra Inc.

Авторы выражают свою признательность В.Г. Соколову и А.А. Гарбуло за предоставленные примеры.

8. Abstract

The article is devoted to analysis of the color reproduction inaccuracies which appear in the process of rendering of the scenes with the complex optical properties. Sources of the color reproduction inaccuracies in RGB space were discovered and analyzed, efficient and accurate spectral simulation solutions were proposed. The proposed solutions were implemented for renderer based on deterministic backward ray tracer and forward Monte-Carlo ray tracing techniques. Accurate and efficient solutions for reduction of scenes with mixture of optical properties defined in different color spaces (including RGB one) to the uniform spectral space were developed.

Keywords: Color reproduction, RGB, XYZ, rendering, spectral optical properties, spectral density, spectral ray tracing, photorealism.

9. Литература

- [1] Mark J. Kilgard, Kurt Akeley. Modern OpenGL: its design and evolution / ACM SIGGRAPH Asia '08 courses.
- [2] Henrik Wann Jensen, Per Christensen. High quality rendering using ray tracing and photon mapping / ACM SIGGRAPH '07 course notes.
- [3] Matt Pharr, Greg Humphreys. Physically Based Rendering - From Theory to Implementation // Morgan Kaufmann, 2004.
- [4] Erik Reinhard, Greg Ward, Garrett Johnson. Color imaging / ACM SIGGRAPH '09 course notes.
- [5] Reinhard E., Stark M., Shirley P., Ferwerda J. Photographic Tone Reproduction for Digital Images / SIGGRAPH '02 conf. proc.
- [6] Д.С. Волосов, М.В. Цивкин. Теория и расчет светооптических систем. // «Искусство», Москва – 1960.
- [7] Gaurav Sharma. Digital Color Imaging Handbook // - CRC Press, 2003.
- [8] Gaurav Sharma, Wencheng Wu, Edul N. Dalal. The CIEDE2000 Color-Difference Formula: Implementation Notes, Supplementary Test Data, and Mathematical Observations / ECE Dept, Univ. of Rochester, Rochester, NY 14627-0126, USA; Xerox Corporation, 800 Phillips Road, Webster, NY 14580.
- [9] Kajiya, J. T. The rendering equation / Siggraph 1986: 143.
- [10] <http://www.integra.jp/en/index.html>
- [11] Brian Smits. An RGB to Spectrum Conversion for Reflectances / University of Utah, January 21, 2000.
- [12] Hoffmann, Gernot (2000), CIE Color Space [Online]: www.fho-empden.de/~hoffmann/ciexyz29082000.pdf

Моделирование распространения света в тонком красящем слое с высокой концентрацией частиц

А.Г. Волобой*, С.В. Ершов*, Э.С. Клышинский*, С.Г. Поздняков**

*Институт прикладной математики им. М.В.Келдыша РАН, Москва

**Национальный исследовательский ядерный университет «МИФИ», Москва

Аннотация

В работе описан подход к моделированию распространения света в тонком красящем слое. Была поставлена задача генерации изображений слоя принтерных чернил, исходя из внутренней структуры слоя и формы красящих частиц. В отличие от красок слой чернил характеризуется высокими объемными концентрациями частиц. Лучевой подход или уравнение переноса излучения практически не применимы для решения этой задачи. Поэтому был выбран путь непосредственного решения волновых уравнений, естественно, при некоторых ограничениях и упрощениях. В статье описаны постановка задачи, проблема генерации явной геометрии красящего слоя для высоких концентраций частиц, моделирование слоя чернил на бумаге. В конце статьи приведены выводы, сделанные на основе результатов моделирования.

Ключевые слова: реалистичная визуализация материалов, взаимодействие света с красящим слоем, физически корректное моделирование, ДФО/ДФП, оптические свойства поверхности.

1. ВВЕДЕНИЕ

Среди работ, посвященных реалистичной компьютерной графике, выделяется целое направление, связанное с моделированием оптических свойств сложных материалов и покрытий поверхности. Разрабатываемые алгоритмы применяются для генерации реалистичных изображений материалов, состав и структура которых нам известны, однако их оптические характеристики не могут быть измерены непосредственно (например, когда таких материалов еще не существует). Если моделирование материалов базируется на законах взаимодействия света с композицией материала, и мы получаем правдоподобные результаты, то подобные алгоритмы могут быть использованы при компьютерном дизайне новых материалов.

Одни из первых разработок были проведены для акварельных красок [1]. Авторы применяют широко известную модель Кубелка-Мунка, которая является очень упрощенным решением уравнения переноса излучения (света) [2]. В настоящее время, с развитием компьютеров и математических методов стало возможным непосредственное расчет распространения света в красящих слоях с использованием уравнения переноса. Моделирование многослойных красок со сложной микроструктурой (типа перламутровых и «металлик»), предложенное в [3-4], является примером таких работ. Краска в них рассматривается состоящей из плоскопараллельных слоев прозрачного лака, содержащих также красящие пигменты, интерференционные или зеркальные частицы. Размеры частиц могут быть достаточно малы по сравнению с длиной

волны света, что требует применения алгоритмов волновой оптики (расчет фазовых функций), но сами частицы расположены достаточно далеко друг от друга, и распространение света в краске может быть рассчитано в рамках уравнения переноса излучения [2]. Результатом моделирования является двунаправленная функция отражения (ДФО) поверхности, покрытой данной краской. Полученные ДФО могут быть использованы при генерации реалистичных изображений автомобилей или других объектов.

Ткани являются сложным для визуализации и моделирования типом материала. В работе [5] расчет ДФО ткани основан на моделировании распространения света внутри нее. Освещенность ткани рассчитывается с учетом рассеяния света поверхностями и объемами нитей, ее образующих. Также принимается во внимание структура переплетения нитей, которая позволяет определить тени и свет, отбрасываемые одними нитями на другие. При расчете теней и объемного рассеяния света учитываются оптические характеристики материала, из которого состоит нить. Поверхностное рассеяние света нитью аппроксимируется по формуле Фонга. Так как все объекты моделирования имеют достаточно большие размеры применение волновой оптики при расчете оптических свойств ткани практически не требуется.

Также интересные работы были сделаны для развития средств визуализации драгоценных камней и кристаллов [6-8].

В представленной работе авторам требовалось разработать алгоритмы моделирования слоя принтерных чернил, используемых при печати фотографий. В этой задаче красящий слой состоит из малых частиц пигмента с размерами $\sim 100\div 500$ нанометров при объемной концентрации пигмента PVC (Pigment Volume Concentration), доходящей до 50%. Малые размеры частиц, высокая концентрация и малая толщина слоя (~ 1 микрона) не позволяют применять уравнение переноса излучения, как в случае с многослойными красками. Поэтому был выбран путь непосредственного решения волновых уравнений, естественно, при некоторых ограничениях и упрощениях.

2. УПРОЩЕНИЯ ВОЛНОВОЙ ЗАДАЧИ

Основное упрощение касается самого использованного волнового уравнения. Точные уравнения, описывающие распространение света, хорошо известны - это уравнения Максвелла [9]. При характерных размерах неоднородностей среды, многократно превышающих атомные размеры, возможно применение так называемых макроскопических уравнений Максвелла, когда свойства сред описываются соответствующими диэлектрической и магнитной проницаемостями. Как правило, в оптике, в частности в нашем случае, магнитная проницаемость равна 1, а

диэлектрическая проницаемость может быть комплексной величиной. Существует небольшое число точных аналитических решений макроскопических уравнений. Все они получены для простых пространственных геометрий с высокой симметрией. Тем не менее, даже в случае очень высокой симметрии, например, рассеяние плоской линейно поляризованной волны на диэлектрической сфере (теория Ми), конечный ответ представляется в виде бесконечного ряда, содержащего специальные функции математической физики. В нашем случае со сложной пространственной геометрией аналитическое решение невозможно и может быть получено только с помощью применения численной процедуры. И хотя, на наш взгляд, применение уравнений Максвелла все же предпочтительнее, нами в качестве волнового уравнения было выбрано так называемое скалярное приближение, уравнение Гельмгольца [9].

Скалярное приближение широко применяется в оптике при расчетах распространения неполяризованного излучения. Оно не учитывает поляризацию электромагнитного излучения, и приближенно можно считать, что оно соответствует электромагнитному излучению, поляризованному в плоскости раздела разных сред. В случае простых геометрий различие между результатами точных электромагнитных и скалярных расчетов может быть заметным. Однако в случае стохастических сред, когда сильное, в значительной степени изотропное, рассеяние приводит к деполаризации излучения, применение скалярного приближения вполне оправдано. Как правило, в таких случаях все существенные волновые эффекты скалярное приближение описывает достаточно хорошо.

Следует отметить одно важное преимущество скалярных расчетов перед точными электродинамическими. В отличие от уравнений Максвелла решение скалярного уравнения непрерывно вместе с его первыми частными производными. Эта непрерывность существенно снижает требование к размерам используемой пространственной сетки и времени вычислений. Кроме этого, геометрия рассеивающего объекта задается в численной процедуре просто через значения диэлектрической проницаемости в узлах сетки. Скалярное приближение оперирует всего с одним комплексным полем вместо 5-6 полей в электромагнитном случае, что позволяет проводить вычисления на отдельных компьютерах с вполне доступным размером оперативной памяти (8Гб и менее), совсем не используя параллельные вычисления.

Важное упрощение задачи касается ее геометрии. В реальности освещение всегда происходит некогерентным светом (случай лазерного освещения здесь не рассматривается), поэтому и некогерентность освещения и стохастичность геометрии в результате дают усредненную картину угловых распределений. При освещении слоя когерентным светом усреднение может быть достигнуто только за счет стохастичности структуры красящего слоя. Для этого расчет должен использовать очень большой участок поверхности слоя, что не оправдано с точки зрения объема оперативной памяти и времени расчета.

Вместо использования больших фрагментов был использован другой подход. Рассеяние вычислялось для бесконечного плоского красящего слоя, составленного из одного элементарного фрагмента размерами $\sim 10 \times 10$ микрон, а затем периодически повторенного в двух измерениях. Использование периодических структур – это прием,

достаточно часто применяемый в оптических расчетах [2]. В таком случае угловые распределения рассеянного света в дальней зоне содержат конечное дискретное число направлений. Интенсивность луча (мощность, переносимая в данном направлении) является случайной величиной и в достаточной степени зависит от геометрии элементарного фрагмента. Чтобы получить гладкие угловые распределения, необходимо произвести расчеты для различных (случайных, но подчиняющихся одной и той же статистике) реализаций элементарного фрагмента, а затем усреднить результаты по ансамблю таких реализаций. Нетрудно видеть, что подобная процедура эквивалентна усреднению по ансамблю, применяемому в статистической физике, и достаточно точно отражает реальную картину рассеяния на красящем слое.

В компьютерной графике свойства поверхностей описываются с помощью соответствующих угловых распределений интенсивности или яркости отраженного света – двунаправленной функцией отражения (ДФО). В данном случае расчеты проводятся для монохроматического освещения. Поскольку излучение с разными длинами волн взаимно некогерентно, то построение спектральных ДФО никаких трудностей не вызывает. При расчетах для периодического слоя угловые распределения имеют дискретную угловую структуру. Полное количество направлений и их плотность определяются размерами элементарного фрагмента. Используемые размеры позволяют получить гладкие угловые распределения для всех существенных направлений. Более того, поскольку согласно постановке нашей задачи красящий слой располагался на подложке (бумаге) с Ламбертовским отражением, задача получения ДФО для системы слой + бумага имеет даже более простое решение.

Уже упоминалось, что красящий слой, как правило, имеет стохастическую структуру, в которой объемная концентрация зерен пигмента (распределенных случайно по объему слоя) может составлять 50% и более. Экспериментальное измерение реальной геометрии слоя практически невыполнимо. Поэтому возникает необходимость генерации такой геометрии с помощью численной процедуры. Для одинаковых шарообразных частиц и объемной концентрации $< 30\%$ генерация достаточно проста и может быть осуществлена случайным разбрасыванием центров частиц по объему элементарного фрагмента, исключая выход за поверхности фрагмента и пересечения между частицами.

В данной работе использовались частицы более сложной формы, приближенной к реальным формам красящих пигментов в чернилах цветных принтеров. Аксиально-симметричная форма таких частиц может быть приближена с помощью нескольких жестко связанных шаров различного радиуса с возможными пересечениями. Зерна пигмента не должны выходить за поверхности элементарного фрагмента и пересекаться между собой. Кроме этого, предполагалось, что протяженные частицы в красящем слое, могут быть полностью или частично ориентированы. Для генерации геометрии с подобными свойствами была применена численная процедура, аналогичная процедуре “молекулярной динамики”. Необходимая ориентация частиц достигалась за счет наложения “магнитного” поля той или иной напряженности.

Таким образом, вся задача моделирования свелась к решению трех самостоятельных задач:

1. Генерация геометрии фрагмента красящего слоя, состоящего из частиц заданных форм и размеров и вычисление распределения коэффициента преломления на используемой пространственной сетке.
2. Решение задачи дифракции на тонком периодическом красящем слое с явным распределением коэффициента преломления конечно-разностными методами.
3. Вычисление ДФО красящего слоя, расположенного на Ламбертовской подложке исходя из результатов решения задачи дифракции.

3. ГЕНЕРАЦИЯ ГЕОМЕТРИИ КРАСЯЩЕГО СЛОЯ

Дифракционный расчет рассеяния света красящим слоем предполагает задание геометрии такого слоя в явном виде. Мы проанализировали структуры реальных красящих слоев, состоящих из прозрачного связующего вещества, наполненного зернами пигмента различной формы. Зерна пигментов в объеме слоя, как правило, располагаются однородно и хаотически. Однако возможны случаи, когда зерна имеют преимущественную ориентацию или образуют конгломераты (агломинация).

Поверхность красящего слоя может быть плоской, так как при высыхании чернил действуют силы поверхностного натяжения связующего вещества. При значительных концентрациях пигмента сил поверхностного натяжения недостаточно для формирования плоской поверхности и часть частиц “выступает” из слоя, образуя поверхность с микрорельефом. Наличие микрорельефа приводит к возникновению дополнительного рассеяния.

Генерация геометрии для небольших концентраций пигментных зерен достаточно проста. Она может быть осуществлена последовательным случайным разбрасыванием центров и ориентаций частиц по объему фрагмента слоя. При этом если новая добавляемая к ансамблю частица пересекается с уже существующими частицами или выходит за границы фрагмента, то такая частица отбрасывается, и происходит случайное разыгрывание следующей частицы. Процедура заканчивается, когда фрагмент слоя наполняется необходимым числом частиц.

В случае очень тонких красящих слоев (толщина ~ 1 микрона) объемная концентрация зерен пигмента может достигать значительной величины $PVC \sim 30\%$ и более. При таких концентрациях случайное разбрасывание не дает необходимого результата и следует применять более сложную процедуру.

Использованный подход можно описать следующим образом. Сначала в расчетной области случайно разбрасывается необходимое число частиц, которые могут пересекаться между собой. Затем между частицами вводится упругое взаимодействие, в нашем случае – отталкивание, которое приводит к возникновению сил между пересекающимися частицами или при пересечении частицей границы области. Под действием сил частицы могут перемещаться (и вращаться), и таким образом пересечения между частицами и частиц с границами устраняются. Когда все нежелательные пересечения устранены процедура заканчивается.

Перемещение частиц под воздействием сил можно находить, например, методами спуска, минимизируя энергию взаимодействия всего ансамбля. В данной работе мы

использовали динамику первого порядка. Это означает, что частицы не обладают инерцией, и движение частицы мгновенно прекращается как только исчезает действующая на нее сила. В таком случае нет необходимости вводить массы частиц. Нетрудно видеть, что настоящий подход практически эквивалентен минимизации энергии ансамбля взаимодействующих частиц методом наискорейшего спуска.

После того как геометрия красящего слоя получена, вычисление распределения коэффициента преломления на пространственной сетке (исходные данные для задачи дифракции) не вызывает затруднений. Отметим, что в нашем случае, красящий слой может состоять из зерен различных пигментов различной формы.

Форма пигментных частиц.

Анализ красящих слоев показал, что с достаточной точностью в качестве пигментных зерен можно использовать аксиально-симметричные частицы. Такие частицы могут быть аппроксимированы некоторым количеством жестко соединенных между собой шаров. Например, цилиндрическая частица может быть аппроксимирована несколькими одинаковыми шарами с центрами, расположенными на оси цилиндра. Точность аппроксимации будет зависеть от количества шаров и расстояний между их центрами. Нетрудно видеть, что используемая аппроксимация частиц шарами позволяет конструировать частицы самой разнообразной формы. Естественно, время расчета геометрии сильно зависит как от числа частиц, так и от сложности этих частиц.

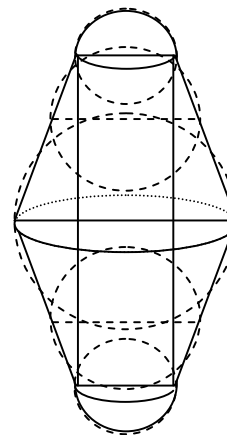


Рис. 1. Частица “рисовое зерно”.

Наш анализ показал, что во многих случаях форма частиц пигмента близка к форме рисового зерна. Поэтому, наряду с возможностью использовать частицы различной формы, мы для удобства расчетов, также использовали частицу именно такой формы (размеры частицы могут быть переменными). Схема частицы в форме рисового зерна приведена на рис. 1.

Взаимодействие между частицами.

Сила взаимодействия между пересекающимися частицами вычисляется как сумма всех сил для пересекающихся шаров, из которых частицы состоят (естественно, шары, составляющие одну частицу, не взаимодействуют между собой). Сила взаимодействия между пересекающимися шарами зависит от величины этого пересечения L (рис. 2):

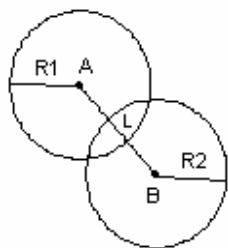


Рис. 2. Пересечение шаров. $L=R1+R2-AB$.

Конкретный вид силы взаимодействия может быть выбран в достаточно широком диапазоне. Как правило, сила взаимодействия максимальна при совмещении центров шаров и становится равной нулю для $L < 0$. В том случае, когда данный шар пересекается с несколькими шарами, действующая на него сила определяется как векторная сумма всех сил.

Если частица состоит более чем из одного шара, кроме перемещения необходимо определить возможные вращения оси симметрии частицы. При этом для каждой протяженной частицы определяется ее “центр масс” (например, геометрический центр частицы), и вычисляется действующий на частицу вращающий момент:

$$M = \sum_i [r_i \times F_i],$$

где r_i – вектор, соединяющий центр частицы и центр ее i -го шара, F_i – сила, действующая на центр i -го шара.

Движение частиц.

Точные уравнения, описывающие движение i -ой частицы имеют следующий вид:

$$\begin{aligned} \mu_i v_i &= F_i \\ J_i \omega_i &= M_i \end{aligned}$$

где v_i – скорость центра частицы, ω_i – вектор угловой скорости вращения оси симметрии частицы.

Величины μ и J могут быть названы коэффициентами трансляционной и вращательной вязкости. В данной задаче нет необходимости интегрировать уравнения движения с высокой точностью, поскольку конечным результатом является ансамбль непересекающихся частиц, а не точные траектории движения. Поэтому возможно применение простейшего метода интегрирования дифференциальных уравнений – метода Эйлера [10].

В процессе интегрирования уравнений движения очень важную роль играет выбор шага интегрирования. Малый шаг интегрирования обеспечивает достаточную точность вычислений, но увеличивает длительность расчетов. Слишком большой шаг интегрирования приводит к заметным ошибкам, в результате ухудшается сходимость решения, возможен переход в режим осциллирующего поведения, и даже потеря сходимости.

Обычный способ состоит в том, чтобы выбрать величину шага интегрирования, для которой изменения правых частей динамических уравнений будут невелики. В нашем случае, когда силы взаимодействия зависят от расстояний между

отталкивающимися шарами, такое условие сводится к малости перемещений шаров по сравнению с их радиусами за один шаг интегрирования. В настоящей работе выбор оптимального шага был произведен экспериментально, учитывая результаты достаточно большого числа пробных расчетов. Эти расчеты показали, что следует использовать шаг интегрирования, при котором максимальные смещения не превосходят ~ 0.2 от характерного радиуса шаров, составляющих частицы пигмента.

Описание алгоритма генерации геометрии слоя.

Вычисление геометрии слоя состоит из нескольких последовательных итераций (число итераций определяется заранее). Каждая итерация стартует с ансамбля, состоящего из некоторого числа непересекающихся частиц, находящихся в неподвижном состоянии (в самом начале расчета частиц нет совсем). Итерации состоят из следующих шагов:

1. Добавление к ансамблю новых частиц. Новые частицы распределены случайно (координаты центров и ориентация оси симметрии). Количество добавляемых частиц определено заранее и зависит от объемных концентраций частиц различных видов, используемых в вычислениях. Все вновь добавленные частицы записываются в текущий список движущихся частиц. Список движущихся частиц должен содержать частицы, которые могут пересекаться с другими частицами или с границами расчетной области. Поскольку перед началом каждой итерации все пересечения устранены, то в текущий список добавляются только новые частицы.
2. Вокселизация. Чтобы ускорить вычисления производится “вокселизация” расчетной области, т. е. вся расчетная область разбивается на одинаковые элементарные подобласти (воксели). Размеры вокселей выбираются большими, чем максимальный размер частицы. Вокселизация частиц состоит в определении соответствующего вокселя для каждой частицы. При рациональном выборе размеров вокселей возможно взаимодействие только между частицами, расположенными в соседних вокселях. Использование вокселизации позволяет существенно ускорить перебор возможных пар взаимодействующих частиц при вычислении сил и вращающих моментов.
3. Вычисление сил и вращающих моментов. Для каждой частицы из текущего списка движущихся частиц вычисляются силы и моменты, возникающие при возможном пересечении со всеми остальными частицами. Естественно, что для перебора всех взаимодействующих пар достаточно перебрать только частицы из текущего списка, т. к. частицы вне этого списка на предыдущем шаге ни с чем не взаимодействовали (покоились). Потенциально взаимодействующие частицы определяются, используя вокселизацию из п. 2. Силы и моменты, действующие на каждую взаимодействующую частицу, суммируются, а сама частица добавляется в новый список движущихся частиц. Перебор взаимодействующих пар выполнен так, чтобы избежать двойного учета взаимодействий.
4. Взаимодействие с границами области. Если частица пытается выйти за границу расчетной области, на нее начинает действовать упругая возвращающая сила, аналогичная силе между пересекающимися частицами. Возвращающая сила направлена по нормали к границе, ее

величина растет по мере удаления частицы от допустимого объема области. Вычисленные силы добавляются к силам, вычисленным в п. 3. Все частицы, вышедшие за границы, добавляются в новый список движущихся частиц.

5. Перемещения и вращения. С помощью вычисленных сил и вращающих моментов определяется шаг интегрирования. Затем происходит перемещение и вращение частиц.

6. Модификация текущего списка движущихся частиц. Вычисленный новый список движущихся частиц переписывается в текущий список. После этого новый список очищается.

7. Если пересечения не устранены и число шагов интегрирования не превышает заранее заданный лимит, то происходит переход к п. 2. В противном случае переход к п. 1.

Итерации прекращаются, когда достигается необходимая объемная концентрация пигментов.

Естественно, настоящий алгоритм имеет ограничения по объемной концентрации. Однако, для реальных концентраций пигмента, например для шарообразных частиц при объемной концентрации пигмента < 60%, результаты работы генератора геометрий вполне удовлетворительные.

Мы провели достаточно большое количество расчетов и вычислили необходимые статистические характеристики распределений частиц, полученных с помощью генератора геометрий. Все расчеты дали вполне удовлетворительные результаты, а статистические характеристики распределений находились в допустимых пределах.

4. РЕШЕНИЕ ЗАДАЧИ ДИФРАКЦИИ И ВЫЧИСЛЕНИЕ ДФО КРАСЯЩЕГО СЛОЯ

Рассматривается стационарный режим, когда красящий слой освещается падающей скалярной монохроматической волной. В этом случае рассматривается волновое уравнение Гельмгольца:

$$\Delta u + k^2(x)u = 0 \quad (1.1)$$

где $k(x) = \eta(x)k_0$ – есть локальное волновое число, k_0 – волновое число в вакууме и η – локальный показатель преломления. В нашем случае показатель преломления различен в вакууме, частицах пигмента и в объемлющей среде.

Формально необходимо проинтегрировать уравнение Гельмгольца (1.1) во всей области, включая красящий слой и вакуум по обе стороны от него. Естественно, прямой подход, который использует дискретизированное поле во всей расчетной области, невозможен. Вместо этого мы разделим все пространство на три части: вакуум выше красящего слоя, сам слой и вакуум ниже слоя. В полупространствах заполненных вакуумом выше и ниже слоя показатель преломления постоянен, а уравнение (1.1) имеет хорошо известные аналитические решения. Таким образом, нет необходимости оперировать с полем выше и ниже слоя на пространственной разностной сетке. Для поля внутри красящего слоя аналитическое решение не существует, поэтому решение находится с помощью численной процедуры.

Решение дифракционной задачи должно быть найдено в трех разных областях. На границах этих областей, т.е. на границах вакуум-красящий слой, должны быть заданы условия сшивки для искомого решения. Для уравнения (1.1) условия сопряжения имеют простой вид: поле u вместе с его нормальной производной должно быть непрерывно на всех границ:

$$\begin{aligned} u|_{z=0-0} &= u|_{z=0+0} \\ \frac{\partial u}{\partial z}|_{x=0-0} &= \frac{\partial u}{\partial z}|_{x=0+0} \\ u|_{z=H-0} &= u|_{z=H+0} \\ \frac{\partial u}{\partial z}|_{x=H-0} &= \frac{\partial u}{\partial z}|_{x=H+0} \end{aligned}$$

Условий сшивки недостаточно. Недостающие условия, называемые условиями излучения, как правило, задаются на бесконечном удалении от рассеивающего объекта. Эти условия утверждают:

- Решение ограничено на бесконечности.
- Все компоненты поля, соответствующие приходящим волнам (за исключением падающей волны в верхнем полупространстве) исчезают на бесконечности.

В нашем случае плоской периодической геометрии условия излучения задаются в явном виде непосредственно на границах красящий слой – вакуум.

Естественно, численная процедура не может использовать бесконечную расчетную область.

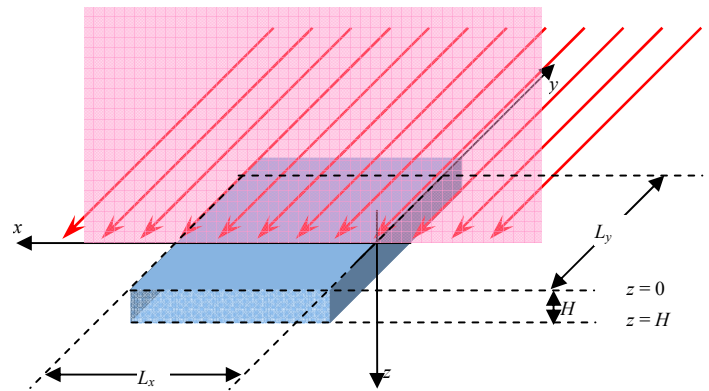


Рис. 3. Красящий слой представлен в виде пластины, параллельной плоскости Oxy . Падающая волна показана стрелками; направление падения лежит в плоскости Oxz .

Поэтому практический подход состоит в использовании бесконечного красящего слоя, состоящего из элементарного фрагмента (имеющего форму параллелепипеда), повторяющегося периодически в двух измерениях (рис. 3). Такой подход широко используется в дифракционных задачах [2]. В случае нормального падения (вдоль оси Oz) полное поле будет периодическим в двух измерениях: для всех элементарных фрагментов освещающие поля, поля внутри фрагментов и поля, рассеянные фрагментами, будут одинаковы. Отсюда следуют условия периодичности:

$$\begin{aligned} u(x + L_x, y, z) &= u(x, y, z) \\ u(x, y + L_y, z) &= u(x, y, z) \end{aligned} \quad (1.2)$$

В случае наклонного падения освещающее поле на поверхности красящего слоя имеет вид:

$$u_0 = e^{ik_0 x \sin \sigma}$$

где k_0 – волновое число в вакууме, σ – угол падения (угол между направлением освещения и нормалью к красящему слою). Поле для двух соседних периодов отличается множителем $e^{ik_0 L_x \sin \sigma}$. Если для некоторого натурального N выполняется следующее соотношение (здесь и далее λ – длина волны в вакууме)

$$\sigma = \arcsin\left(\frac{2\pi}{k_0 L_x} N\right) = \arcsin\left(\frac{\lambda}{L_x} N\right),$$

то поле будет одним и тем же для всех периодов, и мы опять можем применять периодические граничные условия (1.2). Для фрагмента с периодом порядка 100 длин волн “сетка” подобных направлений достаточна для задания всех необходимых направлений падения с приемлемой точностью.

Для других углов падения поле уже не будет периодическим, а граничные условия принимают вид:

$$u(x + L_x, y, z) = u(x, y, z)e^{ik_0 L_x \sin \sigma}$$

$$u(x, y + L_y, z) = u(x, y, z)$$

В таком случае решение дифракционной задачи удобно искать в представлении Блоха:

$$u(x, y, z) = v(x, y, z)e^{ik_0 L_x x \sin \sigma}$$

здесь v – периодическая функция x, y . Представление Блоха широко применяется в методах, использующих преобразование Фурье [2].

Итак, дифракционная задача сводится к уравнению Гельмгольца, которое решается раздельно для трех пространственных областей: заполненное вакуумом верхнее полупространство, красящий слой и нижнее полупространство также заполненное вакуумом. Периодичность позволяет применить дискретное преобразование Фурье по координатам x, y . Если отклонение коэффициента преломления от его среднего значения (внутри красящего слоя) невелико, то для нахождения Фурье гармоник поля $u_{mn}(z)$ можно применить метод теории возмущений, известный как Борновское разложение [11]. К сожалению, Борновское разложение не всегда сходится, поэтому был разработан метод решения уравнения с помощью счета на установление. Подробно математика решения задачи дифракции как с помощью Борновского разложения, так и с помощью оригинального метода счета на установление описана в [12].

Результатом решения дифракционной задачи для периодического красящего слоя являются значения волнового поля на поверхности слоя (комплексные амплитуды Фурье гармоник). Для вычисления двунаправленной функции отражения/преломления (ДФО/ДФП или BDF), описывающей диффузное рассеяние, эти амплитуды нужно преобразовать в угловое распределение рассеянного излучения в дальней зоне. Такое преобразование может быть выполнено для настоящей плоско-периодической структуры с использованием хорошо известной формулы

Кирхгофа [13]. Поскольку поле в дальней зоне представляет собой конечный набор плоских волн, можно непосредственно применить выражение для лучевой яркости [13].

Опуская вычисления, приведем только окончательное выражение для отражения.

$$G(\mathbf{v}) = k_0 \cos \theta \sum_{\text{Im} k_z = 0} |u_{mn}(0)|^2 \delta(\mathbf{v} - \mathbf{v}_{m,n}) \quad (1.3)$$

где $G(\mathbf{v})$ есть искомая плотность углового распределения интенсивности отраженного излучения, отнесенная к единичной площадке бесконечного красящего слоя. Дельта-функции возникают из-за периодичности (аналогично случаю бесконечной дифракционной решетки). Сумма $G(\mathbf{v})$ содержит только убегающие волны, зеркальной компоненте соответствует $m=n=0$. Здесь $u_{mn}(0)$ – комплексная амплитуда Фурье гармоники поля на освещенной поверхности слоя, $\mathbf{v} = (\sin\theta \cos\varphi, \sin\theta \sin\varphi, \cos\theta)$ и $\mathbf{v}_{m,n} = (\sin\theta_{m,n} \cos\varphi_{m,n}, \sin\theta_{m,n} \sin\varphi_{m,n}, \cos\theta_{m,n})$ есть единичные вектора в направлении наблюдения (θ, φ) и направлении распространения соответствующей плоской волны $\theta_{mn} = \arccos(k_z/k_0)$, $\varphi_{mn} = \arctan(k_y/k_x)$,

$k_x = \frac{2\pi m + q}{L}$, $k_y = \frac{2\pi n}{L}$, $k_z = \sqrt{k_0^2 - k_x^2 - k_y^2}$, $q = k_0 \sin \sigma$. Угловое распределение для прошедшего излучения аналогично (1.3) с заменой $u_{mn}(0)$ на $u_{mn}(H)$.

5. МОДЕЛИРОВАНИЕ ЧЕРНИЛ

В настоящей работе рассматривался тонкий красящий слой (чернила), нанесенный на специальную бумагу (рис. 4). Анализ рассеивающих свойств бумаги показал, что с достаточной точностью бумагу можно моделировать как плоскую поверхность, обладающую Ламбертовским отражением с интегральным коэффициентом отражения, близким к 1.

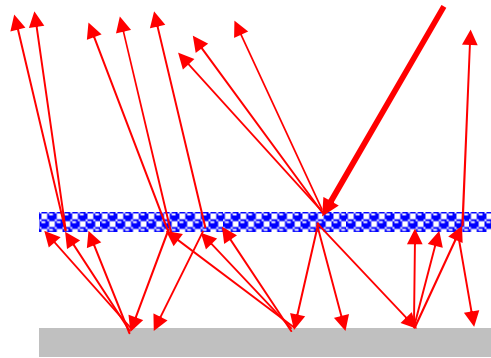


Рис. 4. Красящий слой, расположенный на диффузной подложке (бумаге) и переотражения света между слоем и подложкой. Для наглядности расстояние между слоем и подложкой преувеличено. Первичное освещение показано жирной стрелкой.

В нашем подходе и красящий слой и подложка рассматриваются как поверхности, однозначно характеризующиеся их BDF (ДФО + ДФП). Такое

рассмотрение является точным для статистически однородного слоя конечной толщины, и предполагает, что отражение от подложки полностью некогерентно.

Отметим, что падающий на бумагу свет состоит из двух компонент:

1. первичное освещение, прошедшее через красящий слой
2. свет, отраженный подложкой, а затем отраженный красящим слоем назад в сторону подложки.

При моделировании были использованы следующие входные данные:

- Оптические свойства подложки (бумага): идеальная белая Ламбертовская поверхность с интегральным коэффициентом диффузного отражения 0.9.
- Коэффициент преломления прозрачного связующего вещества слоя $n=1.558$.
- В расчетах были использованы три реальных пигмента Red, Yellow, Blue, а также искусственный пигмент Green. Действительная часть коэффициента преломления всех пигментов $Re(\eta) = 1.4$. Мнимая часть коэффициента преломления пигментов $Im(\eta)$ представлена на рис. 5.

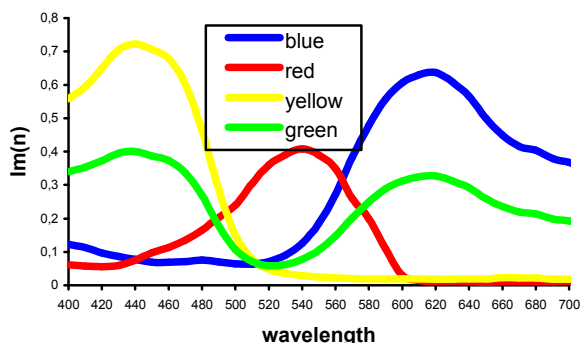


Рис. 5. Спектрограммы мнимых частей коэффициентов преломления использованных пигментов.

В расчетах были использованы различные объемные концентрации пигментов, а также смеси Yellow и Blue при полной концентрации PVC 15%, 30% и 50%.

Были использованы шарообразные и цилиндрические зерна пигментов. Диаметр шарообразных зерен изменялся от бесконечно малой величины (молекулярная смесь) до 450 nm. Длина цилиндрических частиц была равна 6 радиусам цилиндров. Размер цилиндрических частиц определялся с помощью “эффективного диаметра”, т.е. диаметра равновеликого шара. Были использованы цилиндры со следующими значениями эффективного диаметра: 100 nm, 200 nm и 300 nm. Для цилиндрических частиц были использованы три типа ориентации: хаотическая, в плоскости слоя и перпендикулярно плоскости слоя.

В работе было исследовано влияние многочисленных параметров на визуальное отображение красящего слоя. К ним относятся:

- Объемная концентрация частиц пигмента
- Размер частиц
- Форма частиц
- Ориентация частиц

- Агломинация
- Шероховатость освещенной поверхности красящего слоя.

Параметры изменялись в достаточно широких пределах – были промоделированы около 260 вариантов различных комбинаций параметров.

6. РЕЗУЛЬТАТЫ

Были получены цветные изображения для всех вариантов моделирования. Каждое изображение в таблицах состоит из двух частей как показано на рис. 6. Левая вертикальная полоса изображает красящий слой на бумаге при естественном освещении. Вертикальная координата задает угол наблюдения, меняющийся от 0° (верх) до 60° (низ). Квадратное изображение справа представляет диффузную ДФО при освещении параллельным светом под заданным углом. Вертикальная координата есть такой же угол наблюдения, горизонтальная координата задает угол освещения, изменяющийся от 0° до 60° слева направо. Для лучшего сравнения яркость всех изображений была выбрана так, чтобы чисто Ламбертовская поверхность с интегральным коэффициентом диффузного отражения равным 1 давала цвет (255,255,255) в цветовом пространстве RGB.

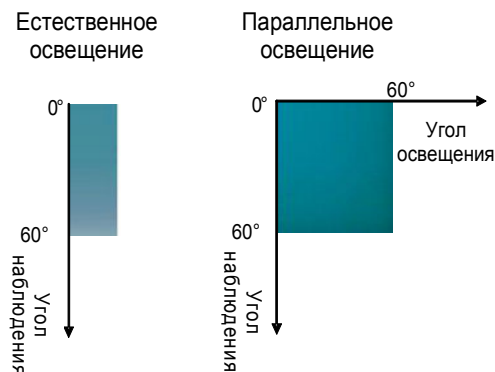


Рис. 6. Визуальное представление результатов моделирования.

Были изучены полученные изображения для всех 260 вариантов моделирования и сделаны следующие выводы:

О влиянии размеров частиц. Цвет красящего слоя становится более насыщенным при росте концентрации пигмента (что вполне естественно) и при уменьшении размеров частиц (а вот такое поведение цвета трудно предположить без моделирования). Кроме этого, влияние размеров сильнее при малых концентрациях пигмента и ослабевает с ростом концентрации. Для шарообразных и цилиндрических частиц роль размера практически одна и та же.

Для объяснения такой роли размеров частиц были проанализированы спектрограммы пропускания красящего слоя для случая шарообразных частиц. Для концентрации 15% пропускание красящего слоя из частиц диаметром 300 nm значительно отличается от пропускания молекулярной смеси (бесконечно малых частиц). Причина такого эффекта заключается в сильных флуктуациях коэффициента преломления в объеме красящего слоя, особенно важны флуктуации мнимой части коэффициента преломления, определяющей поглощение в слое. Для высокой

концентрации 50% пропускание практически совпадает для разных размеров частиц.

При невысоких концентрациях пигмента и достаточно крупных частицах в объеме слоя возникают участки, поглощение в которых ослаблено за счет флуктуаций. За счет этих участков слой пропускает заметно больше света в тех частях видимого спектра, которые хорошо поглощаются при более однородной структуре слоя. Естественно, что размеры областей с уменьшенным поглощением уменьшаются с ростом концентрации или при уменьшении размеров пигментных зерен (за счет их более однородного пространственного распределения).

Также было замечено, что насыщенность цвета при агглютинации соответствует насыщенности цвета для частиц большего размера, что вполне естественно, поскольку образование кластеров увеличивает эффективный размер пигментных зерен.

Программный продукт, построенный на описанных алгоритмах, используется производителем комплексов печати фотографий и цветных принтеров. С помощью программного моделирования возможно отследить тенденции изменения итогового цвета в зависимости от использования частиц различной формы и при различных концентрациях, влияния выравнивания частиц в процессе высыхания и т.д. Это существенно снижает затраты при разработке нового вида чернил.

Работа поддержана грантом РФФИ № 08-01-00649, грантом Президента РФ «Ведущие научные школы» НШ-8129.2010.9, а также фирмой Integra Inc. (Япония).

7. ЛИТЕРАТУРА

- [1] C.J. Curtis, S.E. Anderson, J.E. Seims, K.W. Fleischer and D.H. Salesin. Computer-generated watercolor. In Proceedings of the 24th Annual International Conference on Computer Graphics and Interactive Techniques. ACM Press/Addison-Wesley Publishing Co., 1997, pp. 421-430. DOI= <http://doi.acm.org/10.1145/258734.258896>.
- [2] И. Исимару, Распространение и рассеяние волн в случайно-неоднородных средах. М.: Мир, 1981.
- [3] S. Ershov, K. Kolchin, K. Myszkowski, Rendering pearlescent appearance based on paint-composition modeling. Computer Graphics Forum, Vol. 20, No 3, 2001, pp. 227-238.
- [4] S. Ershov, A. Khodulev, K. Kolchin, Simulation of sparkles in metallic paint. Proceeding of GraphiCon'1999, Moscow, 1999, pp. 121-128.
- [5] N. Lobalzo, A. Voloboy, Physically based lighting model for cloth and its validation. Proceeding of GraphiCon'2008, Moscow, 2008, pp. 61-68.
- [6] В.А. Дебелов, М.А. Саттаров, Проблемы реалистической визуализации кристаллов. Труды Графикон-2003, Москва, 2003, стр. 221-227.
- [7] A. Weidlich, A. Wilkie, Modeling aventurescent gems with procedural textures. Proceeding of SCCG'2008, Budmerice, Slovakia, 2008, pp. 61-68.
- [8] P. Sikachev, I. Tisevich, A. Ignatenko, Using quad-trees for acceleration of physically-based image-space rendering of

glare. Proceeding of GraphiCon'2008, Moscow, 2008, pp. 69-73.

- [9] Н.С. Кошляков, Э.Б. Глинер, М.М. Смирнов, Уравнения в частных производных математической физики. М.: Высшая школа, 1970.
- [10] Л.Э. Эльсгольц, Дифференциальные уравнения и вариационное исчисление. М.: Наука, 1969.
- [11] Л.Д. Ландау, Е.М. Лифшиц, Теоретическая физика. Том III. Квантовая механика. М.: Наука, 1989.
- [12] А.Г. Волобой, С.В. Ершов, С.Г. Поздняков Решение дифракционной задачи для моделирования освещения тонкого красящего слоя. Препринт ИПМ им. М.В. Келдыша РАН № 75, 2009. <http://www.keldysh.ru/e-biblio/>
- [13] С.М. Рыгов, Ю.А. Кравцов, В.И. Татарский, Введение в статистическую радиофизику. Часть II. М.: Наука, 1978.

Abstract

One approach to the problem of lighting simulation inside of ink or paint layer is described. The problem arose in the task of realistic rendering of ink basing on information about shape of ink pigments and layer structure. High pigment volume concentration characterizes ink layer structure. Ray tracing or LTE are hardly applicable here. Therefore it was decided to solve it as a diffraction problem basing on wave optics methods. The article describes task formulation, algorithm of generation of explicit ink layer geometry with high PVC, simulation of visual appearance of ink on a paper. Results and conclusions are presented in the end of article.

Keywords: *realistic rendering, lighting simulation of paint or ink layer, physically based lighting, BRDF/BTDF, surface optical properties.*

Authors:

Alexey G. Voloboy, PhD, senior researcher, Keldysh Institute for Applied Mathematics RAS (KIAM RAS). E-mail: voloboy@gin.keldysh.ru

Sergey V. Ershov, PhD, senior researcher, KIAM RAS.

Edward S. Klyshinsky, PhD, researcher, KIAM RAS.

Sergey G. Pozdnyakov, researcher, NRNU "MEPHI".

Метод Адаптации Цифровых Изображений Для Людей с Аномальным Цветовосприятием (Дихроматов).

В.А. Ковалёв, Э.В. Снежко, В.И. Архипов
Объединённый институт проблем информатики
Национальной академии наук Беларуси, Минск, Беларусь
{vassili.kovalev, eduard.snezhko, arkipau}@gmail.com

Резюме

В статье представлен новый подход к адаптации мультимедийного контента для людей с нарушенным цветовосприятием (дихроматов). При обработке изображения (или видео) представленным способом, учитываются его цветовые характеристики, что позволяет оптимально контрастировать его для лучшего восприятия дихроматом.

Ключевые слова: *цветной медиаконтент, компьютерная графика, восприятие цветов.*

1. ВВЕДЕНИЕ

Общепризнанно, что наиболее действенными подходами повышения эффективности донесения визуальной информации являются персонализация и адаптация визуального содержимого под предпочтения пользователя и его индивидуальные характеристики. Цвет является одной из ключевых составляющих визуального содержимого. Однако около 8% мужчин (и очень малая доля женщин) имеют аномальное цветовосприятие.

Большинство людей (трихроматы) различат множество цветов, образующих в ортогональном цветовом пространстве трёхмерную фигуру, называемую «гамут». Люди с аномальным цветовосприятием (дихроматы) различают цвета, множество которых образует двумерную поверхность, вписанную в гамут. Таким образом, всё трёхмерное пространство цветов, различаемых трихроматами значительно сужается для дихромата. Многие цвета, значительно отличающиеся для трихромата для дихромата практически неразличимы, снижается цветовой контраст.

Среди дихроматов выделяются три группы: протанопы, деутеранопы и тританопы, отличающиеся формой множества воспринимаемых цветов.

В данной статье рассматриваются две наиболее распространённые группы дихроматов протанопы (PRO) и деутеранопы (DEU).

Авторами был разработан новый метод преобразования цветов, который улучшает восприятие цветных сцен, изменяя цветовой контраст, воспринимаемый дихроматами, таким образом, чтобы он стал близок контрасту, воспринимаемому людьми с трихроматическим зрением. Предложенный метод преобразования цветов является адаптивным и предоставляет решение для каждого отдельного изображения, серии изображений или относительно однородной видеопоследовательности. Данный метод может быть использован для репродукции

мультимедийных материалов различного типа, включая художественные картины, цифровые изображения и фотографии, видеоматериал различных типов. Визуальное содержимое, адаптированное для дихроматов, делается персонализированным, а также более эффективным для донесения основной идеи и ключевых акцентов, присутствующих на большинстве видов визуального содержимого.

Предложенный метод преобразования цветов может быть использован во всех областях деятельности, в которых цветоаномалы нуждаются в адаптации отображаемой информации для их индивидуального зрительного восприятия. Они включают в себя использование мобильных устройств различных типов, цифровую фотографию, кинофильмы и видео-музыкальные продукты, Интернет, игровую индустрию, персонализированное телевидение, цветные объекты культурного и исторического наследия, публичные и профессиональные библиотеки и т. д. Одной из наиболее широкой областью применения является широкий набор современных мобильных устройств (мобильные телефоны, смартфоны, наладонные устройства). Второй обширной рыночной нишей является реализованные аппаратно решения для преобразования цветов, встроенные в дорогие компьютерные мониторы высшего ценового уровня, используемые в профессиональной обработке изображений, видеопоследовательностей и компьютерной графики людьми с аномальным цветовосприятием.

2. МЕТОДЫ И РАБОЧЕЕ ОКРУЖЕНИЕ

По сути метод адаптации цветного содержимого заменяет присутствующие на оригинальном изображении цвета на другие. В соответствии с обобщенной формулировкой этого метода, замена не имеет никаких ограничений, то есть белый цвет может быть заменен на черный. Единственной целью адаптации является минимизация некоторой функции стоимости посредством итеративного процесса. Функция стоимости задается таким образом, чтобы отразить, насколько «плохим» является оригинальное изображение в смысле цветового контраста для дихроматов.

Задавая определенным образом целевую функцию, можно достичь различных целей. Эти цели могут быть разделены следующим образом. Во-первых, использование адаптированного цветного содержимого в личном окружении, то есть во всех случаях, когда цветовая палитра некоторого устройства может быть настроена пользователем-цветоаномалом при помощи просто включения функции адаптации цветов (в мобильных телефонах, компьютерных мониторах, телевизионных приемниках и т. д.).

Второй целью является использование адаптированного цветового содержимого в публичном окружении. Данная область приложения также очень обширна (диаграммы, транспортные карты, финансовые графики, географические карты и т. д.). Такой тип визуальной информации должен быть хорошо воспринимаем как пользователями с трихроматическим зрением, так и с дихроматическим. Также следует отметить, что «удовлетворительность» в восприятии должна быть не только в смысле заметного для всех цветового контраста, но также быть эстетически приемлемой.

В данной работе мы оперируем с цветами, присутствующими на каждом статичном изображении, в пространстве CIELAB (CIE 1976 L*a*b*) [1]. Цветовое пространство CIELAB было использовано по той причине, что оно изначально было разработано с целью задания координат цветов таким образом, что координаты любых двух цветов (то есть расстояние между ними) позволило бы предсказать из визуальную разницу.

Тем не менее, авторы понимают, что система CIELAB не была разработана как система визуального представления. Евклидово расстояние, метрика CMC(1:c) либо метрика CIE2000 между двумя точками в CIELAB дают только приближение воспринимаемого различия между соответствующими цветами. Однако пока отсутствует лучшая формулировка использовать данное пространство представляется разумным. Далее по тексту данное пространство будет обозначаться как LAB для краткости.

Цвета большинства визуального содержимого заданы в пространстве RGB. Поэтому для того, чтобы работать с цветами в пространстве LAB, нам вначале следует перевести их из RGB в LAB. Преобразование из RGB в LAB и обратно осуществлялось с использованием конвейера $RGB \Leftrightarrow XYZ \Leftrightarrow LAB$ в соответствии с формулами, приведенными в разделе 2.1.

Тесты по преобразованию цветов были проведены с различными комбинациями параметров для преобразования из RGB в LAB и обратно (таких как стандартный наблюдатель МКО, стандартное излучение МКО, тип рабочего пространства RGB). Наиболее приемлемыми параметрами были признаны следующие:

стандартный наблюдатель 2^0 , стандартный излучатель D50 и рабочее пространство Beta RGB. В данной работе оно было использовано по причине приемлемой комбинации параметров «эффективность области видимости в LAB» (процент всей области видимости LAB пространства, покрываемого данным пространством RGB), а также «эффективность кодирования» (доля кодирующего пространства (RGB), которая представляет реальные цвета). Для Beta RGB эти значения составляют приблизительно 69,3% и 99,0% соответственно. Все примеры и изображения далее по тексту были сформированы с использованием приведенного набор параметров.

2.1. Вычисления

Учитывая все вышеупомянутые параметры (стандартный наблюдатель 2^0 , стандартный излучатель D50, рабочее пространство Beta RGB и координаты соответствующей точки белого

$(X_n, Y_n, Z_n) = (0.96422, 1.00000, 0.82521)$, для преобразования из RGB в XYZ были использованы следующие матрицы:

$$M = \begin{pmatrix} 0.67125 & 0.17458 & 0.11838 \\ 0.30327 & 0.66379 & 0.03294 \\ 0.0000 & 0.04070 & 0.78451 \end{pmatrix}$$

$$M^{-1} = \begin{pmatrix} 1.68323 & -0.42824 & -0.23601 \\ -0.77102 & 1.70656 & 0.04469 \\ 0.04000 & -0.08854 & 1.27236 \end{pmatrix}$$

$RGB \Rightarrow XYZ: [XYZ] = [rgb]M$, где $r = R^\gamma$, $g = G^\gamma$, $b = B^\gamma$, если система RGB – НЕ sRGB.

$XYZ \Rightarrow RGB: R = r^{\frac{1}{\gamma}}$, $G = g^{\frac{1}{\gamma}}$, $B = b^{\frac{1}{\gamma}}$, опять же, если система RGB – НЕ sRGB. $[rgb] = [XYZ]M^{-1}$. Параметр γ в обоих случаях зависит от типа рабочего пространства RGB и для Beta RGB $\gamma = 2.2$.

Имея вычисленные координаты (X, Y, Z) некоторого цвета и координаты точки белого (X_n, Y_n, Z_n) , компоненты в LAB вычислялись следующим образом:

$$L = 116f_y - 16$$

$$a = 500(f_x - f_y)$$

$$b = 200(f_y - f_z), \quad \text{где}$$

$$f_x = \begin{cases} \frac{kx+16}{116}, & x \leq \varepsilon \\ \sqrt[3]{x}, & x > \varepsilon \end{cases}$$

$$f_y = \begin{cases} \frac{ky+16}{116}, & y \leq \varepsilon \\ \sqrt[3]{y}, & y > \varepsilon \end{cases}$$

$$f_z = \begin{cases} \frac{kz+16}{116}, & z \leq \varepsilon \\ \sqrt[3]{z}, & z > \varepsilon \end{cases}, \quad \text{где}$$

$$x = \frac{X}{X_n}, \quad y = \frac{Y}{Y_n}, \quad z = \frac{Z}{Z_n}, \quad \varepsilon = 0.008856, \quad k = 903.3$$

Обратное преобразование $LAB \Rightarrow XYZ$ осуществлялось следующим образом:

$$X = xX_n, \quad Y = yY_n, \quad Z = zZ_n,$$

$$x = \begin{cases} \frac{116f_x - 16}{k}, & f_x^3 \leq \varepsilon \\ f_x^3, & f_x^3 > \varepsilon \end{cases}$$

$$y = \begin{cases} \frac{L}{k}, & L \leq k\varepsilon \\ \left(\frac{L+16}{116}\right)^3, & L > k\varepsilon \end{cases}$$

$$z = \begin{cases} \frac{116f_z - 16}{k}, & f_z^3 \leq \varepsilon \\ f_z^3, & f_z^3 > \varepsilon \end{cases},$$

где (X_n, Y_n, Z_n) - также координаты точки белого, а

$$f_x = f_y + \frac{a}{500}, \quad f_z = f_y - \frac{b}{200},$$

$$f_y = \begin{cases} \frac{ky + 16}{116}, & y \leq \varepsilon \\ \frac{L + 16}{116}, & y > \varepsilon \end{cases}$$

Все вычисления выполнялись с использованием чисел с плавающей точкой двойной точности.

Как уже отмечалось во многих публикациях, стандартное расстояние между цветами в CIELAB

$$CIE76 = \sqrt{(\Delta L)^2 + (\Delta a)^2 + (\Delta b)^2}$$

не может быть использовано как действительная метрика для измерения воспринимаемого различия между цветами. После признания данного факта само цветовое пространство CIELAB не было переделано для решения данной проблемы. Вместо этого были предложены более сложные формулы вычисления расстояния между цветами: CMC [2], CIE94 и CIE2000[3].

Принимая все вышеуказанное во внимание, ниже приведены аргументы за и против использования CIE76 для разработанного метода адаптации.

За:

- ✓ требует выполнения меньшего количества операций по сравнению с другими формулами вычисления расстояния;
- ✓ обеспечивает устойчивое воспринимаемое различие, если значение больше 20 (едва различие достигается при значении около 5);
- ✓ инвариантно относительно вращения/сдвига облака точек изображения в пространстве LAB, обеспечивая таким однородный контекст улучшения изображений (описанный в разделе 2.3).

Против:

- ✓ неточное предсказание воспринимаемого различия цветов.

Так как мы рассматриваем не задачу совмещения цветов, а задачу привнесения более заметного цветового контраста в контекст цветного содержимого с усеченной цветовой палитрой, было решено, что

достоинства расстояния CIE76 превосходят его недостатки. Кроме того, исследования по разработке универсального однородного цветового пространства все еще продолжаются, и в случае успеха разработанный авторами метод, возможно, будет давать более лучшие результаты. Однако в настоящий момент использованием неоднородного пространства CIELAB не является критическим моментом.

2.2. Материалы

Предполагается, что исходные изображения заданы в RGB, так как большинство цифровых устройств представления цветов хранят изображения в данном формате. Если в каком-либо случае это не так, изображения могут быть переведены в RGB либо напрямую в LAB, так как мы заинтересованы в представлении изображений в LAB.

Каждое отдельное изображение (статичное либо кадр видеопоследовательности) рассматривается как облако точек в трехмерном пространстве, т.е. каждый пиксел представлен как точка в 3D с координатами, соответствующими ее компонентам L, а и b (рисунок 1).

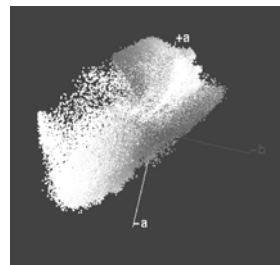


Рисунок 1: Пример изображения и соответствующего ему облака точек в CIELAB

Основываясь на результатах моделирования цветовосприятия дихроматами [4, 5, 6], каждой точке из RGB можно поставить в соответствие точку из видимой области протанопа, девтеранопа и тританопа. Эти уменьшенные области видимости получаются в результате «схлопывания» трехмерного куба RGB в некоторую поверхность, являющуюся подмножеством куба RGB. Далее по тексту эти области будут обозначаться для удобства как PRO и DEU.

2.3. Адаптация

После проецирования облака точек некоторого исходного изображения на поверхности PRO или DEU, также получается множество точек. Обозначим через $d_{i,j}$ евклидово расстояние CIE76 в цветовом пространстве LAB между некоторыми двумя цветами C_i и C_j . Очевидно, что некоторые точки i и j (например, соответствующие красному и зеленому цветам), с большим для трихромата расстоянием $d_{i,j}^N$ между ними в пространстве LAB (другими словами — хорошую различимость для трихромата), после проецирования на

поверхности PRO и DEU будут иметь более низкую различимость, т. е. малые значения $d_{i,j}^P$ и $d_{i,j}^D$ соответственно. Это легко заметить на Рисунке 2, так как поверхности PRO и DEU изменяются незначительно вдоль оси a (ось перехода от красного цвета к зеленому). Примеры подобного уменьшения контраста также приведены в Таблице 1.

Таблица 1: Сравнение расстояний между некоторыми цветами в областях видимости NORM, PRO, DEU.

	1	2	3
c_i , как $[r \ g \ b]$	[238 108 27]	[241 47 5]	[142 88 38]
c_j , как $[r \ g \ b]$	[56 106 10]	[64 119 15]	[79 119 35]
$d_{i,j}^N$	132.44	159.62	74.93
$d_{i,j}^P$	13.47	13.59	15.69
$d_{i,j}^D$	47.74	37.19	1.62

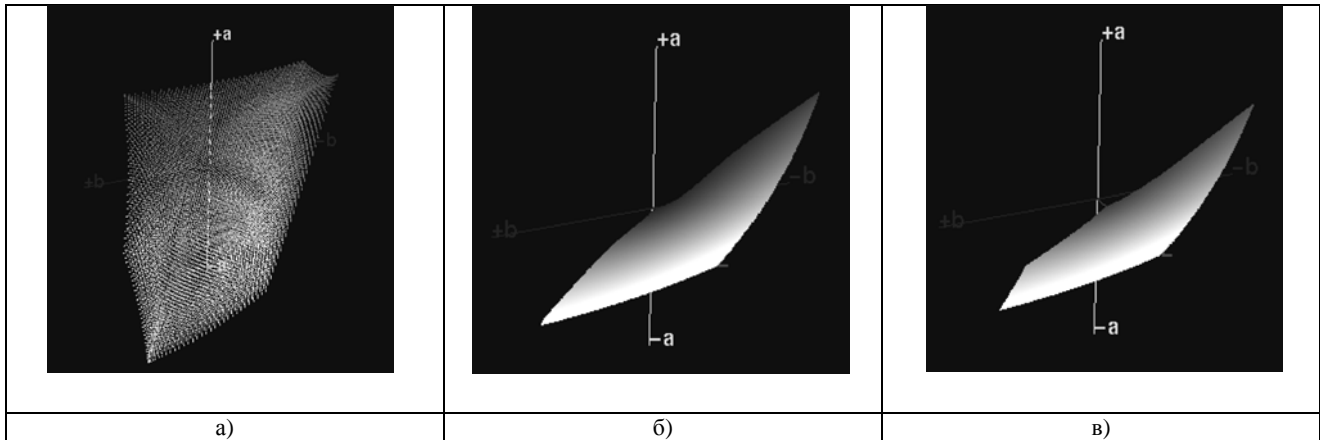


Рисунок 2: Области а) куба RGB в пространстве CIELAB; б) видимости протанопа; в) видимости девтеранопа.

2.3.1 Общая схема адаптации

Общая формулировка метода адаптации изображений состоит в применении преобразований к облаку точек исходного изображения в цветовом пространстве LAB с целью минимизации некоторой функции стоимости. При помощи функции стоимости возможно сформулировать, как близко должно быть адаптированное изображение относительно исходного, либо насколько должен увеличиться цветовой контраст для пользователей с аномальным цветовосприятием после адаптации и т. д. Таким образом, общая формулировка проблемы может быть формализована следующим образом:

$$C_{\text{optimized}} = \arg \min_T (\alpha U^{CB}(C_T) + \beta U^N(C_T)) \quad (1)$$

где:

1. RGB_{LAB} — область в пространстве LAB, границы которой задаются всем возможными триплетами RGB некоторого взятого рабочего пространства RGB;
2. C - любое облако точек в LAB;

3. C^N - облако точек в LAB исходного изображения I ;
4. C^{CB} - «схлопнутое» облако точек исходного изображения, где $CB \in \{PRO, DEU\}$;
5. $T(\cdot)$ - любое преобразование, примененное к облаку точек C с единственным ограничением - $C_T = T(C)$ должно оставаться внутри области RGB_{LAB} ;
6. $U^{CB}(C_T)$ - функция стоимости, которая определяет, насколько «хорошим» является контраст адаптированного изображения для пользователей с аномальным цветовосприятием;
7. $U^N(C_T)$ - функция стоимости, которая определяет, насколько цвета облака точек C_T изменились для пользователей с обычным

цветовосприятием по сравнению с исходным изображением C .

2.3.2 Оптимизация на основе аффинных преобразований.

Предложенный метод оптимизации цветов изображений состоит в применении двух аффинных преобразований (вращение и сдвиг) к облаку точек исходного изображения в цветовом пространстве LAB. Эти преобразования не изменяют евклидово расстояние между любыми двумя цветами в LAB, и в то же время посредством их возможно увеличить воспринимаемое различие между изначально смешивающихся для дихроматов цветов.

Этот процесс может быть представлен как совмещение облака точек с поверхностью видимой области дихромата. На каждой итерации процесса оптимизации производится поиск более лучшего положения и ориентации облака точек изображения по отношению к текущим. Это означает, что каждая точка из C может быть смещена на вектор $(\Delta x, \Delta y, \Delta z)$ и повернута относительно центра облака точек на углы $(\alpha_x, \alpha_y, \alpha_z)$. Масштабирование облака точек может применяться для того, чтобы не выйти за пределы области RGB_{LAB} на каждой итерации, а также для попытки максимального растягивания облака точек.

В таком случае функцию стоимости можно формально записать следующим образом:

$$U_{cost} = \alpha \sum_{i=1}^{N_C} \sum_{j=1}^{N_C} \frac{|d_{i,j}^N - d_{i,j}^{CB}|}{N_C} + \beta \left(\sum_{i=1}^{N_C} \frac{|c_i - c_i^0|}{N_C} \right)^2 \quad (2)$$

где $d_{i,j}^N$ - расстояние для пользователей с обычным цветовосприятием, $d_{i,j}^{CB}$ - расстояние для пользователей с аномальным цветовосприятием, N_C - количество уникальных цветов, присутствующих на изображении (количество точек в облаке точек), а CB - либо PRO , либо DEU .

В общем случае критерий совмещения может быть сформулирован различными способами. Во-первых, мы можем также учитывать частоту встречаемости цветов и сформулировать взвешенную функцию стоимости. Во-вторых можно учитывать информацию о совместной встречаемости цветов, присутствующих на изображении в пределах некоторой окрестности (например, на расстоянии 3 или 5 пикселей). Это может быть осуществлено посредством специальной матрицы совместной встречаемости W (Рисунок 3, [7]). Количество столбцов и строк в этой матрице равно размеру цветовой палитры исходного изображения.

Элемент $W_{i,j}$ матрицы W равен количеству цветов C_i и C_j , встречающихся на исходном изображении на расстоянии менее либо равных d .

Например, взвешенную функцию стоимости можно определить как

$$U_w^{CB} = \sum_{i=1}^{N_C} \sum_{j=1}^{N_C} |d_{i,j}^N - d_{i,j}^{CB}| (1+n_i)(1+n_j) \quad (3)$$

где n_i - относительная частота наиболее часто встречаемого цвета на изображении к частоте цвета C_i .

В обозначениях уравнения (1) предложенный метод адаптации имеет следующие параметры:

- $\alpha = 1, \beta = 1;$
- $C_T = T(C) = (\text{translate}(\text{rotate}(C)))$;
- $U^{CB} = U_{cost}$ из уравнения (1), либо $U^{CB} = U_{cost}^w$ из уравнения (3).

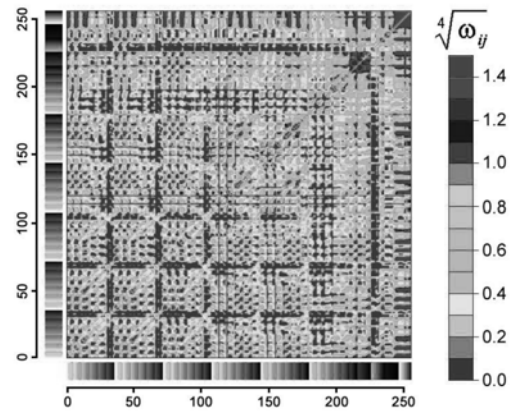


Рисунок 3: Матрица частот совместной встречаемости цветов, вычисленная по базе, содержащей 12 000

изображений. Элементы матрицы $W_{i,j}$ представлены с использованием нелинейного масштаба, приведенного справа. Высокие значения на диагонали матрицы были опущены для того, чтобы отразить детали остальной части матрицы.

До настоящего времени были протестированы два метода оптимизации для минимизации целевой функции (2). Первый метод называется Дифференциальная эволюция [8], и был разработан для поиска с высокой вероятностью глобального минимума многомерных, нелинейных, многомодальных функций стоимости. Так как данный метод относится к классу стохастических методов оптимизации, результаты оптимизации, вообще говоря, могут несколько отличаться на одних и тех же входных данных.

Второй метод называется методом градиентного спуска и имеет преимущество перед предыдущим в том, что дает один и тот же результат на одних и тех же входных данных. Однако он может привести к нахождению локального минимума, пропустив глобальный. Также он довольно сильно зависит от шагов дискретизации величин $\Delta x, \Delta y, \Delta z, \alpha_x, \alpha_y, \alpha_z$.

3. РЕЗУЛЬТАТЫ

Набор тестовых изображений состоял из следующих групп: 1) тесты Рабкина; 2) классические картины; 3)

изображения природы; 4) изображения сцен внутри помещений. Был проведен сбор большой базы таких изображений и осуществлена автоматическая массовая адаптация. В зависимости от содержимого изображений отношение

4. ЗАКЛЮЧЕНИЕ

Предложен новый метод адаптации цветного содержимого для пользователей с аномальным цветовосприятием. Сформулирована общая схема адаптации и потенциальные области ее применения. Также описан один из частных подходов адаптации на основе жестких аффинных преобразований. Эффективность метода была протестирована на основе симуляции видения дихроматов, а также самими пользователями с аномальным цветовосприятием.

Дальнейшая работа будет посвящена трем направлениям: 1) повышению вычислительной эффективности реализации метода; 2) осуществление масштабного тестирования разработанного метода с привлечением тестовой группы людей с аномальным цветовосприятием; 3) исследование других способов формулировки целевой функции и, следовательно, возможности задания других целей адаптации; 4) проведение исследований по сохранению эстетической привлекательности изображений после адаптации. Последний аспект является очень важным, так как желательный эмоциональный эффект от восприятия цветного содержимого должен быть сохранен в каждой области применения. Следует отметить, что это сложная задача, так как тесно связана с психологическими процессами, которые на данный момент формализованы достаточно плохо.

$$\frac{U_{\text{cost}}(C) - U_{\text{cost}}(C_T)}{U_{\text{cost}}(C)} * 100\%$$

изменялось приблизительно от 15% до 45%.

5. СПИСОК ЛИТЕРАТУРЫ

- [1] Hunt, R.W. Measuring Color: 3rd edition / R.W. Hunt. - Fountain Press, 1998.
- [2] Clarke, F. Modification to the JPC79 colour-difference formula / F. Clarke, R. McDonald, B. Rigg // Journal of the Society of Dyers and Colourists. - № 100. - 1984. - P. 128–132 and P. 281–282.
- [3] Luo, M.R. The development of the CIE 2000 colour difference formula / M.R. Luo, G. Cui, B. Rigg // Color Research and Application. - № 26. - 2001. - P. 340–350.
- [4] What do color-blind people see? / F. Viénot [et al.] // Nature. - № 376. - 1995. - P.127–128.
- [5] Brettel, H. Computerized simulation of color appearance for dichromats / H. Brettel, F. Viénot, J. Mollon // Journal Optical Society of America. - № 14. - 1997. - P. 2647–2655.
- [6] Viénot, F. Digital video colourmaps for checking the legibility of displays by dichromats / F. Viénot, H. Brettel, J. Mollon // Color Research and Application. - № 24. - 1999. - P. 243–252.
- [7] Kovalev, V. Optimising the choice of colours of an image database for dichromats / V. Kovalev, M. Petrou // Machine Learning and Data Mining in Pattern Recognition. - 2005. - P. 456–465.
- [8] Storn, R. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces / R. Storn, K. Price // Journal of Global Optimization. - № 11. - 1997. - P. 341–359.

Методики выделения связных компонент в штриховых бинарных изображениях

Максим Стержанов
БГУИР

Минск, Беларусь
accept@bk.ru

Аннотация

В данной работе рассматривается задача выделения связных компонент (СК) штриховых бинарных изображений. Производится классификация и описание алгоритмов. Рассматриваются некоторые методики оптимизации. Осуществляется исследование и сравнительный анализ методов, оценивается их производительность. Предлагается модификация классического алгоритма выделения СК на сжатом растре.

Ключевые слова: бинарный растр, связная компонента, штриховое изображение.

1. ВВЕДЕНИЕ

Под выделением связных компонент понимают присвоение уникальной метки каждому объекту изображения [1]. При последующем анализе данные метки служат в качестве идентификаторов при обращении к объектам. Это делает операцию выделения связных компонент неотъемлемой частью почти всех приложений распознавания образов и компьютерного зрения. Например, перед тем как компьютер может определить или классифицировать любой объект изображения (автомобиль, человека, внутренний орган) группы смежных пикселей должны быть идентифицированы и промаркированы. Каждая выделенная группа пикселей соответствует объекту на изображении. Такая группировка смежных пикселей позволяет исследователю получить необходимые для последующего анализа свойства объектов, такие как высота, ширина, периметр, площадь. Очевидно, что задача выделения связных компонент является фундаментальной задачей обработки изображений. Следует отметить, что для многих приложений данная операция является наиболее затратной, т. е. критической. По этим причинам выделение связных компонент до сих пор остается активной областью исследований. Некоторые недавние работы по данной тематике представлены в [2-3]. В настоящей работе проводится классификация и сравнительный анализ алгоритмов выделения связных компонент с целью определения алгоритма, имеющего наилучшую производительность при работе со штриховыми изображениями, объекты которых представлены совокупностью линий, имеющих относительно одинаковую толщину на всем протяжении. Особенности штриховых изображений являются: большой размер (A0), смешанная графическая и текстовая информация, различные типы объектов (отрезки, символы, дуги кривых). Примерами штриховых изображений являются технические чертежи, электрические схемы, таблицы, поэтажные планы зданий.

2. НЕКОТОРЫЕ ОПРЕДЕЛЕНИЯ

Назовем некоторые определения. Два пикселя (В или W) называются связными, если они являются соседями (расстояние между ними равно 1) в выбранной метрике. Заметим, что определение связности симметрично для черных и белых пикселей. Связная компонента (connected component) изображения (СК) – это связное множество пикселей в соответствии с выбранным типом метрики [4]. Мы будем использовать 8-связную метрику. На рис.1 показано бинарное изображение с пятью связными компонентами. СК можно считать единственной структурной единицей растрового изображения.

Под маркировкой связных компонент бинарного изображения В будем понимать формирование маркированного изображения LB, в котором каждому пикселю присвоена метка связной компоненты, которой принадлежит данный пиксель.

Метка представляет собой некоторый идентификатор, используемый в качестве уникального имени объекта. Хотя иногда применяются символьные метки, для маркировки связных компонент часто более удобными оказываются метки в виде положительных целых чисел. На рис 1.б показаны маркированные связные компоненты бинарного изображения рис. 1.а.

1	1	0	0	1	0	1	0	0
0	1	0	0	1	0	1	0	0
0	1	1	1	1	0	1	1	1
0	0	0	0	0	0	0	0	1
1	1	1	0	0	0	0	0	0
0	0	1	1	1	0	1	0	0
1	1	1	0	1	0	0	0	0
0	0	0	0	0	0	0	1	1

а)

1	1	0	0	1	0	3	0	0
0	1	0	0	1	0	3	0	0
0	1	1	1	1	0	3	3	3
0	0	0	0	0	0	0	0	3
2	2	2	0	0	0	0	0	0
0	0	2	2	2	0	4	0	0
2	2	2	0	2	0	0	0	0
0	0	0	0	0	0	0	5	5

б)

Рисунок 1: Маркировка связных компонент: а) исходное изображение; б) маркированное изображение.

Алгоритмы выделения СК могут быть разделены на следующие пять групп [5]: многопроходные, двухпроходные, однопроходные, алгоритмы с использованием иерархических структур представления изображения, параллельные алгоритмы. В данной работе рассматриваются алгоритмы, принадлежащие первой, второй и третьей группам.

Обозначим через I матрицу изображения. Если $I(i,j)=0$, то пиксель является фоновым. Если $I(i,j)=1$, то пиксель принадлежит объекту. Через L обозначим двумерную матрицу, которая используется для хранения информации о метках и имеющую размеры, равные размерам изображения.

3. МНОГОПРОХОДНЫЕ АЛГОРИТМЫ

Алгоритмы данного семейства последовательно сканируют изображение. При первом, начальном проходе каждый объектный пиксель получает временную метку. На каждой итерации значения меток уточняются. Алгоритм заканчивает свою работу тогда, когда не может быть сделано ни одного изменения значения метки [6]. Метки, которые назначаются пикселям объектов до финальной маркировки, называются промежуточными. Скан-проходы осуществляются в противоположных направлениях и чередуются. При прямом проходе изображение сканируется слева направо и сверху вниз. При обратном проходе изображение сканируется справа налево и снизу вверх. Каждый раз при обнаружении черного пикселя его соседи, принадлежащие скан-маске (рис. 2), исследуются для определения метки, которая будет присвоена рассматриваемому пикселю. Если в скан-маске содержатся только фоновые пиксели, то рассматриваемый пиксель получает новую промежуточную метку. Если скан-маска содержит только один пиксель интереса, то рассматриваемый пиксель получает метку соседа. Если скан-маска содержит несколько точек интереса, то их промежуточные метки являются эквивалентными, среди них выбирается метка, представляющая все эквивалентные метки (метка-представитель). В данном случае пикселю присваивается значение выбранной метки-представителя. В простейшем случае в качестве метки представителя выбирается метка с наименьшим значением. Назовем данный алгоритм MPS (Multi-Pass Simple).

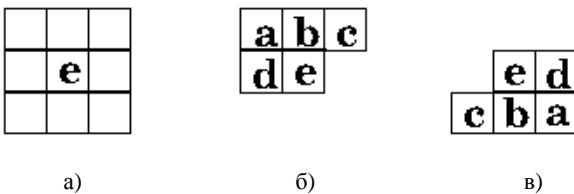


Рисунок 2: Используемая нотация: а) 8 соседей пикселя; б) скан-маска прямого прохода; в) скан-маска обратного прохода.

Обозначим пиксели скан-маски через a, b, c, d, e и будем использовать эти обозначения вместо их (i, j) координат в дальнейшей нотации. Тогда $L[e]$ будет обозначать метку текущего пикселя. Через l обозначим целое число, инициализированное значением 1. Присвоение промежуточных меток в течение первого скан-прохода может быть представлено в виде:

$$L[e] = \begin{cases} 0, & \text{если } I[e] = 0 \\ l, & (l = l + 1), \text{ если } \forall i \in (a, b, c, d), I[i] = 0 \\ \min(L[i]) & (i \in (a, b, c, d) | I[i] = 1), \text{ в противном случае} \end{cases} \quad (1)$$

При последующих скан-проходах метки пикселей получают значение, выраженное следующей формулой:

$$L[e] = \min(L[i]), i \in (a, b, c, d) | I[i] = 1, \text{ если } I[e] = 1 \text{ и } \exists i \in (a, b, c, d), \text{ такое, что } I[i] = 1 \quad (2)$$

Недостатком алгоритма MPS является большое количество скан-проходов, необходимое для конвертации промежуточных меток. Например, при обработке тестового изображения «Зерносушилка» (см. рис. 3) размерами

6206×5883 пикселей требуется 116 прямых и обратных проходов по изображению.

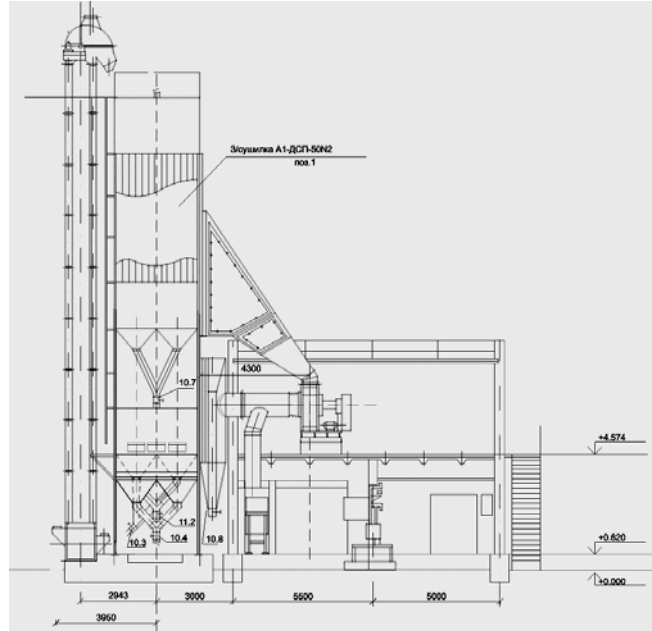


Рисунок 3: Тестовое изображение «Зерносушилка»

Алгоритм может быть улучшен при помощи использования таблицы связности (connection table), хранящей информацию об эквивалентных метках. Таблица связности, предложенная в работе Сузуки [7] является одномерным массивом, имеющим число элементов, равное числу промежуточных меток. Назовем данный алгоритм MPS_СТ. Обозначим через T таблицу связности. При первом проходе значения массивов L и T принимают следующие значения:

$$L[e] = \begin{cases} 0; & \text{если } I[e] = 0 \\ l, T[l] = l, l = l + 1; & \text{если} \\ \forall i \in (a, b, c, d) I[i] = 0 \\ \min(T[L[i]]), T[L[i]] = L[e]; & \text{если} \\ \forall i \in (a, b, c, d) | I[i] = 1 \end{cases} \quad (3)$$

При последующих проходах происходит изменение меток пикселей объектов, которые содержат смежные черные пиксели, принадлежащие скан-маске. Значения меток определяются при помощи следующей формулы:

$$L[e] =_{i \in (a, b, c, d) | I[i] = 1} \min(T[L[i]]), \forall i \in (a, b, c, d) | I[i] = 1, T[L[i]] = L[e] \quad (4)$$

Использование таблицы связности способствует более быстрому распространению меток по сравнению с аналогичными алгоритмами. Проведенные нами эксперименты показывают, что в большинстве случаев данному алгоритму требуется не более 4 скан-проходов. В таблице 1 приведем результаты тестирования многопроходных алгоритмов. Во всех экспериментах тестирование производительности осуществлялось пакетным способом на 127 широкоформатных штриховых изображениях.

Таблица 1. Производительность многопроходных алгоритмов.

Метод	Min, мс	Max, мс	Avg, мс	Disp, мс
MPS	1367	130596	21689	33662
MPS_CT	876	13054	3815	3746

Примечание: В таблицах 1-5 приведено время выделения СК: *min* – минимальное, *max* – максимальное, *avg* – среднее выборочное, *Disp* – квадратный корень среднеквадратичного отклонения.

4. ДВУХПРОХОДНЫЕ АЛГОРИТМЫ

Алгоритм, описанный в статье [8], часто называется классическим, так как он основан на классическом алгоритме поиска связанных компонент на графах. Существуют различные модификации алгоритма, его общая структура заключается в следующем. Изображение последовательно сканируется сверху вниз и слева направо. Новые метки СК присваиваются первому пикселю обнаруженной СК, затем присвоенная метка распространяется на смежные пиксели. Рассмотрим пример на рис. 4.

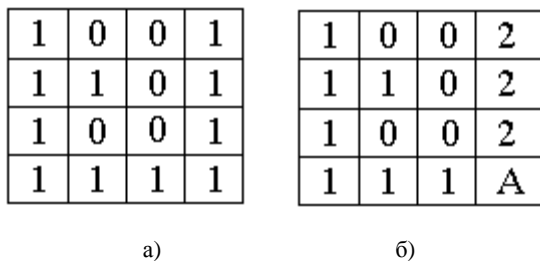


Рисунок 4: Пример маркировки: а) исходное изображение; б) маркированное изображение.

В первой строке имеется два черных пикселя, разделенные двумя белыми. Первый из них получает метку со значением 1, второй – метку со значением 2. Во второй строке первый пиксель получает метку со значением 1, т. к. его сосед сверху уже помечен. Второй пиксель во второй строке также получает метку со значением 1, т. к. его сосед слева помечен. Аналогичные действия выполняются и для третьей строки матрицы изображения. Рассмотрим пиксель А в четвертой строке. Его сосед слева имеет метку со значением 1, а сосед сверху имеет метку со значением 2. Следовательно, все пиксели, имеющие метки со значением 1 и 2 принадлежат одной связной компоненте, другими словами, эти метки являются эквивалентными. Существующие модификации алгоритма отличаются в решении следующих вопросов:

1. Какая метка должна быть присвоена пикселю А?
2. Каким образом хранится информация об эквивалентных метках?
3. Каким образом информация об эквивалентности преобразуется при окончательной разметке?

Классический алгоритм [8] требует большой глобальной таблицы для хранения информации об эквивалентности меток. На первом проходе, называемом фазой сканирования, все объектные пиксели изображения получают временные метки. Если одному пикселю может быть присвоено несколько меток, то выбирается метка с минимальным значением, а информация об эквивалентности меток заносится в таблицу эквивалентностей. В классическом

алгоритме каждая запись в этой таблице представляет собой упорядоченную пару значений, соответствующих эквивалентным меткам. Затем при помощи построения транзитивного замыкания множества эквивалентностей, обнаруженных на первом шаге, выделяются классы эквивалентности. Данная процедура является фазой анализа. Каждому классу эквивалентности присваивается уникальная метка, обычно значение минимальной метки в классе. На втором проходе, называемом фазой окончательной разметки, каждая промежуточная метка заменяется меткой соответствующего класса эквивалентности.

Серьезной проблемой при использовании классического алгоритма является размер таблицы эквивалентностей, которая может иметь очень большой размер при обработке широкоформатных изображений. В предлагаемой нами реализации классического алгоритма таблица эквивалентностей представляет собой хэш-таблицу, реализованную методом цепочек. Данный алгоритм будем называть ТРС (Two pass classic).

Одним из решений проблемы неконтролируемого роста размера глобальной таблицы эквивалентностей заключается в использовании небольшой локальной таблицы эквивалентностей, хранящей информацию о метках, принадлежащих двум скан-строкам изображения – текущей и предыдущей. Следовательно, максимальное число отношений эквивалентности равно числу столбцов матрицы изображения. Каждая строка матрицы изображения последовательно сканируется в течение двух итераций. На первой итерации происходит заполнение локальной таблицы эквивалентностей. Затем производится анализ таблицы эквивалентностей, определяются метки-представители, которые будут присвоены пикселям текущей строки при повторном скан-проходе (вторая итерация). Следовательно, в дальнейшем анализе будут использоваться обновленные значения меток. Следует отметить, что второй скан-проход осуществляется в направлении, противоположном первому. При применении данной техники будем добавлять суффикс `_LM` к названию алгоритма.

Эффективным методом, позволяющим осуществить компактное хранение информации об эквивалентности и быстрый поиск, является использование структуры данных для объединения-поиска [9] (`union-find`). Алгоритм объединения-поиска, который динамически строит классы эквивалентности после обнаружения связанных отношений эквивалентности объектов, стал широко использоваться в различных прикладных задачах. Структура данных, ориентированная на объединение и поиск, позволяет эффективно формировать классы эквивалентности, представленные в виде древовидных структур данных и выполнять операции над ними. Добавление этой структуры данных в алгоритм маркировки оказалось полезным усовершенствованием классического алгоритма поиска связанных компонент на графах.

Опишем структуру данных для объединения-поиска. Каждое непересекающееся множество хранится в форме древовидной структуры, в каждом узле которой хранится метка и ссылка на один родительский узел. Как правило, метка, присвоенная корневому элементу дерева, выбирается в качестве окончательной метки для всех промежуточных меток, входящих в дерево. Процедура поиска принимает в качестве параметра метку *X*, прослеживает родительские узлы вверх к

корню дерева и находит метку корневого узла дерева, которому принадлежит X . Процедуре объединения передаются две метки – X и Y . Для этих меток находятся корневые узлы. Если корневые узлы не совпадают, то узел одной из меток назначается родительским узлом для узла второй метки.

Традиционным способом для представления деревьев является использование указателей. При использовании динамического выделения памяти узлы дерева размещаются в памяти в произвольном порядке. Следовательно, процедура поиска будет обращаться к памяти неэффективно. Многие авторы предлагают хранение дерева при помощи массива, так как ячейки массива размещаются в смежных ячейках памяти.

Если глубина дерева является большой, то поиск корня для листов будет медленным. Поэтому может применяться компрессия пути, т.е. при добавлении узла T , имеющего корень S , к новому корню C для всех узлов, принадлежащих пути от T до S устанавливается корень C . Реализацию классического алгоритма с применением структуры данных для объединения поиска назовем TPC_UF, реализацию с применением компрессии назовем TPC_UF_PC.

Таблица 2. Производительность двухпроходных алгоритмов

Метод	Min, мс	Max, мс	Avg, мс	Disp, мс
TPC	1234	175868	26204	45954
TPC_UF	244	5312	1070	1308
TPC_UF_LM	292	1000	509	200
TPC_UF_PC	147	624	302	138
UF_LM_PC	290	991	496	203

5. ОДНОПРОХОДНЫЕ АЛГОРИТМЫ

Наиболее простыми методами маркировки связанных компонент являются рекурсивные алгоритмы поиска. Процесс поиска сводится к следующим операциям. Находится непомеченный черный пиксель. Ему присваивается новая метка и вызывается процедура поиска *search* всех черных необработанных соседей. Для каждого из найденных соседей производится рекурсивный вызов *search*. Очевидно, что для избежания множественных рекурсивных вызовов поиск в ширину реализуется с помощью структуры данных очередь. Изначально в очередь заносятся координаты исходного черного пикселя. Затем из очереди извлекается элемент, он помечается текущей меткой СК, находятся все непомеченные соседи и помещаются в очередь. Обработка продолжается до тех пор, пока очередь не опустеет. Данная стратегия носит название поиск в ширину (Breadth-first search, BFS). В качестве структуры данных для хранения информации о подлежащих обработке пикселях может использоваться стек. В таком случае будет осуществляться поиск в глубину (Depth-first search, DFS).

Наиболее эффективным алгоритмом данного семейства является алгоритм обхода контуров (СТ), предложенный в работе [10]. Алгоритм осуществляет проход по изображению, при встрече непомеченного объектного пикселя проверяется его принадлежность внутренней либо внешней границе объекта. Если пиксель является граничным, то создается новая метка и осуществляется прослеживание контура. Все контурные точки получают значение созданной метки. В

противном случае соседний слева пиксель является уже помеченным, текущий пиксель получает его метку.

Таблица 3. Производительность однопроходных алгоритмов

Метод	Min, мс	Max, мс	Avg, мс	Disp, мс
DFS	159	663	318	147
BFS	170	686	325	154
СТ	127	340	192	62

6. РЕАЛИЗАЦИЯ НА СЖАТОМ РАСТРЕ

Эффективным методом представления изображения является кодирование длин серий (RLE). Многие алгоритмы нахождения СК могут быть реализованы на сжатом растре [11]. Каждая серия получает дополнительное поле для хранения метки СК. При работе на сжатом растре количество обращений к объектам изображения значительно уменьшается, т.к. каждая серия является группой пикселей и нет необходимости обрабатывать каждый объектный пиксель в отдельности. Более того, фоновые пиксели вообще не участвуют в обработке. Количество промежуточных меток также сокращается, следовательно, уменьшается время, требуемое на разрешение эквивалентностей.

В работе [2] авторы утверждают, что их алгоритм является эффективным при обработке штриховых изображений. На первом проходе анализируется смежность серий в i и $i-1$ скан-строке. Если серия не имеет смежных, ей присваивается новая метка k , которая заносится в позицию k массива R меток-представителей. Пусть две метки A и B принадлежат одной СК, т.е. являются эквивалентными. Тогда всем ячейкам массива R , имеющим значение B присваивается значение A . На втором проходе каждая серия получает значение метки из массива R . Данный алгоритм назовем по именам авторов – HCSW.

Нами предлагается следующая модификация классического двухпроходного алгоритма. На первом проходе сохраняется информация об эквивалентности серий. Предлагается использование следующей структуры данных. Каждая серия конфигурируется не числовым идентификатором метки, а указателем на структуру метки, которая содержит числовой идентификатор, являющийся одновременно и порядковым номером СК. Эквивалентные метки имеют одинаковое значение идентификаторов и хранятся с помощью однонаправленного списка.

Списки структур меток размещаются в ячейках таблицы эквивалентностей, которая является одномерным массивом, причем индекс в таблице для каждого списка равен идентификатору метки.

Если рассматриваемая серия не имеет смежных, то создается новый список, содержащий одну структуру метки со значением k , он помещается в k -ю позицию таблицы. Затем значение k увеличивается на единицу.

Пусть две серии являются смежными и имеют разные числовые значения идентификаторов в структуре метки. Пусть a и b - значения идентификаторов, причем $a < b$. Все идентификаторы, находящиеся в позиции b таблицы имеют значение b . Для всех элементов списка изменим это значение на a и переместим данный список в конец списка, находящегося в позиции a . Чтобы объединение списков осуществлялось быстро, мы для каждого списка храним

указатель на последний элемент. Благодаря этому избегается полный просмотр списка с целью нахождения последнего элемента. После объединения списков сдвинем все ячейки таблицы эквивалентностей на одну позицию вниз, начиная с позиции $b + 1$. Для каждого сдвигаемого списка изменим значение идентификатора метки, т.е. сделаем его равным значению позиции в таблице.

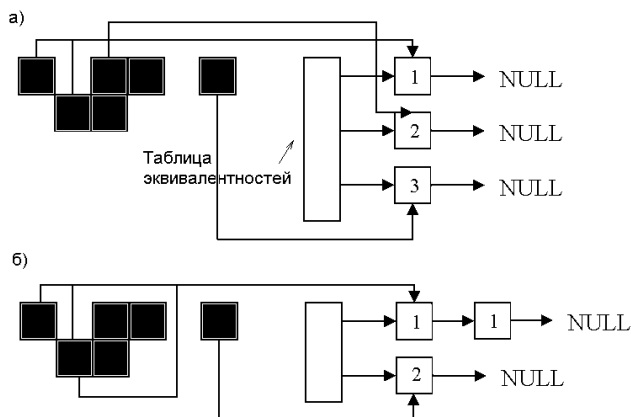


Рисунок 5: Пример работы алгоритма: а) обработка первой скан-строки; б) обработка второй скан-строки.

Второй проход алгоритма является фазой финальной маркировки. Пиксели каждой серии добавляются к некоторой связанной компоненте в соответствии с порядковым идентификатором метки. Заметим, что в данном алгоритме отсутствует отдельная фаза нахождения финальных значений меток, т. к. данные действия выполняются на первом шаге алгоритма.

Назовем предлагаемую модификацию классического алгоритма CL_MOD. Приведем результаты тестирования производительности алгоритмов выделения СК на сжатом растре в таблице 4.

Таблица 4. Производительность алгоритмов на сжатом растре

Метод	Min, мс	Max, мс	Avg, мс	Disp, мс
MP	12	1050	300	410
MP_CT	29	187	74	48
CL_MOD	4	28	15	6
UF	10	489	119	126
UF_PC	6	23	12	5
HCSW	6	33	15	7

7. МЕТОДИКА «РАЗДЕЛЯЙ И ВЛАСТВУЙ»

Известной техникой, применяемой при обработке изображений большого размера, является методика "Разделяй и властвуй", которая позволяет разбить исходную задачу на подзадачи меньшей размерности. В случае с выделением СК изображение разбивается на N прямоугольных областей.

Для каждой области выполняется выделение СК традиционным способом (например, классическим алгоритмом). Затем найденные компоненты объединяются, т. е. происходит "сшивка" областей.

Проведенные нами эксперименты показали, что применение данного подхода позволяет значительно увеличить производительность методов попиксельного анализа, использующих глобальную таблицу для разрешения эквивалентностей, в частности, классического алгоритма[8].

Таблица 5. Производительность некоторых алгоритмов с использованием техники «разделяй и властвуй»

Метод	Min, мс	Max, мс	Avg, мс	Disp, мс
MP	423	4298	1865	1341
MP_CT	444	1476	930	359
TPC	172	2409	827	688
TPC_UF	138	625	265	142
TPC_UF_LM	211	665	343	116
TPC_UF_PC	147	624	302	138
UF_LM_PC	290	991	496	203

8. ЗАКЛЮЧЕНИЕ

В данной работе классифицируются алгоритмы выделения связанных компонент, описываются особенности их программной реализации. Проведенные нами эксперименты показали, что оптимальными алгоритмами для выделения СК на штриховых изображениях являются алгоритмы, использующие сжатое представление изображения. Предлагается модификация классического алгоритма разметки с использованием сжатого раstra. Предложенный алгоритм имеет высокую производительность, прост в реализации и эффективен при обработке широкоформатных изображений. Рассматривается методика «разделяй и властвуй» как эффективное средство увеличения производительности алгоритмов выделения СК, использующих попиксельный анализ.

9. ЛИТЕРАТУРА

- [1] Л. Шапиро, Дж. Стокман, *Компьютерное зрение*, М.: Бинном. Лаборатория знаний, 2006.
- [2] L. He, Y. Chao, K. Suzuki, K. Wu, "Fast connected-component labeling", *Pattern Recognition*, Vol. 42, No. 9., 2009. – pp. 1977-1987.
- [3] H.Hedberg, F.Kristensen, and V.Owall, "Implementation of a labeling algorithm based on contour tracing with feature extraction", *IEEE International Symposium on Circuits and Systems(ISCAS)*, 2007, pp. 1101–1104.
- [4] С. В. Абламейко, Д. М. Лагуновский, *Обработка изображений: технология, методы, применения*, Амалфея, Минск, 2000.
- [5] K. Wu, E. Otoo, and K. Suzuki, "Optimizing two-pass connected-component labeling algorithms", *Pattern Analysis & Applications*, 2009, Vol. 12, No. 2., pp. 206–220.
- [6] R.M. Haralick, *Some Neighborhood Operations*, *Real Time/Parallel Computing Image Analysis*, Plenum Press, New York, 1981.
- [7] K. Suzuki, I. Horiba, N. Sugie, "Linear-time connected-component labeling based on sequential local operations", *Comput. Vis. Image Underst.*, Vol. 89, No. 1, 2003, pp. 1–23.

- [8] A. Rosenfeld, P. Pfaltz, "Sequential Operations in Digital Picture Processing", *Journal of the Association for Computing Machinery*, Vol. 12., 1966.
- [9] C. Fiorio, J. Gustedt, "Two linear time union-find strategies for image processing", *Theor. Comput. Sci.*, Vol. 154, No. 2, 1996, pp. 165-181.
- [10] F. Chang, C.-J. Chen, and C.-J. Lu, "A linear-time component-labeling algorithm using contour tracing technique," *Comput. Vis. Image Underst.*, vol. 93, no. 2, 2004, pp.206–220.
- [11] L. Shapiro, "Connected Component Labeling and Adjacency Graph Construction", *Topological algorithms for digital image processing* / T. Kong [et al.], – Amsterdam, 1996, pp. 1–31.

Благодарности

Автор выражает благодарность своим родителям за поддержку, своему другу Ивану Байдакову за создание тестовой среды и проведение тестов, и Борису Борисовичу Гребенщикову за неповторимую атмосферу, создаваемую его музыкой.

Об авторе

Максим Стержанов – аспирант БГУИР. email: accept@bk.ru.

Abstract

In this paper we analyze a problem of connected component labeling for monochrome line drawings. We provide both a classification and description of the existing algorithms. Some optimization methods are also considered, with experimental measurements and comparative analysis of algorithms being provided. We also present a modification of the classical algorithm for connected component labeling which utilises a compressed representation of the raster image.

Keywords: *binary raster, connected component, line image.*

Морфологическая сегментация изображений земной поверхности с использованием стековых деревьев

Юрий Визильтер, Владимир Горбацевич
ФГУП "ГосНИИАС"

{viz,gvs}@gosniias.ru

Аннотация

В данной статье рассматривается техника морфологической сегментации с использованием критериальных морфологических фильтров. При этом предлагается нетривиальный способ представления данных позволяющий использовать при решении этой задачи метод динамического программирования.¹

Keywords: Морфология, сегментация, динамическое программирование, деревья.

1. Введение

Типовой задачей технического зрения, возникающей при реализации функций наведения и навигации летательных аппаратов, является задача взаимной привязки разномоментных и разномасштабных изображений земной поверхности. Для решения этой задачи традиционно используются корреляционные алгоритмы сравнения изображений. Однако на практике такие алгоритмы обладают рядом известных недостатков, наиболее очевидными из которых являются малая устойчивость к изменениям условий регистрации, а также необходимость хранения в бортовых системах больших объемов видеоданных. В связи с этим современный подход к решению данной задачи предполагает осуществление предварительного компактного кодирования изображений с последующим сравнением уже не изображений, в их описаний. Таким образом, необходимым элементом решения задачи привязки изображений оказывается их сегментация с последующей возможностью более или менее точной реконструкции по описанию.

В данной статье рассматривается техника морфологической сегментации изображений с использованием стековых деревьев. При этом в центре внимания оказывается проблема эффективной алгоритмической реализации описанных в [1] проективных критериальных морфологических фильтров вида:

$$\psi_{\alpha}A = \operatorname{argmin}_{B \in \Omega} F_{\alpha}(A, B) \quad (1)$$

При этом минимизируется критерий морфологического риска

$$F_{\alpha}(A, B) = J(A, B) + \alpha \times Q(B), \quad (2)$$

где $A, B \subseteq \Omega$ - дискретные одномерные или двумерные функции (изображения), имеющие конечные области определения и значения, обобщенно называемые также образцами; $J(A, B) \in [0, +\infty)$ - функционал (критерий) соответствия морфологической реконструкции B наблюдаемому образу A ; $Q(B) \in [0, +\infty)$ - функционал (критерий) качества морфологической реконструкции B ; $\alpha \geq 0$ - действительное число, называемое моделирующим параметром.

В работе [2] для проективных морфологических фильтров на базе критериев типа (1)-(2) была описана эффективная алгоритмическая реализация на основе метода динамического программирования. При этом рассматривались как процедуры аппроксимации, так и процедуры интерполяции одномерных функций и двумерных кривых. К сожалению, строгое обобщение метода динамического программирования на случай оптимизации функционалов от функций, имеющих более одного аргумента, сталкивается с серьезными проблемами, поскольку уже координаты точек из R^2 , в отличие от точек R^1 , не являются полностью упорядоченными. По этой причине граф структурных связей между элементами (пикселями) изображения, в отличие от аналогичного графа для элементов одномерных функций, имеет вид не цепи, а решетки, то есть включает множество циклов. Между тем, как известно из теории динамического программирования, этот метод работает только для таких структур, граф структурных связей между элементами которых имеет вид ациклического графа (АСГ) или дерева [3]. Таким образом, растровое изображение (прямоугольная решетка пикселей) оказывается неподходящей структурой данных для непосредственного строгого применения методов динамического программирования. Следовательно, необходимо определить такое представление данных в алгоритмах решения задачи (1) методом динамического программирования, которое обеспечивало бы непосредственное и естественное обобщение на двумерный случай.

В данной работе в качестве такого унифицированного представления функций предлагается использовать т.н. стековое представление, идея которого восходит, с одной стороны, к идее бинарных морфологических остовов [4], а с другой - к стековому представлению (threshold decomposition) полутоновых изображений [5].

2. Описание одномерных и двумерных данных при помощи деревьев срезовых сегментов

Пусть дана одномерная дискретная функция $f(x) \in \{0, 1, \dots, z-1\} \subseteq Z$, $x \in R$. Определим срез функции на уровне $t \in \{0, \dots, z\} \subseteq Z$ как бинарную функцию (Рис.1)

$$f_t(x) = \{1: f(x) \geq t; 0: f(x) < t\}.$$

Полным срезовым стеком функции $f(x)$ является упорядоченный набор ее срезов

$$f = \{f_0(x), \dots, f_z(x)\},$$

на основе которых можно определить оператор реконструкции срезового стека δ :

$$f(x) = \delta f = \sum_{t=0, \dots, z} \{f_t(x)\} = \max_{t=0, \dots, z} \{t \times f_t(x)\}.$$

При этом между любыми срезами $f_t(x)$ и $f_r(x)$ выполняется т.н. стековое свойство (Рис.1):

$$r > t \Rightarrow f_t(x) \geq f_r(x). \quad (3)$$

¹ Работа выполнена при поддержке РФФИ, проект 09-07-13551-офи ц.

Легко заметить, что $f_0(x) \equiv 1$ и $f_z(x) \equiv 0$, то есть стек начинается наибольшим (единичным) и заканчивается наименьшим (нулевым) бинарным образом.

Представим теперь срез функции $f_i(x)$ на уровне t как комбинацию бинарных функций $f_{i,k}(x)$, соответствующих всем его $n(t)$ непрерывным ненулевым срезovým сегментам

$$f_i(x) = \sum_{k=1, \dots, n(t)+1} \{f_{i,k}(x)\} = \max_{k=1, \dots, n(t)+1} \{f_{i,k}(x)\}$$

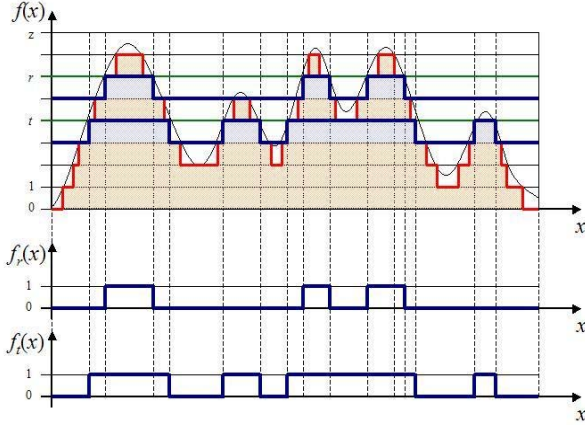


Рис.1. Стековое свойство элементов срезовой цепи.

С точки зрения возможных преобразований разница между срезovým стеком и набором сегментов заключается в том, что между любыми двумя срезами $f_i(x)$ и $f_j(x)$ всегда выполняется *стековое свойство* (3), в то время как в наборе сегментов это соотношение выполняется лишь для некоторых сегментов. Таким образом можно построить граф G_f , каждому узлу которого соответствует срезовой сегмент из f , а наличие ребра между двумя узлами отражает выполнение между ними стекового свойства. Такой граф всегда является ациклическим графом типа “дерево” и позволяет полностью описать исходную функцию. Будем называть такое дерево G_f (рис.2) деревом стековых сегментов или просто стековым деревом функции $f(x)$. Оператор реконструкции по стековому дереву

$$f(x) = \delta G_f$$

вводится аналогично введенному ранее оператору реконструкции по срезовой стеку.

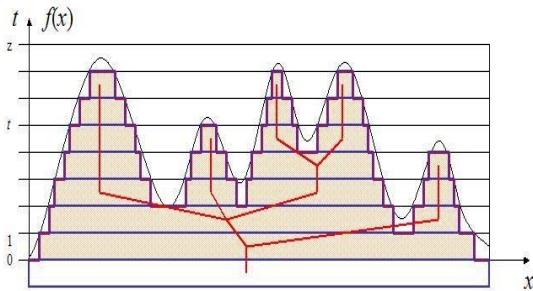


Рис.2. Стековое дерево одномерной функции.

Пусть теперь имеется двумерная функция $f(x,y)$. Представим ее срез $f_i(x,y)$ на уровне t как комбинацию бинарных функций $f_{i,k}(x,y)$, соответствующих всем ее $n(t)$ непрерывным ненулевым срезovým областям.

$$f_i(x,y) = \sum_{k=1, \dots, n(t)+1} \{f_{i,k}(x,y)\} = \max_{k=1, \dots, n(t)+1} \{f_{i,k}(x,y)\}.$$

Введем новую индексацию i , сохраняющую порядок записи областей, определяемый парами индексов (t,k) :

$$(t,k) \rightarrow i: i(t,k) > i(u,m) \Rightarrow \{(t > u) \text{ или } \{(t = u) \text{ и } (k > m)\}\}.$$

Полным набором областей функции $f(x,y)$ назовем упорядоченный набор пар

$$f = \langle (f(x,y), t_i) \rangle_n = \langle (f_0(x,y), t_0), \dots, (f_n(x,y), t_n) \rangle,$$

на основе которого функция может быть восстановлена при помощи оператора реконструкции δ , записываемого в форме аддитивного накопления

$$f(x,y) = \delta f = \sum_{i=1, \dots, n} \{f_i(x,y)\}.$$

или в форме монотонного накопления

$$f(x,y) = \delta f = \max_{i=1, \dots, n} \{t_i \times f_i(x,y)\}.$$

Здесь также можно построить стековое дерево G_f и определить оператор реконструкции по стековому дереву

$$f(x,y) = \delta G_f.$$

3. Морфологическая сегментация с использованием стековых деревьев

Опишем в общем виде задачу морфологической сегментации функций с использованием стековых деревьев.

Поскольку между функциями и стековыми деревьями существует взаимно-однозначное соответствие, для любого оператора φ результат h его применения к функции f всегда может быть представлен в виде стекового дерева G_h с набором областей h . Значит, всегда существует такой оператор преобразования стековых деревьев θ , который соответствует оператору преобразования функций φ :

$$h = \varphi f \Leftrightarrow G_h = \theta G_f,$$

При этом

$$\varphi^2 = \varphi \Leftrightarrow \theta^2 = \theta,$$

то есть любому морфологическому проектору функций φ должен соответствовать морфологический проектор стековых деревьев θ .

Пусть теперь оператор θ является переменным. Тогда задача морфологической сегментации (1) может быть следующим образом обобщена с использованием стековых деревьев:

$$\psi_\alpha f = \operatorname{argmin}_\theta F_\alpha(\delta G_f, \delta \theta G_f), \quad (4)$$

Поскольку стековое дерево – ациклический граф, задача оптимизации (4) может решаться методом динамического программирования для таких критериев (1)-(2), которые представимы в рекурсивной форме на монотонной последовательности поддеревьев дерева G_f [3]. Морфологический фильтр ψ_α (4), реализуемый методом динамического программирования, будем называть ДП-фильтром на базе стекового дерева. Сложность решения задач динамического программирования по «дереву» определяется числом узлов и количеством возможных значений, соответствующих каждому узлу. Поэтому эффективную программную реализацию допускают такие задачи сегментации, в которых на способы изменения значений сегментов накладываются жесткие ограничения.

Рассмотрим в качестве примера типовой задачи морфологической сегментации задачу кусочно-постоянной монотонной интерполяции по стековому дереву.

Задача интерполяции полутоновой функции по срезовой дереву заключается в том, что среди n срезových сегментов выбирается $l < n$ опорных, которые сохраняются без изменений, после чего остальные срезových сегменты заменяются их интерполяцией (реконструкцией) по выбранным опорным

сегментам. Пусть при этом кусочно-постоянная интерполяция возможна между любыми двумя вершинами стекового графа, одна из которых находится по дереву выше другой вне зависимости от наличия между ними точек ветвления. При этом возникает возможность изменения структуры дерева. Проиллюстрируем такие структурные преобразования двумя простыми примерами уменьшающей интерполяции без сохранения топологии графа.

Первый пример (рис.3) демонстрирует возможность перемещения и «поглощения» (склеивания) узлов графа при монотонной интерполяции одномерной функции, сохраняющей стековые отношения между сегментами. В данном примере три различных узла исходного дерева (красный граф), расположенные на разных уровнях структуры, «сползают» вниз по дереву и объединяются в один узел выходного дерева (синий граф) с большим количеством дочерних ветвей. Второй пример (рис.4) заключается в отсечении концевой ветви графа, то есть «выступа» исходной функции, в результате которого узел исходного графа перестает быть узлом.

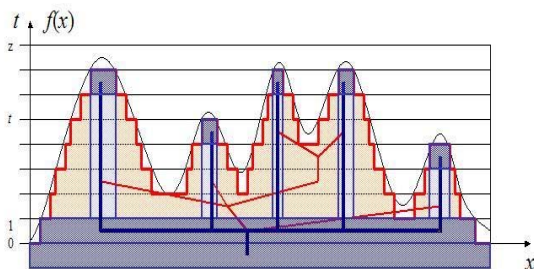


Рис.3. Пример преобразования стекового дерева. Изменение узлов.

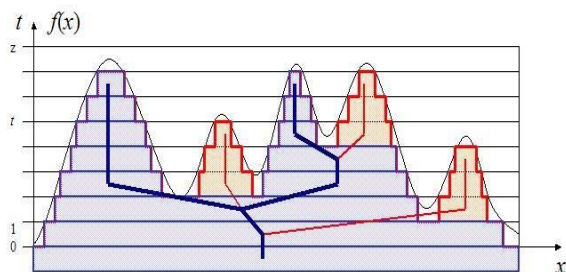


Рис.4. Пример преобразования стекового дерева. Удаление ветвей.

Определим оператор морфологического ДП-открытия как задачу нахождения такого *минимального набора опорных сегментов*, который обеспечивает *максимум нормы L^1* (площади под графиком) *интерполированной по этим сегментам кусочно-постоянной функции*, причем график функции-решения *целиком покрывается* графиком исходной функции, т.е. $h(x) \leq f(x)$ (рис.3,4).

Двойственный оператор ДП-закрытия на базе стекового дерева φ_α может быть определен через оператор ДП-открытия ψ_α и двойное инвертирование функции:

$$\varphi_\alpha f(x) = z - \psi_\alpha(z - f(x)).$$

4. Морфологическая сегментация изображений земной поверхности

Описанная выше техника морфологической сегментации обрабатывалась и тестировалась в задачах описания изображений земной поверхности, получаемых при съемке с космических и авиационных носителей. В качестве критериев J и Q использовались L^1 норма и количество опорных областей соответственно.

При этом оценивались компактность получаемого описания и точность реконструкции исходного изображения по этому описанию. В таблице приведены примеры ДП-открытия и ДП-закрытия для нескольких реальных изображений земной поверхности (Рис. 5-9).

Таблица 1. Зависимость количества срезовых сегментов от величины моделирующего параметра.

Параметр α	Im_1 (рис 5а) 548x408	Im_2 (рис 5б) 697x539	Im_1 548x408	Im_2 697x539
	ДП-открытие		ДП-закрытие	
0	2358	3352	2358	3352
10	1044	1487	1266	2102
100	520	719	588	902
1000	88	167	118	202
10000	21	35	28	46

Как видно из приведенных примеров даже при небольшом количестве опорных сегментов стековые деревья позволяют достаточно точно описать изображения земной поверхности исследуемого типа.

5. Заключение

Предложен эффективный алгоритм критериальной морфологической фильтрации для решения задач сегментации изображений. Особенностью алгоритма является представление изображения в виде стекового дерева, что позволило использовать метод динамического программирования для получения оптимального в некотором смысле описания изображения. Результаты тестирования подтвердили адекватность полученного описания.

Данный этап работы был посвящен решению задачи критериальной морфологической сегментации и последующей реконструкции типовых изображений земной поверхности. На следующих этапах исследования планируется разработать быстрые алгоритмы привязки фрагментов разномоментных

изображений фрагментов земной поверхности с использованием стековых древовидных описаний.

6. Литература

1. Визильтер Ю.В. Обобщенная проективная морфология. // Компьютерная оптика, Том 32, №4. 2008.
2. Визильтер Ю.В., Желтов С.Ю., Ларетина Н.А. Проективные морфологии на базе операторов фильтрации и сегментации изображений, вычисляемых методом динамического программирования // Вестник компьютерных и информационных технологий. 2009. №6.
3. Werner T. A Linear Programming Approach to Max-sum problem: A Review. Research reports of CMP. Czech Technical University in Prague, No 25, 2005. 46p.
4. Serra J. Image Analysis and Mathematical Morphology, Academic Press, London, 1982.
5. Yeong-Chyang Shih F., Mitchell O.R. Threshold Decomposition of gray Scale Morphology into Binary Morphology. IEEE trans. on pattern analysis, machine intelligence, vol, II, No 1, January 1989.

Информация об авторах

Визильтер Юрий Валентинович, доктор физико-математических наук, начальник лаборатории компьютерного машинного зрения ГосНИИАС.
Email: viz@gosniias.ru

Горбачевич Владимир Сергеевич, инженер лаборатории компьютерного машинного зрения ГосНИИАС.
Email: gvs@gosniias.ru

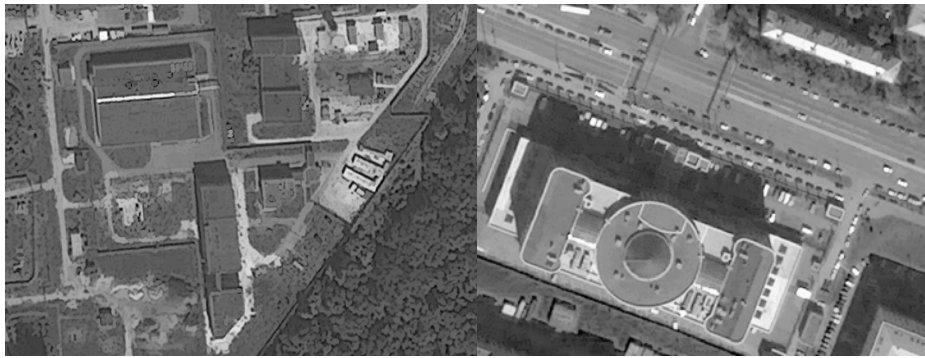


Рис.5. Исходные изображения: а) количество сегментов 2358 б) количество сегментов 3352.



Рис.6. ДП-открытие: а) $\alpha=1000$ (количество сегментов 88), б) $\alpha=10000$ (количество сегментов 21).

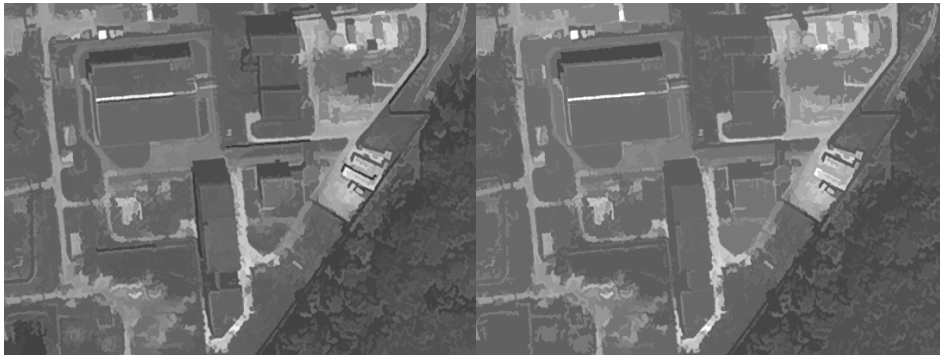


Рис.7. ДП-закрытие: а) $\alpha=1000$. (количество сегментов 118), б), $\alpha=10000$. (количество сегментов 28).



Рис.8. ДП-открытие: а) $\alpha=1000$. (количество сегментов 167), б), $\alpha=10000$. (количество сегментов 35).

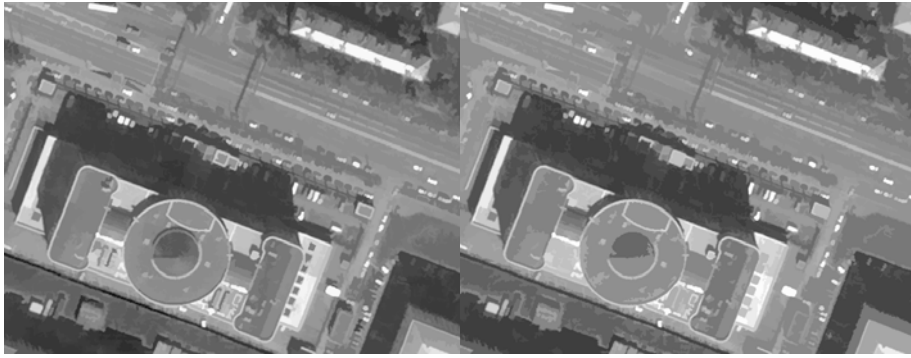


Рис.9.ДП-закрытие: а) $\alpha=1000$. (количество сегментов 202), б), $\alpha=10000$. (количество сегментов 46).

Агломеративный метод сегментации иероглифического текста

Кугаевских А.В.

Институт математики и компьютерных наук
Тюменский государственный университет, Тюмень, Россия
a-kugaevskikh@yandex.ru

Аннотация

В работе предложен новый алгоритм сегментации, основанный на агломеративном подходе. Основной принцип состоит в построении пирамиды сегментации, каждый уровень которой представлен в виде неориентированного графа. На первом уровне вершинами графа являются пиксели изображения, на втором и последующих – выделенные сегменты. Также строятся деревья вложенности сегментов. Верность сегментации проверяется подсистемой распознавания на каждом уровне, при этом сокращается количество непроверенных сегментов и корректируются деревья вложенности.

Ключевые слова: сегментация текста, агломеративная кластеризация, пирамида сегментации, деревья вложенности.

1. ВВЕДЕНИЕ

Сегментация текста, написанного по правилам древнеегипетской грамматики, необычайно трудная задача. Древнеегипетский текст содержит около 7000 символов, часто отличающихся только несколькими штрихами, и осложнен такими элементами как группировки и картуши, к тому же, часто практиковалось написание текста с разными ориентациями. Дополнительную сложность вносит написание текста в вертикальных колонках. Поэтому необходимо проводить анализ иерархической структуры документа и распознавать не только символы, но и грамматические элементы структуры текста (колонки, картуши и т.д.). Пример текста показан на рисунке 1.



Рисунок 1: Исходное изображение

Особым моментом является тот или иной уровень повреждения оригинала, что затрудняет как восстановление структуры текста, так и успешное распознавание, содержащих его, символов. В отличие от OCR-систем, работающих с использующимися на данный момент языками, в данном случае мы не можем привлечь на помощь лингвистику. Так как нашего знания древнеегипетского языка не достаточно для создания действительно работающего лингвистического модуля, который исправлял бы ошибки модуля распознавания, как это сделано, например, в Abbyy FineReader.

На данный момент разработано достаточно большое количество разнообразных подходов. Среди них нахождение границ через обнаружение разрывов яркости, выращивание областей, слияние и разделение регионов, метод морфологических водоразделов [1-2]. В последнее время активно развиваются методы, основанные на применении теории графов и оптимизационного подхода [3]. Общая идея методов, основанных на теории графов, следующая. Изображение представляется в виде взвешенного графа, с вершинами в точках изображения. Вес ребра графа отражает сходство точек в некотором смысле (расстояние между точками по некоторой метрике). Среди всех методов, основанных на теории графов, особо хочется отметить метод SWA [4-6]. Основная идея метода состоит в построении пирамиды взвешенных графов, каждый из которых получен из предыдущего путем объединения схожих вершин. На каждом шаге веса связей пересчитываются. В процессе построения пирамиды вычисляются различные статистики, характеризующие форму, цвет, текстуру регионов, эти статистики используются для вычисления меры сходства регионов. Затем, следуя идеологии методов теории графов, для полученного графа вводится функционал стоимости разреза и ищется разрез минимальной стоимости. На рисунке 2 показан результат применения метода SWA к изображению на рисунке 1 (слева) и правильная сегментация, выполненная вручную (справа). Как видно, сегментация изображения явно ошибочна.



Рисунок 2: Результат работы метода SWA (слева) и ручная сегментация (справа)

Подсистема сегментации должна формировать гипотезы о сегментах, содержащие предполагаемые границы символов или их элементов, и передавать их на подсистему распознавания для проверки правильности сегментирования.

2. ОБЩИЙ АЛГОРИТМ

Предлагаемый в данной работе метод является вариацией на тему агломеративной кластеризации пикселей [2]. Каждому пикселю ставится метка соответствия одному из классов (шум, фон, символ или его элемент). Для получения лучшей сегментации подобные в некоторой метрике пиксели рекурсивно сливаются. При этом предлагается многоуровневый алгоритм слияния токенов (заполнителей, указывающих на наличие в изображении искомой структуры), с проверкой правильности слияния на каждом уровне. На первом уровне токенами являются пиксели, являющиеся предполагаемыми граничными элементами сегментов, а на последующих – выделенные сегменты изображения.

Общий алгоритм сегментации таков:

1. Подготовительный этап
 - a. Для каждого пикселя изображения вычисляется уровень яркости
 - b. К матрице яркостей применяется фильтр Собеля для расчета уровня градиента
 - c. По матрице значений градиента вычисляется порог разделения фона и контуров объектов с помощью метода Отсу
2. Построение 1-ого уровня пирамиды сегментации
 - a. Построение неориентированного взвешенного графа по изображению
 - b. Агломеративная кластеризация пикселей
3. Построение последующих уровней пирамиды сегментации
 - a. Построение неориентированного взвешенного графа по набору сегментов
 - b. Построение деревьев вложенности сегментов
 - c. Агломеративная кластеризация сегментов

3. ПОДГОТОВИТЕЛЬНЫЙ ЭТАП

Методы поиска границ хорошо разработаны для полутоновых изображений. Полутоновое изображение рассматривается как функция двух переменных (x и y), и предполагается, что границы регионов соответствуют максимумам градиента этой функции. Для их поиска применяется аппарат дифференциальной геометрии (в простейшем случае это фильтры Робертса, Превитта, Собеля, лапласиан), позволяющий найти дискретное приближение двумерного градиента. Из приведенных фильтров, применение фильтра Собеля дает наименьший шум на изображении, поэтому именно он был выбран для вычисления градиента. На рисунке 3 показано применение фильтра Собеля к изображению текста.

Фильтр Собеля [7] основан на свёртывании изображения небольшими раздельными целочисленными фильтрами в вертикальном и горизонтальном направлениях. Фильтр использует ядра 3×3 , с которыми свёртывают оригинальное изображение к вычислению приближений производных по горизонтали и по вертикали.



Рисунок 3: Фильтр Собеля

Пусть A исходное изображение, а G_x и G_y — два изображения, где каждая точка содержит приближенные производные по x и по y . Они вычисляются следующим образом:

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A; \quad G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * A \quad (1)$$

где $*$ обозначает двумерную операцию свертки.

Координата x здесь определена возрастающей «направо», а y — «вниз». В каждой точке изображения полученное приближение градиента состоит из двух величин:

$$Gr = \sqrt{G_x^2 + G_y^2} \quad (2)$$

К матрице Gr применяется метод бинаризации Отсу [8] для получения порога разделения границы и фона. Алгоритм предполагает наличие в изображении двух классов пикселей и ищет оптимальный порог, разделяющий эти два класса так, что бы их внутриклассовая дисперсия была минимальна. Отсу показал, что минимизация девиации внутри класса это тоже самое, что и максимизация девиации между классами:

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = w_1(t)w_2(t)[\mu_1(t) - \mu_2(t)]^2 \quad (3)$$

которая выражается в терминах вероятности w_i и среднее арифметическое класса μ_i , которое в свою очередь может обновляться итеративно. Эта идея привела к эффективному алгоритму:

Строится гистограмма по значениям $p_i = n_i/N$, где N — это общее кол-во пикселей на изображении, n_i — это кол-во пикселей с уровнем градиента i . Диапазон градиентов делится на два класса с помощью порогового значения уровня градиента $t \in [0 \dots L]$. Каждому классу соответствуют относительные частоты:

$$w_1(t) = \sum_{i=1}^t p_i \quad (4)$$

$$w_2(t) = 1 - w_1(t) \quad (5)$$

Средние уровни для каждого из двух классов изображений:

$$\mu_1(t) = \sum_{i=1}^t \frac{ip_i}{w_1(t)} \quad (6)$$

$$\mu_2(t) = \sum_{i=t+1}^L \frac{ip_i}{w_2(t)} \quad (7)$$

Идея алгоритма состоит в максимизации межклассовой дисперсии $\sigma_b^2(t)$, вычисленной по формуле (3).

Таким образом, мы имеем две характеристики, отличающие соседние пиксели, это яркость пикселя и степень принадлежности к границе, выраженная значением градиента. Применение этих характеристик позволит нам построить контур сегмента.

4. ПОСТРОЕНИЕ ПИРАМИДЫ СЕГМЕНТАЦИИ

На первом уровне пирамиды сегментации исходное изображение представляется в виде неориентированного взвешенного графа $G_1 = (V, E)$, где вершинами V являются пиксели, соединенные друг с другом по принципу 8-смежности [1], т.е. со всеми пикселями-соседями, ребрами E , показывающими степень похожести пикселей. В качестве метрики похожести пикселей предложена формула (8), полученная эмпирическим путем:

$$w_{ij} = Gr_j |L_j - L_i| \quad (8)$$

где L_i - значение яркости текущего пикселя,

Gr_j - значение градиента соседнего пикселя,

L_j - значение яркости соседнего пикселя.

Для поиска сегмента применяется алгоритм Дейкстры, с той лишь разницей, что начальная и конечная вершины совпадают.

1. Ищется пиксель с максимальным значением градиента Gr
2. Вершина графа, соответствующая данному пикселю, становится начальной вершиной
3. Выполняется алгоритм Дейкстры для поиска замкнутого контура
4. Шаги 2-3 повторяются для всех оставшихся пикселей, значение градиента которых выше порога t , полученного по методу Отсу (формула 3).

Выход первого уровня характеризуется множеством найденных сегментов $S = [s_1, s_2, \dots, s_k]$. На рисунке 4 схематично показана часть сегментов, найденных на изображении с рисунка 1. Система строит сегменты по найденным контурам и, в идеале, они точно повторяют границы символа. Найденные области могут принадлежать символу, элементу символа или шуму. Они передаются на следующий уровень $l \in [2, \infty)$, который представляется в виде полного неориентированного взвешенного графа $G_l = (S, E)$, где вершинами являются найденные сегменты, соединенные друг с другом ребрами E , показывающими близость пары сегментов. Веса ребер являются w_{ij} - минимальное евклидово расстояние между точками токенов i и j .



Рисунок 4: Сегментация изображения

На втором уровне строятся также деревья вложенности сегментов, которые корректируются на последующих уровнях, рисунок 5. Деревья сегментов позволяют решить проблему с несколькими контурами одного и того же символа, как на рисунке 1, и играют главную роль при восстановлении структуры текста.

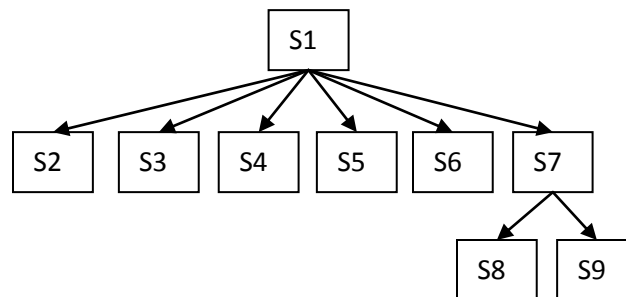


Рисунок 5: Дерево вложенности сегментов

На основании весов графов составляется список гипотез сегментации. Сегмент с наибольшим периметром становится опорным. Список гипотез составляется путем объединения токенов, при этом кандидаты на объединение берутся только при выполнении условия $w_{ij} \leq 3x$, где x - максимальное значение высоты или ширины токена. Каждая гипотеза об объединении токенов проверяется подсистемой распознавания. В результате для каждой гипотезы дается вероятность ее правдоподобия. Гипотеза, имеющая максимальную вероятность, признается истинной. Остальные токены передаются на следующий уровень, при этом множество S сокращается и корректируется дерево вложенности сегментов. Формирование уровней пирамиды и их проверка подсистемой распознавания продолжается до устранения неопределенности.

В случае невозможности классифицировать сегмент, возможно, имеет место недостаточная сегментация. При этом в список гипотез включается гипотеза о недостаточной

сегментации объекта. В этом случае сегмент делится на объекты, и к каждому из них применяется процедура распознавания. Предполагается, что точка сегментации лежит в области геометрического центра токена. В ней проводится линия, соответствующая наклону почерка сверху-вниз или справа-налево в зависимости от отношения вертикального и горизонтального размеров прямоугольника, описывающего токен. Полученные в результате распознавания данные сравниваются с остальными гипотезами. Процедура разбиения также носит иерархический характер и продолжается до устранения неопределенности.

Если символ распознан, но недостаточно уверенно (нормированный выход системы распознавания по этому классу от 50 до 70%), выделяется структура признаков распознанного символа. Каждый символ можно представить в виде неориентированного невзвешенного графа, элементами которого будут характерные линии, углы или пересечения линий, рисунок 6. Если в подобную структуру подставить данные по выходам подсистемы выделения признаков, можно проверить наличие опознанных признаков и, при необходимости, откорректировать гипотезы. Эти же признаки играют роль при обучении подсистемы распознавания. Таким образом, на основании данных распознавания признаков можно получить информацию об относительном расположении недостающих сегментов и, произведя их поиск на графе токенов, дополнить сегмент и увеличить вероятность верной сегментации и классификации.

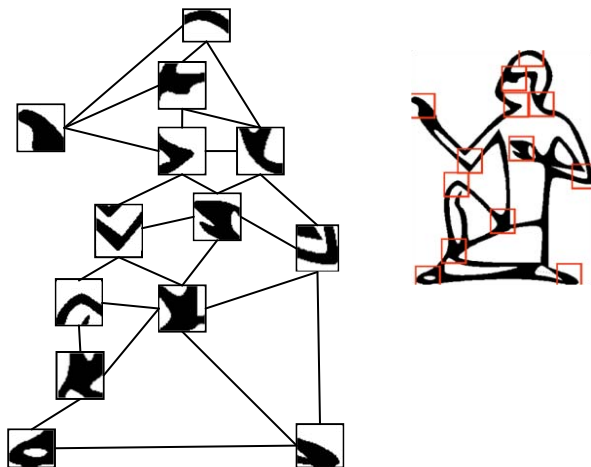


Рисунок 6: Структура признаков

5. ПРОВЕРКА СЕГМЕНТАЦИИ

Проверку гипотез можно возложить на формальные искусственные нейронные сети, хорошо зарекомендовавшие себя в решении задач распознавания образов. Из всего многообразия ИНС выгодно отличается неокогнитрон Фукушимы [9]. Данная сеть обладает требуемым уровнем инвариантности к сдвигу и деформации образов. Структура сети изображена на рисунке 7. Сеть делится на слои, слои на плоскости, плоскости на отдельные ячейки. Каждая плоскость слоя U_{C1} реагирует на изученные специфические признаки символа в определенном положении на изображении. Применительно к задаче сегментации нас интересуют выходы слоя U_{C1} (извлечение признаков) и U_{C4} (результат распознавания образа). Сеть обучается с помощью конкурентного контролируемого обучения. Учитель подает символ и указывает его признаки, отмечая характерные

линии, углы, пересечения линий. Для каждого нового признака должна быть создана новая плоскость в слое U_{S1} . Нейрон, оказавшийся в центре рецептивного поля и попадающий на признак, дает максимальный выход. Таким образом, обеспечивается конкурентное обучение всей плоскости. Нейроны последующих слоев имеют рецептивные поля большего размера. S-ячейки промежуточных стадий (U_{S2} и U_{S3}) самоорганизованы при помощи неконтролируемого конкурентного обучения. Уровень U_{C4} является уровнем распознавания, реакция которого показывает конечный результат распознавания образов сетью. Уровень U_{S4} обучается через контролируемое конкурентное обучение.

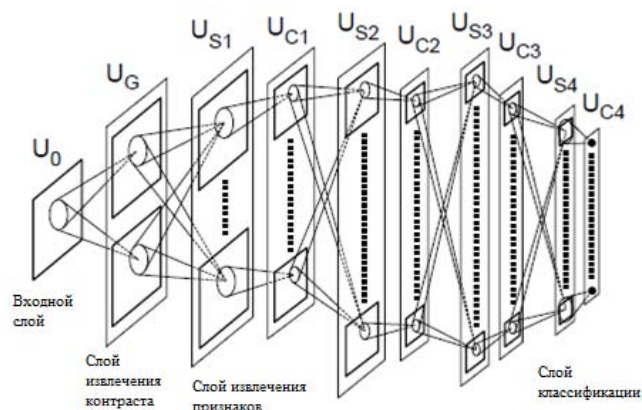


Рисунок 7: Структура неокогнитрона

Связи, ведущие к S-нейрону, имеют изменяемые весовые коэффициенты, получаемые в ходе обучения сети. В результате обучения S-нейрон активизируется только тогда, когда в его рецептивном поле оказывается именно тот признак образа, на который обучали данный нейрон. S-нейроны введены в сеть для того, чтобы уменьшить ошибку позиционирования. Каждый S-нейрон получает в качестве входа сигналы, поступающие от группы S-нейронов, которые выделяют одинаковый признак образа, но с небольшими смещениями в положении рецептивного поля. Функция активации S-нейрона такова, что S-нейрон переходит в активное состояние тогда, когда хотя бы один из этих S-нейронов находится в активном состоянии. Таким образом, S-нейрон распознает выделяемый элемент образа, даже если он немного сдвинут в своем положении, что обеспечивает инвариантность к сдвигам.

Каждый слой сети имеет более абстрактную, менее специфичную реакцию по сравнению с предшествующим; выходной слой реагирует на полные образы, показывая высокую степень независимости от их положения, размера и ориентации во входном поле. При этом для правильной работы классификатора, построенного на базе неокогнитрона, необходимо осуществлять предобработку изображения, подающегося на вход сети. У любой ИНС ограниченные размеры рецептивного поля, поэтому входное изображение необходимо масштабировать до необходимых размеров. С этой задачей очень хорошо справляется бикубическая интерполяция, обладая приемлемым качеством при высокой скорости работы алгоритма. Неокогнитрон налагает и свои требования: изображение должно быть бинарным, штрих символа должен обладать единичной толщиной. Бинаризовать изображение и получить штрих единичной толщины можно при векторизации изображения,

полученного путем слияния сегментов, находящихся на разных уровнях дерева вложенности. Внешний контур символа окажется на вышележащем уровне, тогда как внутренний контур – на нижележащем. Это изображение уже бинарно так как содержит только границы.

Примем выход фоторецепторных ячеек слоя U_0 за $u_0(n)$, где n показывает позицию ячейки. Выходом ячейки слоя U_G , где n - позиция центра рецептивного поля, является

$$u_G(n, k) = \varphi \left[-1^k \sum_{|v| < A_G} a_G(v) u_0(n + v) \right] \quad (9)$$

где $k=1,2$ и $\varphi [\]$ определяется как $\varphi [x] = \max(x; 0)$. Параметр $a_G(\xi)$ показывает силу фиксированной связи с ячейкой. Слой U_G состоит из двух плоскостей: одна содержит ячейки с оп-центром ($k=2$), другая ячейки с off-центром ($k=1$). A_G обозначает радиус диапазона суммирования v , т.е., размера пространственного распространения входных связей с ячейкой, что суть есть радиус рецептивного поля нейрона.

Входные связи с единственной ячейкой слоя U_G проектируются таким образом, чтобы их итоговая сумма была равна 0. Другими словами, связь $a_G(\xi)$ должна удовлетворять

$$\sum_{|v| < A_G} a_G(v) = 0 \quad (10)$$

где v - индекс связи, k - номер плоскости, n - индекс ячейки.

Для расчета массива параметров a_G чаще всего применяются вейвлеты.

Пусть $u_{S1}(n; k)$ и $u_{C1}(n; k)$ являются выходами S- и C-ячеек k -ой плоскости стадии 1, где n представляет позицию центра рецептивного поля ячейки, v - индекс связи, k - номер плоскости. Слой U_{S1} содержит также V-ячейки, выход которых определяется как $v_1(n)$. Выход S- и V-ячеек соответственно равен:

$$u_{S1}(n, k) = \frac{\theta_l}{1 - \theta_l} \varphi \left[\frac{1 + \sum_{\kappa=1}^{K_{Cl-1}} \sum_{|v| < A_{S1}} a_{S1}(v, \kappa, k) u_{Cl-1}(n + v, \kappa)}{1 + \theta_l b_{S1}(k) v_1(n)} - 1 \right] \quad (11)$$

$$v_1(n) = \sqrt{\sum_{\kappa=1}^{K_{Cl-1}} \sum_{|v| < A_{S1}} c_{S1}(v) \{u_{Cl-1}(n + v, \kappa)\}^2} \quad (12)$$

Параметр $a_{S1}(v, \kappa, k) (\geq 0)$ - сила переменной возбуждающей связи, идущей от C-ячейки $u_{Cl-1}(n + v, \kappa)$ предыдущей стадии. Следует отметить, что все ячейки плоскости совместно используют то же самое множество входных связей, следовательно $a_1(v, \kappa, k)$ не зависит от n . A_{S1} обозначает радиус рецептивного поля ячейки, т.е., размер пространственного сдвига входящего образа относительно центра рецептивного поля. Параметр $b_1(k) (\geq 0)$ - сила переменной тормозящей связи, идущей от V-ячейки. Параметр $c_{S1}(v)$ представляет силу постоянной возбуждающей связи с V-ячейкой, и является монотонно убывающей функцией от $|v|$. В данной работе он вычисляется также с помощью вейвлета «сомбреро», но берутся только положительные значения.

Положительная константа $\theta_l (< 1)$ является порогом чувствительности S-ячейки и определяет степень селективности в извлечении признаков.

В (3) и (4) для $l = 1$, $u_{Cl-1}(n; k)$ заменяется на $u_C(n; k)$, и мы имеем $K_{Cl-1} = 2$, где K - количество плоскостей.

Каждой S-ячейке соответствует своя V-ячейка.

Схематично связи между ячейками можно представить таким образом как они отображены на рисунке 8.

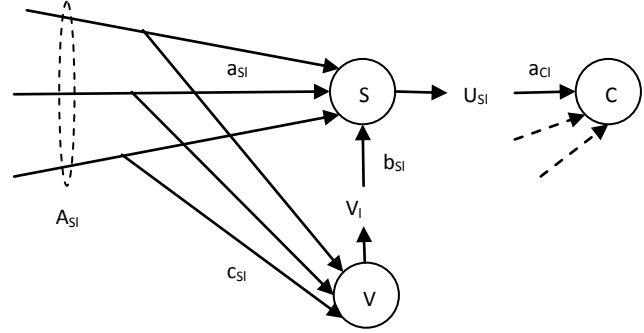


Рисунок 8: Схема связей между ячейками в слое

Задачей комплексных ячеек является уменьшение зависимости реакции системы от позиции образов во входном поле. Для достижения этого каждый комплексный узел получает в качестве входного образа выходы набора простых узлов из соответствующей плоскости того же слоя. Эти простые узлы покрывают непрерывную область простой плоскости, называемую рецептивной областью комплексного узла. Возбуждение любого простого узла в этой области является достаточным для возбуждения данного комплексного узла. Таким образом, комплексный узел реагирует на тот же образ, что и простые узлы в соответствующей ему плоскости, но он менее чувствителен к позиции образа, чем любой из них.

Таким образом, каждый слой комплексных узлов реагирует на более широкую область входного образа, чем это делалось в предшествующих слоях. Эта прогрессия возрастает линейно от слоя к слою, приводя к требуемому уменьшению позиционной чувствительности системы в целом.

Ответ S-ячейки уровня U_{S1} пространственно размыт в последующих C-ячейках уровня U_{C1} . Математическим ответом S-ячейки U_{C1} , исключая самую высокую стадию U_{C4} , является

$$u_{C1}(n, k) = \psi \left[\sum_{|v| < A_{C1}} a_{C1}(v) u_{S1-1}(n + v, \kappa) \right] \quad (13)$$

где $\psi[x] = \frac{\varphi[x]}{1 + \varphi[x]}$. Параметр $a_{C1}(v)$ показывает силу постоянных возбуждающих связей, сходящихся от группы S-ячеек, которые распространяются в пределах радиуса A_{C1} .

Плоскость C-ячеек соединена только с соответствующей плоскостью S-ячеек.

Обучение сети выполняется от более низких стадий к более высоким: после завершения обучения более низкой стадии, начинается обучение следующей. То же самое множество обучающих образов используется для обучения всех стадий, кроме U_{S1} .

Хотя метод выбора начальных параметров ячеек во время обучения немного отличается в разных слоях, правило усиления изменяющихся связей $a_1(v, \kappa, k)$ и $b_1(k)$ одинаково для всех слоев.

Сначала выбирается простой узел, веса синапсов которого должны быть увеличены, он рассматривается как представитель всех узлов в плоскости, вызывая увеличение их связей на том же самом образе. Это делается каждый раз, когда представляется новый обучающий образ и для каждой плоскости. Представители выбираются из тех S-ячеек, которые дают большой выход, но число представителей ограничивается так, чтобы был не более чем один представитель от любой одной S-плоскости. Таким образом, все узлы в плоскости обучаются распознавать одни и те же свойства, и после обучения будут делать это независимо от позиции образа в поле комплексных узлов в предшествующем слое.

Для представителя S-ячейки усиливаются только входные соединения, через которые приходят не равные нулю сигналы. С этой процедурой представитель S-ячейки становится выборочно отзывчивым только на те характеристики стимула, который представлен. Все другие S-ячейки в S-плоскости, от которых выбран представитель, имеют такое же усиление своих входных соединений, как и их представитель. Эти отношения могут количественно быть выражены следующим образом.

Пусть ячейка $u_{SI}(\hat{n}, \hat{k})$ была выбрана в качестве начальной, следовательно, изменяющиеся связи $a_i(v, \kappa, \hat{k})$ с этой ячейкой, как и со всеми S-ячейками в этой плоскости, увеличиваются следующим образом:

$$\Delta a_i(v, \kappa, \hat{k}) = q_1 c_{SI}(v) u_{CI-1}(\hat{n} + v, \kappa) \quad (14)$$

где q_1 - положительная константа, определяющая скорость обучения. Хотя было предложено много алгоритмов определения тормозящей связи $b_i(\hat{k})$, мы используем метод, в котором $b_i(\hat{k})$ определяется исходя непосредственно из возбуждающих связей $a_i(v, \kappa, \hat{k})$, следующим образом:

$$b_{SI}(\hat{k}) = \sqrt{\sum_{\kappa=1}^{K_{CI-1}} \sum_{|v| < A_{SI}} \frac{\{a_{SI}(v, \kappa, \hat{k})\}^2}{c_{SI}(v)}} \quad (15)$$

Однако в ячейках S-плоскости, от которых не выбран представитель, входные соединения не изменяются. Выбор начальных величин изменяемых соединений имеет небольшой эффект для надежности неокогнитрона, при условии, что они невелики и определены таким способом, что каждая S-плоскость имеет другой ряд начальных величин для своих входных соединений.

S-ячейки промежуточных уровней (U_{S2} и U_{S3}) самоорганизованы при помощи неконтролируемого конкурентного обучения, применяемого в стандартном неокогнитроне. Начальные ячейки определяются принципом "Победитель получает все" (WTA). Каждый раз, когда обучающий образ подается на входной слой, каждая S-ячейка конкурирует с другими ячейками поблизости в области, называемой областью конкуренции. Если выход ячейки больше остальных в области конкуренции, ячейка выбирается в качестве начальной, при этом в плоскости может и не быть выбран ни один представитель. Как можно увидеть по формуле (14), каждая входная связь начальной ячейки увеличивается на число пропорциональное ответу ячейки, от которой идет связь. Из-за совместного использования связей в пределах каждой плоскости, все ячейки плоскости имеют то же множество входных связей, что и начальная ячейка.

Так как представители определяются этим способом, каждая S-плоскость становится выборочно чувствительной на одну из характеристик обучающих образов и нет возможности образования излишних соединений так, чтобы две или более S-плоскостей использовались для обнаружения одной и той же характеристики. Случайно представители выбираются только из небольшого количества S-плоскостей за один раз, остальная часть S-плоскостей производит представителей, когда представляются другие стимулирующие образцы. S-ячейки самой высокой стадии (U_{S4}) обучаются при помощи контролируемого конкурентного обучения. Правило обучения напоминает конкурентное обучение, использованное в слоях U_{S2} и U_{S3} , однако здесь также используются имена классов обучающих образов. Когда сеть изучает варианты деформированных обучающих образов через конкурентное обучение, более чем одна плоскость для одного класса обычно создается в U_{S4} . Поэтому каждая плоскость изучает обучающий образ и имя класса.

По результатам моделирования и серии экспериментов получились следующие параметры сети:

1. Размер слоя U_0 : 65*65
2. Размер плоскости слоя U_G : 71*71, a_G 7*7, $A_G=3.3$, $A_{g0}=1.2$
3. Размер плоскости слоя U_{S1} : 68*68, количество плоскостей 47915, a_{S1} 6*6, $\theta_1 = 0.55$
4. Размер плоскости слоя U_{C1} : 37*37, количество плоскостей 47915, a_{C1} 18*18
5. Размер плоскости слоя U_{S2} : 38*38, количество плоскостей 95850, a_{S2} 6*6, $\theta_2 = 0.51$
6. Размер плоскости слоя U_{C2} : 21*21, количество плоскостей 95850, a_{C2} 14*14
7. Размер плоскости слоя U_{S3} : 22*22, количество плоскостей 143725, a_{S3} 6*6, $\theta_3 = 0.58$
8. Размер плоскости слоя U_{C3} : 13*13, количество плоскостей 143725, a_{C3} 8*8, $A_{C3}=5$
9. Размер плоскости слоя U_{S4} : 5*5, количество плоскостей 142785, a_{S4} 9*9, $\theta_4 = 0.30$
10. Размер плоскости слоя U_{C4} : 1*1, количество плоскостей 6845, a_{C4} 5*5

На обучающей выборке из 6845 символов сеть показала качество распознавания в 99%, на тестовой выборке из 27380 символов, представляющих те же самые обучающие образы с повреждениями в каждой из четвертей символа, сеть показала результат в 86% [10]. Тестовая выборка создавалась путем искажения символов из обучающей выборки в каждой из четвертей квадрата символа.

6. ЗАКЛЮЧЕНИЕ

Построение пирамиды и проверка правильности сегментации на каждом уровне позволяет сократить количество ошибочно выделенных сегментов и, как следствие, повысить качество распознавания символов или элементов текста. Деревья вложенности сегментов не только необходимы для восстановления структуры текста, но и, участвуя в распознавании, также позволяют повысить его качество, дополняя информацию о сегменте. Обратные связи от подсистемы распознавания к подсистеме сегментации позволяют сократить количество проверяемых гипотез и улучшить качество сегментации.

7. СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений.- М.: Техносфера, 2005
2. Форсайт Д., Понс Ж. Компьютерное зрение. Современный подход.: Пер. с англ.- М.: Издательский дом «Вильямс», 2004
3. Вежнев А., Барина О. Методы сегментации изображений: автоматическая сегментация. Компьютерная графика и мультимедиа, 4(4), 2006. <http://cgm.computergraphics.ru/content/view/147>
4. Sharon E., Brandt A., Basriy R. Fast Multiscale Image Segmentation. CVPR 2000
5. Sharon E., Brandt A., Basriy R. Segmentation and Boundary Detection Using Multiscale Intensity Measurements. CVPR 2001
6. Galun M., Sharon E., Brandt A., Basriy R. Texture Segmentation by Multiscale Aggregation of Filter Responses and Shape Elements. ICCV 2003
7. Дуда Р., Харт. П. Распознавание образов и анализ сцен. Пер. с англ. – М.: Мир, 1976.
8. Otsu N. A threshold selection method from gray-level histograms. IEEE Trans. Sys., Man., Cyber. 9: 62-66, 1979.
9. Fukushima K. Neocognitron for handwritten digit recognition, Neurocomputing, 51, pp. 161-180, 2003.
10. Кугаевских А.В., Ивашко А.Г. Возможность применения искусственных нейронных сетей "неокогнитрон" для распознавания древнеегипетских иероглифов // Вестник Тюменского Государственного Университета, 2009, №6, с. 209-214.

Об авторе

Кугаевских Александр Владимирович, аспирант Тюменского государственного университета, факультет математики и компьютерных наук. Контактный e-mail: a-kugaevskikh@yandex.ru.

Инструментальная реализация алгоритмов реконструкции искаженных изображений

Кирьянов К.А.

Санкт-Петербургский государственный университет
информационных технологий, механики и оптики, Санкт-Петербург
e-mail: kiryancon@front.ru

Аннотация

В работе рассматриваются особенности инструментальной реализации алгоритмов реконструкции искаженных (смазанных, дефокусированных или/и зашумленных) изображений [1, 2, 5, 6]. Рассмотрена прямая задача (моделирование искажения или экспериментальное получение реального искаженного изображения) и более сложная и важная для практики обратная задача (реконструкция, восстановление изображения) [1–3, 5, 6].

Ключевые слова: искажение изображения, прямая и обратная задачи, основные соотношения

1. Реализуемые алгоритмы

Прежде чем вести речь о технической реализации алгоритмов, необходимо сказать несколько слов о самих алгоритмах. Основные соотношения и уравнения, описывающие эти задачи в непрерывном виде, имеют вид [5, 6]:

$$\int_{-\infty}^{\infty} h(x-\xi) w_y(\xi) d\xi = g_y(x) + \delta g, \quad (1)$$

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x-\xi, y-\eta) w(\xi, \eta) d\xi d\eta = g(x, y) + \delta g, \quad (2)$$

где h – функция рассеяния точки (ФРТ), в большинстве случаев являющаяся пространственно-инвариантной, w и g – распределение интенсивности по неискаженному и искаженному изображениям соответственно, δg – помеха. В (1) ось x направлена вдоль смаза, а y играет роль параметра.

Интегральное уравнение (ИУ) (1) (точнее, набор 1-мерных интегральных уравнений) обычно используется в задаче смазывания, а (2) – в задаче дефокусирования. Наиболее подходящим методом решения интегральных уравнений (1) и (2) является метод регуляризации Тихонова [2–6]. Так как для инструментальной реализации интерес представляют численные методы, в данной работе им и уделено внимание. При численной реализации метода регуляризации Тихонова для решения обратной задачи используются преобразование Фурье (ПФ) и метод квадратур. Из методов итераций следовало бы упомянуть метод простой итерации Фридмана.

Метод квадратур. Данный метод эффективен для решения набора 1-мерных ИУ (1) [10], каждое из которых можно записать в операторной форме:

$$Aw = g + \delta g, \quad (3)$$

где A – интегральный оператор. Для нахождения функции w необходимо построить такой оператор A^{-1} , который давал бы устойчивое приближение к w . Непрерывным уравнениям (1) можно поставить в соответствие дискретные выражения типа (3). Тогда выражение (3) становится системой линейных алгебраических уравнений (СЛАУ), где A – матрица, связанная с ФРТ. Однако задача решения уравнений (1)–(3) является *некорректной*.

В методе регуляризации Тихонова вместо (3) решается:

$$(\alpha E + A^T A) w_\alpha = A^T g, \quad (4)$$

где $\alpha > 0$ – параметр регуляризации, E – единичный оператор (единичная матрица). Матрица $\alpha E + A^T A$ является квадратной, симметричной и положительно-определенной, СЛАУ имеет решение и численно устойчива [11]. В результате получаем решение в виде:

$$w_\alpha = (\alpha E + A^T A)^{-1} A^T g, \quad (5)$$

Способы выбора α изложены, например, в [2–6].

Метод регуляризации Тихонова с использованием преобразования Фурье. Этот метод применим для решения уравнений типа свертки. Рассмотрим частный случай интегрального уравнения Фредгольма I рода – уравнение типа свертки 1-мерное (1) и 2-мерное (2). Если уравнение общего вида (1) при его численном решении методом квадратур требует размещения в компьютерной памяти матрицы СЛАУ и это ограничивает возможности метода, то для решения 1-мерного уравнения типа свертки возможно применение метода преобразования Фурье, оперирующего лишь с векторами, что существенно расширяет возможности метода в отношении памяти и времени решения. Еще серьезнее дело с распределением памяти обстоит при аппаратной реализации метода. Сказанное в еще большей степени характерно для 2-мерного уравнения (2).

Задача смазывания сводится к решению 1-мерного ИУ Фредгольма I рода типа свертки (1) относительно $w_y(\xi)$ при каждом фиксированном значении y , играющем роль параметра. В уравнении (1) ФРТ выражается формулой [3, 5, 6]:

$$h(x) = \begin{cases} 1/\Delta, & -\Delta \leq x \leq 0, \\ 0, & \text{иначе.} \end{cases} \quad (6)$$

Решение уравнения (1) методом ПФ с регуляризацией Тихонова имеет вид [3, 5, 6]:

$$w_{\alpha y}(\xi) = \frac{1}{2\pi} \int_{-\infty}^{\infty} W_{\alpha y}(\omega) e^{-i\omega\xi} d\omega, \quad (7)$$

где

$$W_{\alpha y}(\omega) = \frac{H(-\omega)G_y(\omega)}{|H(\omega)|^2 + \alpha\omega^{2p}}, \quad (8)$$

$$H(\omega) = \int_{-\infty}^{\infty} h(x) e^{i\omega x} dx, \quad G_y(\omega) = \int_{-\infty}^{\infty} g_y(x) e^{i\omega x} dx, \quad (9)$$

где $p \geq 0$ – порядок регуляризации.

2. Особенности технической реализации изложенных алгоритмов

Постепенно отживающие системы аналогового электронного телевидения (ТВ) передают движущееся (постоянно изменяющееся) изображение как последовательность кадров.

Структурная схема устройства формирования аналогового сигнала представлена на рис. 1.

Вся элементная база для систем обработки изображений должна обладать высокой производительностью, о чем под-

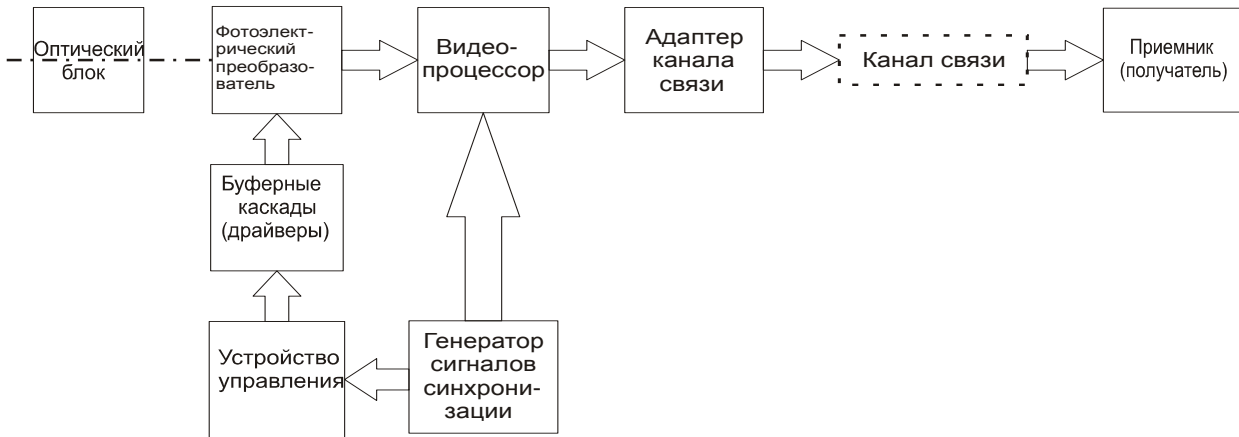


Рис.1. Структурная схема устройства формирования аналогового сигнала

Каждый кадр передается последовательно в виде двух полукадров. Неподвижное изображение в таких системах передается в виде многократного повторения одного и того же кадра. Особенности формирования ТВ-сигнала и используемые в них цветовые модели формируемых изображений изложены в [17, 18]. В таких системах практически невозможно восстановление смазанных и дефокусированных изображений.

Однако задачи (1), (2) довольно успешно решаются с помощью цифровых вычислительных средств. Устройство (или система) цифровой обработки сигналов изображений представляет собой комплекс технических средств, в состав которых входят ЭВМ, специализированные устройства для ввода, обработки и вывода сигнальной информации, а также соответствующее программное обеспечение. В общем случае подобная система помимо обработки должна также обеспечивать ввод, вывод и передачу сигналов и изображений [9]. Под вводом изображения понимаются процедуры преобразования исходного изображения к виду, удобному для его дальнейшей обработки в вычислительной системе. Ввод может производиться в нашем случае с вышеупомянутых телевизионных камер или ПЗС-линеек. Под выводом изображения понимается оперативная визуализация его на видеомониторе, а также архивация с целью долговременного хранения и документирования необходимой информации. Структурная схема такого устройства изображена на рис. 2.

робнее написано ниже.

Многие задачи обработки изображений также могут быть решены на современных персональных ЭВМ, если к скорости обработки не предъявляются высокие требования. В этом случае те или иные процедуры обработки изображений на ПЭВМ реализуются путем создания специального программного обеспечения, в которых ввод изображения в реальном масштабе времени выполняется специализированными устройствами ввода, пример структурной схемы которого приведен на рис. 3. Для обработки сигнальной информации в реальном масштабе времени требуется производительность, превышающая производительность современных ПЭВМ.

Одним из важнейших требований к техническим средствам систем ЦОС является обеспечение режима обработки сигналов в реальном времени (в темпе их поступления) при приемлемых стоимостных и весо-габаритных характеристиках системы. Такое требование, по сути, влечет за собой исключительно жесткие требования к быстродействию, прежде всего, средств предварительной обработки сигналов, что связано с большой интенсивностью входного потока сигнала в системах ЦОС. Так, некоторые технические задачи цифровой обработки изображений (такие, как обнаружение и распознавание самолетов-нарушителей территориальных границ и пр.) требуют быстродействия аппаратных средств порядка $10^9 - 10^{10}$ оп/сек для обеспечения обработки в реальном времени. Другой особенностью подобных задач является необходимость обработки больших объемов (от 1 Мбайта и более) сложно структурированных многомерных данных, что вносит дополнительные трудности в организацию вычислений.

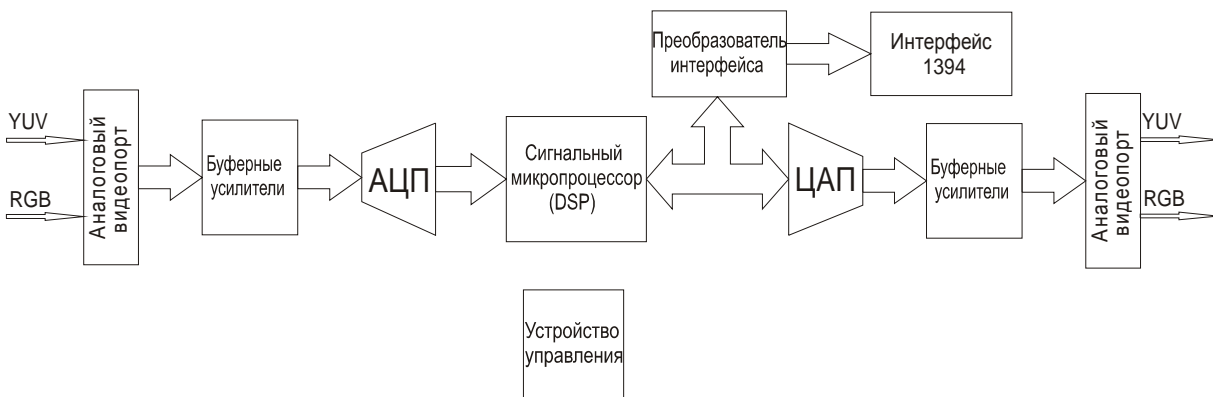


Рис. 2. Устройство цифровой обработки

Аналогово-цифровой преобразователь (АЦП) является важнейшим узлом, к быстродействию которого предъявляются особые требования. В системах для обработки изображений применяются самые быстрые на сегодняшний день – конвейерные (pipelined) АЦП, частота дискретизации (выборки) которых сегодня достигает единиц ГГц (GSPS) [12–14]. При таком быстродействии на сегодняшний день они разработаны с разрядностью до 16 бит.

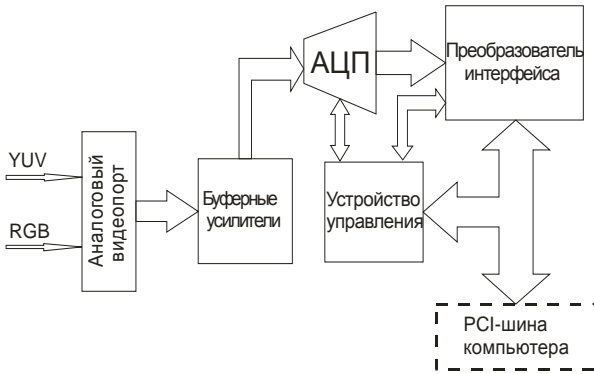


Рис. 3. Устройство ввода изображения в компьютер

При выборе вычислительного устройства, которое осуществляет цифровую обработку (DSP – Digital Signal Processing), существует два основных подхода при реализации систем обработки изображений.

Первый подход состоит в том, что для реализации того или иного вида обработки изображений на базе программируемой логической интегральной схемы (ПЛИС) реализуется конечный автомат, способный выполнять только одну функцию. При этом достигается максимально-возможная производительность, которая определяется используемой разработчиком и производителем ПЛИС технологией изготовления (ALTERA, XILINX), но гибкость решения присутствует лишь в той мере, в которой ее предусмотрел разработчик. Для задачи реконструкции изображений при решении методом квадратур необходимо правильно сформировать матрицу A , произвести ее транспонирование A^T , получить уравнение (4), перемножив транспонированную матрицу на исходную и на правую часть, и решить уравнение Тихонова (5), которое может быть решено методом Гаусса-Жордана, и схема такого устройства предложена в [8]. Сложность здесь только в том, что размерность матрицы нельзя сделать больше определенной величины, которая закладывается при разработке.

Точно также можно реализовать решение и для метода регуляризации Тихонова с преобразованием Фурье. Последнее реализуется лучше всего для данной задачи с помощью многоканальной систолической матрицы, особенности реализации которой описаны в [7]

Второй подход заключается в использовании специализированных процессоров (DSP). Такой подход обеспечивает максимальную гибкость, так как обработка реализуется программными методами. Здесь оба метода, как и на персональной ЭВМ можно реализовать как на языке ассемблера для используемого микропроцессора, так и на языке программирования C++, что гораздо проще на практике. Алгоритм более гибкий, структура устройства не меняется, так как реализация сводится к пошаговому выполнению инструкций. Но в то же время производительность даже специализированного процессора всегда меньше, чем производительность

специализированного аппаратного блока, реализованного на ПЛИС. Особое внимание следует уделять выбору сигнального микропроцессора. Наиболее часто используемыми в обработке видео в реальном времени находят применение процессоры Texas Instruments, несколько реже Analog Devises [15, 16].

В настоящее время подход выбирается в зависимости от сложности алгоритма обработки и требований по временным параметрам. Довольно часто выбирают комбинированный подход, где присутствует как программируемая логика, так и цифровой сигнальный процессор.

В схемотехнике камер сложность цифрового блока определяется разрешающей способностью камеры (ПЗС-матрицы). На сегодняшний день пока не существует таких камер, где реализован алгоритм восстановления смазанных и дефокусированных изображений. Пример цифровой камеры приведен на рис. 4. В таком устройстве в принципе может быть реализован рассмотренный алгоритм.

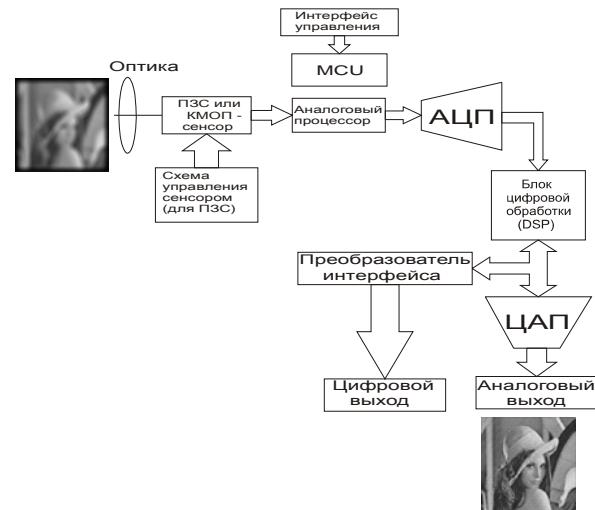


Рис 4. Структурная схема цифровой камеры

3. Выводы

Новизна выполненной работы состоит в инструментальной реализации новых приемов в обработке изображения: «усечение», «размытие краев» и «поворот» при использовании известных методов преобразования Фурье или квадратур с регуляризацией Тихонова. Прием усечения используется для того, чтоб избежать так называемых «граничных условий» [2]. Прием размытия краев используется для понижения эффекта Гиббса (эффекта ложных волн на изображении). Прием поворота используется для решения задачи смазывания при произвольном угле. Подробности этих новых приемов изложены в работе [6].

Все это может найти применение в таких областях, как томография (возникновение смазывания из-за случайных движений пациента во время обследования), обнаружение самолетов-нарушителей территориальных границ и т.п. Здесь необходима реализация методик на современной элементной базе, специализированной для обработки сигналов и адаптированной для быстрых алгоритмов. В процессе работы был проведен анализ имеющейся элементной базы и выяснилось, что более всего подходят для решения задачи компоненты фирмы Texas Instruments: процессор TMS320C6457, АЦП ADS5522 и ЦАП THS5661. Задача является актуальной на сегодняшний день.

Работа выполнена при поддержке РФФИ (грант № 09-08-00034-а).

Список литературы

1. Бейтс Р., Мак-Доннелл М. Восстановление и реконструкция изображений. М.: Мир, 1989. 336 с.
2. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М.: Техносфера, 2006. 1072 с.
3. Бакушинский А.Б., Гончарский А.В. Некорректные задачи. Численные методы и приложения. М.: Изд-во МГУ, 1989. 199 с.
4. Верлань А.Ф., Сизиков В.С. Интегральные уравнения: методы, алгоритмы, программы. Киев: Наук. думка, 1986.
5. Сизиков В.С. Математические методы обработки результатов измерений. СПб.: Политехника, 2001. 240 с.
6. Сизиков В.С., Римских М.В., Мирджамолов Р.К. Реконструкция смазанных и зашумленных изображений без использования граничных условий // Оптический журнал. 2007. Т. 76. № 5. С. 38–46.
7. Кухарев Г.А., Тропченко А.Ю., Шмерко В.П. Систематические процессоры для обработки сигналов. Минск: Беларусь, 1988. 127 с.
8. Кухарев Г.А., Тропченко А.Ю. Систематический процессор для обращения матриц // Изв. вузов. Приборостроение. 1990. Т. 33. № 11. С. 23–27.
9. Тропченко А.Ю. Аппаратные средства для цифровой обработки сигналов. Уч.-методическое пособие по дисциплине "Методы обработки сигналов и изображений". СПб.: Изд-во СПбГУ ИТМО, 2005. 138 с.
10. Поршнев С.В. Вычислительная математика. Курс лекций. СПб.: БХВ-Петербург, 2004. 304 с.
11. Киреев В.И., Пантелеев А.В. Численные методы в примерах и задачах. М.: Высшая школа, 2006. 480 с.
12. Texas Instruments. 2009. Amplifier and Data Converter Guide. www.ti.com/dataconverters
13. Кестер У. Как выбрать тип АЦП для ваших приложений // ЭЛЕКТРОНИКА: Наука, Технология, Бизнес. 2006. № 4. С. 12–17.
14. Загорский В. Передовые методы постобработки для высокоскоростных систем аналого-цифрового преобразования // Компоненты и технологии. 2004.
15. Texas Instruments. Digital Signal Processors & ARM Microprocessors. www.ti.com/dsp
16. Analog Devices. Embedded processing and DSP. <http://www.analog.com/dsp>
17. Самойлов В.Ф., Хромой Б.П. Основы цветного телевидения. М.: Радио и связь, 1982. 1047 с.
18. Фисенко В.Т., Фисенко Т.Ю. Компьютерная обработка и распознавание изображений. Уч. пособие. СПб.: Изд-во СПбГУ ИТМО, 2008. 192 с.

5. Автор

Кирьянов Константин Александрович, аспирант СПбГУ ИТМО, контактный адрес kiryancon@front.ru

Определение положения объекта на изображении по фрагментам границы

И.С. Симакин

Ярославский государственный университет им. П.Г. Демидова,
Факультет информатики и вычислительной техники,
Ярославль, Россия
--sis--@mail.ru

Аннотация

В работе рассматривается задача распознавания объектов на изображении по фрагментам границы. Предлагаемый метод, основанный на машинном обучении, позволяет находить объекты, совпадающие с заданными эталонами, с учетом преобразований вращения, масштабирования и параллельного переноса. Метод не требует замкнутости границы, что позволяет ему работать с пересекающимися и частично закрытыми объектами. Ключевой особенностью данного метода является обучение процессу подбора положения объекта а не процедуре сравнения.

Ключевые слова: выделение объектов, распознавание по форме границы, контурный анализ, машинное обучение.

1. ВВЕДЕНИЕ

Распознавание объектов на изображении по форме границ заключается в определении положения объектов, каждый из которых совпадает с одним из заданных эталонов.

В литературе описаны эффективные методы решения данной задачи, работающие в случае, когда положение объектов определяется аффинными преобразованиями, включающими составляющие поворота, масштабирования и параллельного переноса, а граница распознаваемого объекта представляется в виде замкнутого контура.

В книге [1] описан метод, основанный на представлении контура границы в виде периодического комплексного сигнала инвариантного к параллельному переносу. Умножение такого сигнала на комплексный множитель приводит к повороту и масштабированию контура. В этом случае распознавание сводится к выбору начальной фазы сигнала при помощи вычисления линейной свертки сигнала выделенного контура и сигнала эталонного контура и определению коэффициента корреляции.

Другим распространенным вариантом решения данной задачи является метод, описанный в [2], когда положение выделенного контура сначала нормализуется, а затем представляется в виде вектора признаков, после чего решается задача классификации.

Подобные методы обладают рядом ограничений. Во-первых, для определения формы предполагаемого объекта, его требуется сначала выделить, что вызывает трудности при работе со сложными объектами. Во-вторых, поскольку такие методы работают с замкнутыми контурами, то они не позволяют распознавать пересекающиеся или частично закрытые объекты. В-третьих, не смотря на то, что подобные методы не очень чувствительны к шуму, они плохо справляются с объектами, у которых есть размытая часть с неустойчивой границей.

Предложенный в данной работе метод определения положения объекта по фрагментам границы, позволяет работать в приведенных выше случаях. Данный метод основан на машинном обучении процессу совмещения эталона с заданными фрагментами границы. Процедура подбора положения эталона сведена к трем подзадам: выбору начального приближения для положения эталона по паре фрагментов, локальной оптимизации положения эталона и применению приемов совмещения для преодоления проблем, возникающих в результате использования локальной оптимизации.

Предложенный метод рассмотрен на примере задачи распознавания альфа-треков на фотографиях плоских диэлектрических детекторов (см. рисунок 1) [3].

Оставшаяся часть работы организована следующим образом. Во втором разделе приведено описание модели, в том числе способ представления фрагментов границ. Третий раздел посвящен процессу подбора положения объекта по набору фрагментов. В четвертом разделе кратко описана процедура поиска объектов, основанная на предложенном методе. В пятом разделе приведены условия и результаты компьютерного эксперимента.

2. ОПИСАНИЕ МОДЕЛИ

Пусть входные изображения и эталоны представлены в виде наборов ломанных линий, описывающих границы. Такое представление можно получить из исходного растрового изображения при помощи алгоритма выделения границ [4]. На рисунке 1 представлен пример фотографии детектора альфа-частиц и соответствующий набор граничных линий. Набор эталонов представлен на рисунке 2.

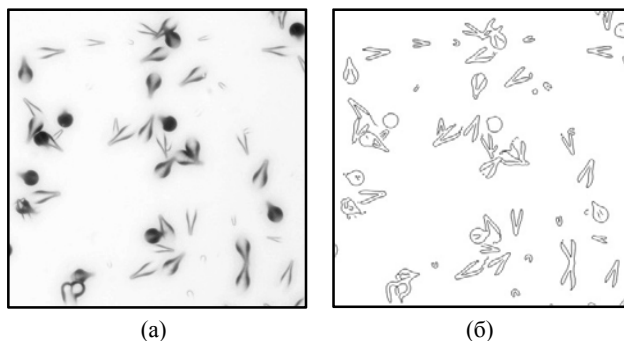


Рисунок 1: Растровое изображение поверхности детектора альфа-частиц (а) и соответствующие линии границ (б).



Рисунок 2: Набор из девяти эталонов.

Линия Γ представлена конечной последовательностью точек (Y_1, Y_2, \dots, Y_n) , заданных парами координат (Y_{x_i}, Y_{y_i}) , расставленными с шагом, равным 1.

Задача заключается в том, чтобы на входном изображении найти положения объектов, совпадающих с заданными эталонами. Положение каждого объекта определяется преобразованием координат T . В данной работе ограничимся рассмотрением аффинных преобразований, не содержащих составляющей перекоса, которые можно представить как суперпозицию вращения T_φ , равномерного по обеим осям координат масштабирования T_α с коэффициентом $\alpha > 0$ и параллельного переноса T_Δ [1]

$$T(\Delta_x, \Delta_y, \varphi, \alpha) = T_\varphi \cdot T_\alpha \cdot T_\Delta.$$

Расстояние между положениями T_1 и T_2 линии Γ определим как

$$d_T(T_1, T_2, \Gamma) = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^n |Y_i \cdot T_1 - Y_i \cdot T_2|^2}.$$

Далее построим функцию $\zeta(P, \Gamma)$ степени расхождения (противоположную к качеству совмещения) эталона P с частью входного изображения Γ . Определим функцию ζ для одной точки γ линии таким образом, чтобы она вычисляла расстояние до ближайшей к ней точки эталона P :

$$\zeta(P, \gamma) = \min_{\rho \in P} |\rho - \gamma|.$$

Для всей части входного изображения Γ зададим ζ как

$$\zeta(P, \Gamma) = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^n \zeta^2(P, \gamma_i)}. \quad (1)$$

2.1 Фрагменты линий

В основу рассматриваемой системы распознавания положена работа с небольшими фрагментами линий. Фрагмент F длины n линии Γ , начинающийся с точки под номером s , задается в виде

$$(f_1, f_2, \dots, f_n) = (Y_s, Y_{s+1}, \dots, Y_{s+n-1}), \\ ((x_{f_1}, y_{f_1}), \dots, (x_{f_n}, y_{f_n})) = ((x_{Y_s}, y_{Y_s}), \dots, (x_{Y_{s+n-1}}, y_{Y_{s+n-1}})).$$

Такое представление фрагментов будем называть точечным или координатным.

В параметрическом представлении форму и положение фрагмента будут определять пять величин, включая координаты вершины фрагмента ξ_x и ξ_y , угол наклона ξ_φ , кривизну ξ_c , и выраженность s-образного загиба ξ_s :

$$(\xi_x, \xi_y, \xi_\varphi, \xi_c, \xi_s).$$

Геометрический смысл параметрического представления фрагмента показан на рисунке 3.

Такое параметрическое представление хорошо описывает фрагменты гладких линий, то есть при переходе от точечного представления к параметрическому необходимая для распознавания объектов информация остается, а несущественная шумовая составляющая отбрасывается.

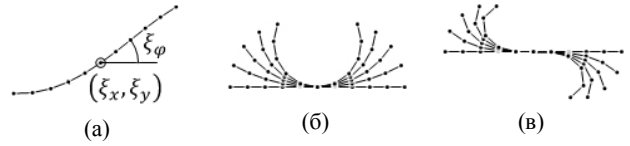


Рисунок 3: Фрагмент линии (а), изменение кривизны (б) и изменение s-образного загиба (в).

Процедура перехода от точечного задания фрагментов к параметрическому основана на преобразовании линии в угловое представление, интегрировании и последующем линейном разложении результата по специально выбранному ортонормированному базису.

К фрагментам, представленным в параметрическом виде, могут быть применены аффинные преобразования

$$\xi' = \xi \cdot T.$$

При этом параметры преобразованного фрагмента можно получить по формулам

$$(\xi'_x, \xi'_y) = (\xi_x, \xi_y) \cdot T, \\ \xi'_\varphi = \xi_\varphi + \varphi_T, \\ \xi'_c = \frac{\xi_c}{\alpha_T}, \\ \xi'_s = \xi_s.$$

Результат такого преобразования будет точным, если T не содержит составляющей масштабирования ($\alpha_T = 1$). В противном случае предложенные формулы будут являться хорошим приближением, достаточным для применения в рассматриваемой системе.

Функцию, вычисляющую расстояние между фрагментами ξ и ψ , заданными в параметрическом виде, обозначим как

$$d_\xi(\xi, \psi).$$

Она вычисляется через формулу корня квадратного из взвешенной суммы квадратов разностей специально построенных величин, заданных непосредственно через параметры фрагментов без перехода к точечному представлению.

Степень расхождения ζ эталона P и некоторого набора фрагментов изображения $\Xi = \{\xi_1, \xi_2, \dots, \xi_n\}$ определим через расстояние между фрагментами. Обозначим множество фрагментов эталона P как $\Psi(P)$. Зададим функцию ζ для одного фрагмента ξ , как расстояние от ξ до ближайшего к нему фрагмента эталона P :

$$\zeta(P, \xi) = \min_{\psi \in \Psi(P)} d_\xi(\xi, \psi).$$

После этого определим функцию ζ для всего набора фрагментов Ξ через формулу

$$\zeta(P, \Xi) = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^n \zeta^2(P, \xi_i)}.$$

3. ПОДБОР ПОЛОЖЕНИЯ ОБЪЕКТА ПО ФРАГМЕНТАМ ГРАНИЦЫ

Рассматриваемый метод распознавания основан на процедуре подбора положения эталона по нескольким заданным фрагментам таким образом, чтобы достичь совпадения со всеми этими фрагментами. Это можно сформулировать в виде оптимизационной задачи на минимизацию степени расхождения набора из n фрагментов $\xi_1, \xi_2, \dots, \xi_n$ и эталона P :

$$\begin{aligned} \tau_n(P, \xi_1, \xi_2, \dots, \xi_n) &= \tau_n(P, \Xi) = T_{\min}: \\ \zeta(P \cdot T_{\min}, \Xi) &= \min_T \zeta(P \cdot T, \Xi), \\ \Xi &= \{\xi_1, \xi_2, \dots, \xi_n\}. \end{aligned} \quad (2)$$

Поскольку от процедуры подбора положения требуется достичь совпадения со всеми заданными фрагментами, то после решения оптимизационной задачи (2), нужно проверить, что степень расхождения каждого из них с эталоном не превосходит порог ζ_0

$$\forall \xi \in \Xi, \quad \zeta(P \cdot T_{\min}, \xi) \leq \zeta_0.$$

При невыполнении этого условия констатируется, что предложенные фрагменты несовместимы, то есть не могут принадлежать одному объекту.

Если выбрать размер фрагмента таким, чтобы он составлял не более трети всего эталона, то можно сделать допущение, что одного фрагмента не достаточно для уверенного определения положения объекта. Такое допущение позволяет избежать необходимости реализации процедуры подбора положения по одному заданному фрагменту.

3.1 Подбор положения объекта по двум фрагментам

Начнем с рассмотрения процедуры подбора положения по двум заданным фрагментам. Пусть даны два фрагмента ξ и ψ в параметрическом представлении

$$\begin{aligned} (\xi_x, \xi_y, \xi_\varphi, \xi_c, \xi_s), \\ (\psi_x, \psi_y, \psi_\varphi, \psi_c, \psi_s). \end{aligned}$$

Вычислим расстояние d между вершинами фрагментов

$$d = \sqrt{(\xi_x - \psi_x)^2 + (\xi_y - \psi_y)^2}.$$

Зададим нормализующее преобразование $T_N(\xi, \psi)$, таким образом, чтобы оно переводило вершины фрагментов в симметричные относительно начала координат точки $(-d/2, 0)$ и $(d/2, 0)$. Такое преобразование будет содержать составляющие вращения и параллельного переноса. После применения нормализующего преобразования параметры фрагментов примут вид

$$\begin{aligned} \xi_N &= \xi \cdot T_N(\xi, \psi) = \left(-\frac{d}{2}, 0, \xi_\varphi + \varphi_N, \xi_c, \xi_s\right), \\ \psi_N &= \psi \cdot T_N(\xi, \psi) = \left(\frac{d}{2}, 0, \psi_\varphi + \varphi_N, \psi_c, \psi_s\right). \end{aligned}$$

Пример применения нормализующего преобразования приведен на рисунке 4.

Если взять два фрагмента ξ и ψ , принадлежащих эталону P , то соответствующее нормализующее преобразование T_N будет определять некоторое положение эталона, в котором

эталон будет совмещен с нормализованными фрагментами ξ_N и ψ_N .

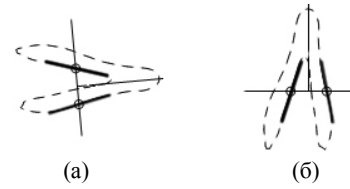


Рисунок 4: Нормализующее преобразование; (а) – положение эталона до нормализации (б) – положение эталона после нормализации по двум фрагментам.

Нормализованное положение инвариантно относительно параллельного переноса и вращения исходных фрагментов, то есть, каким бы образом ни перемещались и ни вращались фрагменты, после нормализации они примут одно и то же положение.

Пусть задана пара фрагментов исходного изображения ξ_Γ и ψ_Γ . Поскольку в рассматриваемой модели распознаваемые объекты являются преобразованными эталонами, то исходной паре соответствует некоторая пара фрагментов эталона ξ_P и ψ_P . Так как нормализованное положение инвариантно относительно преобразования вращения и параллельного переноса, то достаточно подобрать масштаб эталона α , чтобы после нормализации эти пары фрагментов совпали. Это можно условно записать в виде системы уравнений

$$\begin{aligned} \xi_\Gamma \cdot T_N(\xi_\Gamma, \psi_\Gamma) &\approx \xi_P \cdot T_\alpha \cdot T_N(\xi_P \cdot T_\alpha, \psi_P \cdot T_\alpha), \\ (\psi_\Gamma \cdot T_N(\xi_\Gamma, \psi_\Gamma)) &\approx \psi_P \cdot T_\alpha \cdot T_N(\xi_P \cdot T_\alpha, \psi_P \cdot T_\alpha). \end{aligned} \quad (3)$$

Поскольку в случае реального распознавания пара ξ_P и ψ_P заранее не известна, то ее нужно выбрать из всех возможных пар фрагментов эталона P . Масштаб α , задающий преобразование T_α выбирается из диапазона допустимых значений. Таким образом, у системы есть три неизвестных ξ_P, ψ_P и α .

Оговорка, что система уравнений (3) записана условно, означает, что она не совсем корректна с математической точки зрения. Вследствие зашумленности изображений и других факторов эти уравнения не будут выполняться строго, поэтому их нужно решать приближенно. Однако такая форма записи, во-первых, более наглядна, а во-вторых, позволяет выполнять алгебраические преобразования уравнений.

Для достижения корректности формулировки воспользуемся метрикой сравнения фрагментов и заменим систему уравнений (3) на следующую оптимизационную задачу:

$$\begin{aligned} \chi(\xi_P, \psi_P, \alpha) &= \\ &= |\xi_\Gamma \cdot T_N(\xi_\Gamma, \psi_\Gamma) - \xi_P \cdot T_\alpha \cdot T_N(\xi_P \cdot T_\alpha, \psi_P \cdot T_\alpha)| + \\ &+ |\psi_\Gamma \cdot T_N(\xi_\Gamma, \psi_\Gamma) - \psi_P \cdot T_\alpha \cdot T_N(\xi_P \cdot T_\alpha, \psi_P \cdot T_\alpha)|, \\ (\xi_P, \psi_P, \alpha)_{\min} &= \min_{\substack{\xi_P \in P \\ \psi_P \in P \\ \alpha_{\min} \leq \alpha \leq \alpha_{\max}}} \chi(\xi_P, \psi_P, \alpha). \end{aligned} \quad (4)$$

Следует отметить, что значение функции χ в точке минимума равно значению степени расхождения ζ между нормализованной парой фрагментов $(\xi_\Gamma, \psi_\Gamma)$ и масштабированным нормализованным эталоном:

$$\chi(\xi_P, \psi_P, \alpha) = \zeta(\{\xi_\Gamma, \psi_\Gamma\} \cdot T_N(\xi_\Gamma, \psi_\Gamma), P \cdot T_\alpha \cdot T_N(\xi_P \cdot T_\alpha, \psi_P \cdot T_\alpha)).$$

Данную оптимизационную задачу можно решить при помощи перебора, если зафиксировать некоторый набор возможных значений масштаба α (фрагменты ξ_P и ψ_P выбираются из конечного множества фрагментов эталона P), однако, это неэффективно.

Для построения более эффективного решения заметим, что в нормализованном положении пара фрагментов полностью описывается семимерным вектором b :

$$b(\xi_N, \psi_N) = (d, \xi_\varphi + \varphi_N, \psi_\varphi + \varphi_N, \xi_c, \psi_c, \xi_s, \psi_s).$$

Если считать, что каждая возможная пара фрагментов эталона определяет класс, то выбор пары фрагментов ξ_P и ψ_P эталона по заданному вектору b сводится к задаче многоклассовой классификации. При этом пару фрагментов не обязательно искать точно, так как для большинства случаев нахождения близкой пары фрагментов будет достаточно. В качестве функции ошибки классификации используем расстояние d_T между нормализующими преобразованиями определяемыми найденной (ξ_P, ψ_P) и оптимальной (ξ_P^*, ψ_P^*) парами фрагментов

$$err((\xi_P, \psi_P), (\xi_P^*, \psi_P^*)) = d_T(T_N(\xi_P, \psi_P), T_N(\xi_P^*, \psi_P^*), P).$$

Обучающее множество будем строить из набора всевозможных пар фрагментов эталона P , к которому применяются преобразования масштабирования с различными значениями α из допустимого диапазона. Количество обучающих примеров выбирается таким образом, чтобы их было достаточно для уверенной классификации.

Следует отметить, что для построения эффективной процедуры подбора положения по двум фрагментам требуется использовать быстрый метод классификации, обладающий меньшей трудоемкостью, чем полный перебор. В качестве примера такого метода можно привести [5].

Вернемся к процедуре подбора положения эталона по двум фрагментам. Будем считать, что пара фрагментов ξ_P и ψ_P уже определена (при помощи метода классификации), следовательно для решения оптимизационной задачи (4) осталось найти значение α . Вычислить его через отношение расстояния между вершинами фрагментов изображения ξ_G и ψ_G к расстоянию между выбранными вершинами фрагментов эталона ξ_P и ψ_P :

$$\alpha = \frac{d(\xi_G, \psi_G)}{d(\xi_P, \psi_P)}.$$

Далее найдем преобразование T , переводящее фрагменты эталона во фрагменты объекта, то есть задающее положение объекта. Выразим T из первого уравнения системы (2), применив к левой и правой части обратное преобразование $T_N^{-1}(\xi_G, \psi_G)$.

$$\xi_G \cdot T_N(\xi_G, \psi_G) \cdot T_N^{-1}(\xi_G, \psi_G) = \xi_P \cdot T_\alpha \cdot T_N(\xi_P, \psi_P) \cdot T_\alpha \cdot T_N^{-1}(\xi_G, \psi_G).$$

Так как суперпозиция прямого и обратного преобразования является тождественным преобразованием,

$$\xi_G = \xi_P \cdot (T_\alpha \cdot T_N(\xi_P, \psi_P) \cdot T_\alpha \cdot T_N^{-1}(\xi_G, \psi_G)).$$

Полученная формула осуществляет преобразование фрагмента эталона ξ_P во фрагмент объекта ξ_G . Легко проверить, что эта формула справедлива также для перехода от ψ_P к ψ_G . Следовательно, преобразование T , совмещающее эталон с парой фрагментов изображения ξ_G и ψ_G будет определяться формулой

$$\tau_2(P, \xi_G, \psi_G) = T = T_\alpha \cdot T_N(\xi_P, \psi_P) \cdot T_\alpha \cdot T_N^{-1}(\xi_G, \psi_G).$$

Таким образом, будем считать, что процедура подбора положения эталона по двум заданным фрагментам построена. Пример применения процедуры подбора положения эталона представлен на рисунке 5.

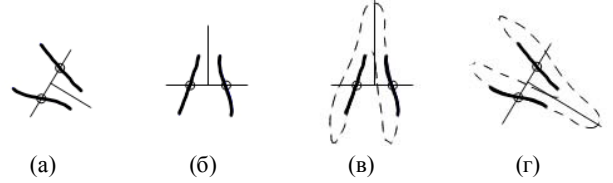


Рисунок 5: Стадии предсказания положения объекта по двум фрагментам; (а) – пара фрагментов; (б) – нормализованная пара фрагментов; (в) – предсказанное положение в нормализованной системе координат; (г) – предсказанное положение в исходной системе координат.

3.2 Локальная оптимизация положения эталона методом наименьших квадратов

Перейдем к рассмотрению процедуры подбора положения эталона по $n \geq 3$ фрагментам изображения. Итак, пусть заданы n фрагментов $\xi_1, \xi_2, \dots, \xi_n$ изображения G и эталон P . Как было определено в (2), процедура совмещения обозначается как

$$T = \tau_n(P, \xi_1, \xi_2, \dots, \xi_n).$$

Возьмем первые $n-1$ фрагментов и выполним их совмещение с эталоном P при помощи рекурсивного вызова процедуры подбора положения. Полученное положение назовем начальным и обозначим как T_0

$$T_0 = \tau_{n-1}(P, \xi_1, \xi_2, \dots, \xi_{n-1}).$$

Вспомним, что под выполнением процедуры τ_n подразумевается решение оптимизационной задачи с целевой функцией $\zeta(P \cdot T, \Xi)$ и со свободной переменной T . Будем пытаться решать эту задачу методом локальной оптимизации положения эталона с целевой функцией $\zeta(P \cdot T, \Xi)$. В качестве основы для построения метода будем использовать упрощенную версию модели, описанной в работе [6].

На каждой итерации используемого метода локальной оптимизации для каждого выделенного фрагмента изображения выбирается ближайший к нему фрагмент эталона. После этого система изменяет преобразование координат T так, чтобы “притянуть” сопоставленные пары фрагментов для достижения совмещения. Это сводится к приближенному решению некоторой системы линейных уравнений методом наименьших квадратов.

Итерации выполняются до тех пор, пока происходит существенное уменьшение значения целевой функции. В итоге система приходит в точку локального минимума. Результат локальной оптимизации T^* положения эталона P

для множества выделенных фрагментов изображения Ξ , выполненной из начального положения T_0 , обозначим как

$$\tau^*(P, \Xi, T_0) = T^*.$$

Примеры применения локальной оптимизации положения эталона приведены на рисунке 6.

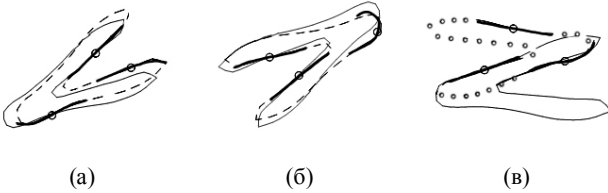


Рисунок 6: Локальная оптимизация положения; (а) и (б) – примеры удачного применения; (в) – пример, когда локальная оптимизация не позволяет достичь совмещения.

Может оказаться, что полученная точка локального минимума будет сильно отличаться от искомой точки глобального минимума. В этом случае эталон будет совмещен с изображением неправильно. Пример попадания в неправильную точку локального минимума приведен на рисунке 6 (в). Данная проблема в рамках предложенного метода локальной оптимизации не решается.

3.3 Приемы совмещения

Если внимательно проанализировать работу метода локальной оптимизации положения эталона на конкретных входных данных, то можно заметить, что многие случаи попадания в “неправильную” точку локального минимума похожи между собой. Следует подчеркнуть, что в данном случае речь идет исключительно о реально совместимых фрагментах, принадлежащих одному объекту.

Можно проклассифицировать такие случаи и для каждого класса придумать простой способ, который позволил бы выбраться из точки локального минимума, чтобы после этого достичь правильного совмещения.

Для реализации таких способов выхода из “неправильных” точек локального минимума будем использовать приемы совмещения. Под применением приема η будем понимать умножение текущего положения T_0 эталона P на некоторое преобразование T_η .

Использование приема η из начального положения T_0 будем считать успешным в случае, если выполняются два условия. Во-первых, необходимо, чтобы в результате применения преобразования T_η и последующей локальной оптимизации уменьшилось значение целевой функции ζ :

$$\zeta(P \cdot \tau^*(P, \Xi, T_0 \cdot T_\eta), \Xi) < \zeta(P \cdot \tau^*(P, \Xi, T_0), \Xi). \quad (5)$$

Во-вторых, потребуем, чтобы положение локального минимума, полученное после использования приема, значительно отличалось от положения, полученного без использования приема. В качестве порога “значительности” отличия положений прием половину расстояния между соответствующими начальными положениями (до локальной оптимизации)

$$d_T(\tau^*(P, \Xi, T_0 \cdot T_\eta), \tau^*(P, \Xi, T_0), P) > \frac{1}{2} \cdot d_T(T_0 \cdot T_\eta, T_0, P). \quad (6)$$

Рассмотрим, использование набора приемов $H = \{\eta_1, \eta_2, \dots, \eta_m\}$ в процессе совмещения эталона P с заданными фрагментами изображения $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n-1}\}$ из начального положения T_0 . Будем последовательно пробовать выполнять приемы $\eta_1, \eta_2, \dots, \eta_m$ и проверять условия успешности их применения (5) и (6).

При обнаружении первого успешного приема η_j , процесс проверки прекращается и выполняется переход к новому положению T_1

$$T_1 = T_0 \cdot T_{\eta_j}.$$

Далее положение T_1 объявляется начальным для следующей итерации и процесс попытки применения приемов повторяется.

Если успешных приемов среди $\eta_1, \eta_2, \dots, \eta_m$ найти не удастся, то положение локального минимума $\tau^*(P, \Xi, T_0)$ объявляется конечным результатом совмещения

$$\tau_n(P, \Xi) = \tau^*(P, \Xi, T_0).$$

В итоге может выстроиться целая цепочка последовательного применения приемов, заканчивающаяся выполнением локальной оптимизации

$$\tau_n(P, \Xi) = \tau^*(P, \Xi, T_0 \cdot T_{\eta_{j_1}} \cdot T_{\eta_{j_2}} \cdot \dots \cdot T_{\eta_{j_k}}).$$

Пример набора приемов для нескольких эталонов представлен на рисунке 7.

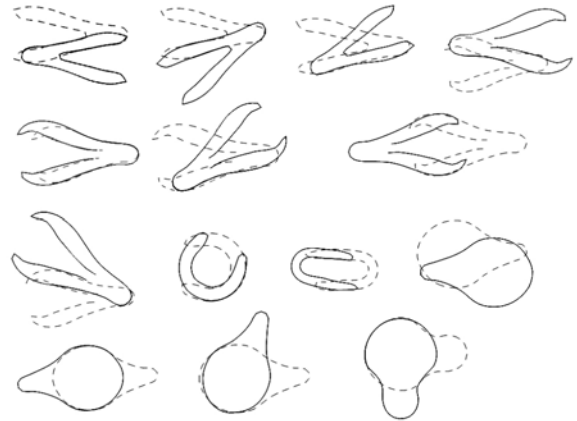


Рисунок 7: Пример набора приемов совмещения для нескольких эталонов.

Перейдем к вопросу о том, каким образом можно получить множество приемов совмещения для заданного эталона. Поскольку приемы используются для преодоления точек локального минимума, то можно предположить, что их следует строить на основе примеров попадания в локальные минимумы, полученных при обработке реальных данных. Сконструируем процедуру построения множества приемов совмещения как метод машинного обучения на примерах.

Пусть дан пример входного изображения Γ , и эталон P . Первым делом построим на изображении Γ множество возможных положений частичного совмещения K . Для этого будем много раз применять процедуру подбора положения по тройке (или по паре) случайно выбранных попарно совместимых неперекрывающихся фрагментов. Поскольку

множество приемов для эталона P еще не построено, то при подборе положений ограничимся предсказанием начального положения по двум фрагментам и локальной оптимизацией

$$\kappa = \tau^*(P, \{\xi_1, \xi_2, \xi_3\}, \tau_2(P, \xi_1, \xi_2)).$$

Кроме того, обеспечим, чтобы множество возможных положений частичного совмещения состояло из заметно отличающихся друг от друга элементов, расстояние между которыми превосходит заданный порог

$$\forall \kappa_i, \kappa_j \in K, \quad i \neq j, \quad \delta_T(\kappa_i, \kappa_j, P) > \delta_{\min}.$$

В результате в окрестности каждого объекта на изображении Γ будет содержаться по одному или несколько различных положений частичного совмещения из множества K .

После этого выполняется стадия обучения, заключающаяся в построении набора возможных приемов. Для этого много раз выполняется попытка совмещения по тройке случайно выбранных попарно совместимых неперекрывающихся фрагментов $\Xi_3 = (\xi_1, \xi_2, \xi_3)$.

Как и в процессе обычного совмещения, первым делом по двум фрагментам вычисляется начальное положение T_0

$$T_0 = \tau_2(P, \xi_1, \xi_2).$$

После этого применяется локальная оптимизация из положения T_0

$$T_0^* = \tau^*(P, \Xi_3, T_0).$$

Далее выполняется проверка на попадание в “неправильную” точку локального минимума. Для этого сначала выбирается подмножество положений частичного совмещения K_{Ξ_3} , которые могут иметь отношение к выбранной тройке фрагментов Ξ_3 . Положение частичного совмещения κ_j будем считать имеющими отношение к Ξ_3 , если степень расхождения эталона P с фрагментами будет меньше некоторого порогового значения ζ_κ

$$\zeta(P \cdot \kappa_j, \Xi_3) < \zeta_\kappa \Rightarrow \kappa_j \in K_{\Xi_3}.$$

После этого из множества положений частичного совмещения, имеющих отношение к тройке Ξ_3 , выбирается положение κ_{\min} , обеспечивающее самое лучшее совмещение после применения локальной оптимизации

$$\begin{aligned} \kappa_{\min}: \zeta(P \cdot \tau^*(P, \Xi_3, T_0 \cdot \kappa_{\min}), \Xi_3) \\ = \min_{\kappa \in K_{\Xi_3}} \zeta(P \cdot \tau^*(P, \Xi_3, T_0 \cdot \kappa), \Xi_3). \end{aligned}$$

Положение локального минимума, достижимое из κ_{\min} , назовем конечным и обозначим как $T_{\kappa_{\min}}^*$:

$$T_{\kappa_{\min}}^* = \tau^*(P, \Xi_3, T_0 \cdot \kappa_{\min}).$$

Далее проверим четыре условия. Во-первых, конечное положение $T_{\kappa_{\min}}^*$ должно обеспечивать лучшее совмещение, чем начальное положение, подвергнутое локальной оптимизации.

$$\zeta(P \cdot T_{\kappa_{\min}}^*, \Xi_3) < \zeta(P \cdot T_0^*, \Xi_3).$$

Во-вторых, необходимо, чтобы было достигнуто совмещение для каждого из фрагментов тройки Ξ_3

$$\forall \xi_i \in \Xi_3, \quad \zeta(P \cdot T_{\kappa_{\min}}^*, \xi_i) < \zeta_0.$$

Второе условие позволяет исключить обучение приемам, применение которых хотя и улучшает значение целевой функции, однако не приводит в результате к правильному совмещению.

В-третьих, для того, чтобы получить нетривиальный прием, конечное положение $T_{\kappa_{\min}}^*$ должно существенно отличаться от начального положения T_0 :

$$d_T(T_{\kappa_{\min}}^*, T_0, P) > \delta_\eta.$$

В-четвертых, необходимо, чтобы конечное положение $T_{\kappa_{\min}}^*$ заметно отличалось от результата локальной оптимизации T_0^* начального положения T_0 . Отсутствие проверки этого условия привело бы к обучению приемам, работающим в ситуациях, с которыми и так справляется метод локальной оптимизации. Для проверки данного условия воспользуемся выражением, построенным по аналогии с (6):

$$d_T(T_{\kappa_{\min}}^*, T_0^*, P) > \frac{1}{2} \cdot d_T(T_{\kappa_{\min}}^*, T_0, P).$$

В случае нарушения любого из этих условий, будем считать, что построение приема из начального положения T_0 не удалось, и следует перейти к другой тройке случайных фрагментов для продолжения обучения.

Если же данные условия выполняются, то на основе положений T_0 и $T_{\kappa_{\min}}^*$ можно построить новый прием η . Преобразование T_η , задающее прием η , выбирается таким образом, чтобы оно переводило эталон P из начального положения T_0 в найденное конечное положение $T_{\kappa_{\min}}^*$:

$$T_0 \cdot T_\eta = T_{\kappa_{\min}}^*.$$

Отсюда следует, что

$$T_\eta = T_0^{-1} \cdot T_{\kappa_{\min}}^*.$$

Построенный прием η добавляется в список приемов N эталона P , после чего происходит переход к следующей попытке.

В процессе обучения приемам учитываются ранее построенные приемы. Если множество приемов N не пусто, то при выполнении попытки обучения новому приему, сначала проверяется возможность применения существующих приемов. Если из начального положения T_0 удастся успешно выполнить хотя бы один из этих приемов, то происходит переход к следующей попытке обучения с новой тройкой фрагментов.

После построения множества возможных приемов выполняется их ранжирование, то есть упорядочивание по убыванию полезности. Более полезными считаются приемы, применимые в большем количестве ситуаций.

Ранжирование выполняется следующим образом: сначала ищется прием, применимый в самом большом количестве ситуаций. После этого ищется прием, который лучше всего помогает в ситуациях, с которыми не справляется первый

прием. Далее – лучший прием, для ситуаций, с которыми не справились первые два приема, и так далее.

Таким образом, процедура $\tau_n(P, \Xi)$ подбора положения эталона P по n фрагментам $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n-1}\}$ изображения была сведена к процедуре τ_{n-1} подбора начального положения по $n - 1$ фрагменту, применению подготовленных на основе обучения приемов совмещения $H = \{\eta_1, \eta_2, \dots, \eta_m\}$ и локальной оптимизации $\tau^*(P, \Xi, T_0)$.

4. ПРОЦЕДУРА ПОИСКА ОБЪЕКТОВ

Процедура поиска объектов отвечает, во-первых, за выделение наборов фрагментов, по которым выполняются попытки подбора положения эталона, а, во-вторых, за выбор лучшего набора положений при построении конечной расстановки объектов на изображении, выдаваемой в качестве результата распознавания.

Поскольку построение процедуры поиска объектов рассматривалось при выполнении исследований как второстепенная задача, то для ее решения был выбран наиболее простой алгоритм, достаточный для проверки работоспособности и перспективности метода подбора положения эталона.

Итак, из множества фрагментов $\Psi(\Gamma)$ изображения Γ по порядку выбираются фрагменты ψ_i . На каждом шаге ψ_i обозначается как первый выделенный фрагмент ξ_1

$$\xi_1 = \psi_i \in \Psi(\Gamma).$$

После этого строится подмножество Ψ_{ξ_1} множества фрагментов $\Psi(\Gamma)$, соседних с ξ_1 , то есть расположенных на расстоянии, не превосходящем диаметр эталона P . После этого из Ψ_{ξ_1} случайным образом выбирается второй фрагмент ξ_2 , совместимый и не перекрывающийся с ξ_1 . Положение k , полученное в результате совмещения, добавляется во множество возможных положений объектов K , обозначенное по аналогии с множеством возможных положений частичного совмещения из раздела 3.3:

$$k = \tau_2(P, \xi_1, \xi_2), k \rightarrow K.$$

Далее из Ψ_{ξ_1} случайным образом выбирается фрагмент ξ_3 , попарно совместимый и не перекрывающийся с фрагментами ξ_1 и ξ_2 . Если вся тройка фрагментов оказывается совместимой, то результат совмещения k добавляется во множество K

$$k = \tau_3(P, \xi_1, \xi_2, \xi_3), k \rightarrow K.$$

Добавление новых фрагментов к выборке выполняется до тех пор, пока удается достигать совмещения. Попытки построения случайных наборов фрагментов $\{\xi_1, \xi_2, \dots\}$ выполняются для каждого ψ_i заданное количество раз.

Поскольку одному реальному объекту на изображении может соответствовать несколько различных элементов построенного множества возможных положений K , то необходимо выбрать лучшую расстановку объектов, таким образом, чтобы одному реальному объекту соответствовало одно положение эталона.

В качестве целевой функции для выбора лучшей расстановки объектов используется суммарное качество расстановки объектов. Значение этой величины зависит от качества совмещения с изображением каждого объекта, входящего в

расстановку, и от меры неразличимости каждой пары выделенных объектов.

Величина, характеризующая качество совмещения эталона P с частью изображения Γ для заданного положения k , строится на основе функции $\zeta(P \cdot k, \Gamma)$ (1).

Мера неразличимости двух объектов $\theta(P_1 \cdot k_1, P_2 \cdot k_2)$, заданных положениями k_1 и k_2 эталонов P_1 и P_2 , принимает значения от 0 до 1 и характеризует правдоподобность того, что оба объекта могут быть одновременно различимы на изображении. Если два объекта не пересекаются, то они легко различимы друг от друга, следовательно, для них значение функции θ будет равно 0. Если же два объекта полностью совпадают, то они являются заведомо неразличимыми, поэтому

$$\theta(P \cdot k, P \cdot k) = 1.$$

Для выбора лучшей расстановки объектов используется простой жадный алгоритм, выбирающий на каждом шаге из множества возможных положений K лучшее положение k , максимально увеличивающее суммарное качество расстановки. Выбор объектов осуществляется до тех пор, пока происходит увеличение суммарного качества расстановки.

Описанная выше реализация процедуры поиска объектов характеризуется повышенной вероятностью появления ошибок распознавания в сложных случаях и относительно низким быстродействием, недостаточным для работы с большими объемами данных в режиме пакетной обработки и в режиме реального времени.

5. КОМПЬЮТЕРНЫЙ ЭКСПЕРИМЕНТ

Для выполнения компьютерного эксперимента были использованы пять фотографий поверхности детектора альфа-частиц, масштабированные до размера 784×588 пикселей. Эти фотографии были разрезаны на куски размером 256×256 пикселей. Каждый из 34 полученных кусков использовался как входное изображение. Пример входного изображения представлен на рисунке 1.

Набор из девяти эталонов (см. рисунок 2) был построен вручную. Каждый эталон был вырезан из исходного изображения и повернут в горизонтальное положение.

После этого выполнялось обучение классификаторов для предсказания положения объектов по двум фрагментам, по одному на каждый эталон. На обучение одного классификатора требовалось около 5-7 минут.

После этого была выполнена процедура обучения приемам совмещения на части входных изображений, пока суммарное количество приемов не достигло 50.

Далее все входные изображения были обработаны системой распознавания. На обработку одного изображения размером 256×256 пикселей требовалось около 1 минуты. Результаты распознавания были проверены визуально. Типичный пример распознанного изображения приведен на рисунке 8. На рисунке 9 показано множество лучших положений и лучшая расстановка объектов для того же самого изображения.

Мелкие объекты остались нераспознанными в следствие того, что для них не были заданы эталоны. Недостаточная тщательность распознавания (пропуск) крупных объектов объясняется, во-первых, упрощенным устройством процедуры поиска объектов, а во-вторых, несовершенством механизма вычисления меры неразличимости объектов,

используемым в процессе выбора лучшей расстановки. Эти проблемы являются устранимыми и не имеют непосредственного отношения к самому метод определения положения объектов по фрагментам границы.

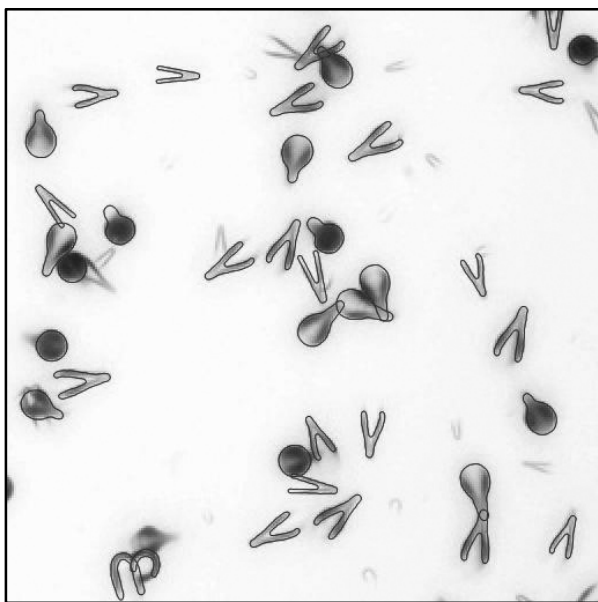


Рисунок 8: Результат распознавания.

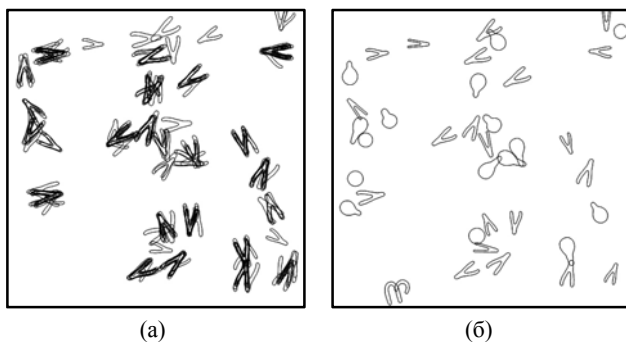


Рисунок 9: Множество возможных положений для одного из эталонов (а) и лучшая расстановка (б).

6. ЗАКЛЮЧЕНИЕ

Ключевой идеей предложенного метода можно считать использование приемов совмещения, получаемых при помощи процедуры обучения на примерах. Приемы позволяют добиться эффективной работы с эталонами с учетом самоподобия их частей.

Рассмотренная система распознавания на некоторых этапах работы использует машинное обучение на примерах. Важной особенностью, на которую следует обратить внимание, является использование процедуры обучения **процессу совмещения** эталонов с объектами, тогда как большинство описанных в литературе методов распознавания, основанных на обучении на примерах, осуществляет обучение **процессу узнавания** эталонов по набору признаков.

Не смотря на то, что процедура поиска объектов рассматривалась в данной работе лишь как часть системы, необходимая для исследования метода подбора положения эталона по фрагментам, и была построена в максимально

упрощенном виде, она, тем не менее, обеспечила возможность распознавания входных изображений, содержащих реальные данные. Результаты компьютерных экспериментов подтвердили работоспособность и перспективность идей, заложенных в основу системы.

В заключение следует отметить, что предложенный метод поиска объектов можно применять не только к изображениям, представленным в виде наборов линий. Вместо фрагментов границ можно было бы использовать, например, особые точки изображения.

7. ЛИТЕРАТУРА

- [1] Фурман Я.А. Введение в контурный анализ; приложения к обработке изображений и сигналов. 2-е изд., исп. – М.: ФИЗМАТЛИТ, 2003. – 592 с. ISBN 5-9221-0374-1.
- [2] Giannakou V., Tzouveli P., Avrithis Y. and Kollias S. Affine Normalization Curve Matching Method Using Curvature Scale-Space and Normalization. Sixth International Workshop on Content-Based Multimedia Indexing (CBMI 2008), London, 2008
- [3] Roussetski A.S. CR-39 Track Detectors in Cold Fusion Experiments: Review and perspectives. Condensed Matter Nuclear Science. Proceedings of the 11th International Conference on Cold Fusion, 2004.
- [4] Devernay F. A Non-Maxima Suppression Method for Edge Detection with Sub-Pixel Accuracy. Research report 2724, INRIA Sophia-Antipolis, 1995.
- [5] Andoni A., Indyk P. Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions. Communications of the ACM, vol. 51, no. 1, 2008.
- [6] Cootes T. F., Taylor C. J., Cooper D. H., and Graham J. Active shape models - their training and application. Computer Vision and Image Understanding, Jan. 1995.

Благодарности

Автор выражает благодарность сотрудникам лаборатории дозиметрии и радиоактивности окружающей среды химического факультета МГУ Ю.А. Сапожникову, С.Н. Калмыкову и И.Э. Власовой за поставленную задачу распознавания альфа-треков и предоставленные фотографии детекторов альфа-частиц.

Об авторе

Симакин Иван – аспирант Ярославского государственного университета им. П.Г. Демидова, факультет информатики и вычислительной техники, email: --sis--@mail.ru.

ESTIMATION OF OBJECT POSITION ON IMAGE USING EDGE FRAGMENTS

Abstract

In this paper we consider a problem of object recognition using edge fragments. The proposed method is based on machine learning. It permits to detect objects matching specified templates regardless of various rotation, scaling and translation transforms. The proposed method doesn't require closed edge contours therefore it allows working with intersecting and covered objects. Key feature of this method is learning the object position estimation process instead of learning the comparing procedure.

Keywords: object detection, contour analysis, shape recognition, machine learning.

Встроенная видеоаналитика для детектирования и сопровождения объектов при помощи многомасштабных признаков

Птицын Н.В., ООО «Синезис», pr@synesis.ru

Аннотация

Предложен новый алгоритм встроенной видеоаналитики для автоматического детектирования и сопровождения подвижных объектов в системах охранного наблюдения.

Для детектирования объектов заимствованы принципы анализа изображения из зрительной коры мозга. Использована последовательность простых пиксельных операций: линейных (взвешенное суммирование) и нелинейных (выбор максимального значения, насыщение).

Для сопровождения объектов использован гибридный алгоритм на основе (1) анализа временного ряда детектора объектов и (2) пространственной корреляции признаков текущего кадра и статистической модели объекта.

Отличительными особенностями разработанного конвейера являются вычислительная эффективность на видеопотоке высокой четкости (HD) и возможность сопровождения слабоконтрастных перекрывающихся целей на изменчивом фоне. Алгоритм реализован и внедрен на различных платформах, в том числе на сигнальных процессорах (DSP). Оценка точности встроенных алгоритмов детектирования и сопровождения произведена по методике i-LIDS. Полностью встроенная реализация на DSP одобрена i-LIDS как система первичного обнаружения для формирования оперативных тревог и для записи событий в приложениях видеонаблюдения стерильной зоны.

Ключевые слова: *встроенная видеоаналитика, первичная зрительная кора VI, многомасштабный анализ, детектирование движения, сопровождение целей, поточное видео, i-LIDS.*

1. ВСТУПЛЕНИЕ

Автоматизация обработки поточного видео в системах охранного телевидения является актуальной научно-инженерной задачей. *Видеоаналитикой* называются программное обеспечение на базе алгоритмов машинного зрения, которое позволяет детектировать, сопровождать, классифицировать и/или идентифицировать подвижные объекты в поле зрения видеокамеры без участия оператора [1], с. 287-312. Перспективным направлением видеоаналитики является встраивание алгоритмов непосредственно в камеру или ip-устройства [2]. По сравнению с серверной реализацией, *встроенная аналитика* обрабатывает видеосигнал без искажений, вносимых цифровым или аналоговым каналом связи. Такая встроенная обработка видео на более высоком разрешении и с большей частотой кадров потенциально обеспечивает более высокую точность распознавания.

С другой стороны, известные алгоритмы видеоаналитики достаточно сложно адаптировать для встроенной реализации в камерах массового производства из-за их вычислительной сложности. Аппаратные ресурсы однокристальной платформы камеры (набор процессорных команд, тактовая частота и размер памяти) лимитированы из-за ограничений по тепловыделению и стоимости. Таким образом, необходима

разработка принципиально новых алгоритмов с большей вычислительной эффективностью, особенно, если в камере используется сенсор высокой четкости.

Главная задача встроенной аналитики – обеспечить первичное обнаружение и сопровождение целевого объекта в поле зрения камеры. Результатом работы такой аналитики являются координаты, траектория движения и признаки объектов. Другие задачи, такие как уточняющая классификация, идентификация и межкамерное сопровождение могут быть эффективно выполнены на стороне сервера.

2. КЛАССИЧЕСКИЙ ПОДХОД К ДЕТЕКТИРОВАНИЮ ОБЪЕКТОВ

Общий алгоритмический подход к детектированию подвижных объектов состоит в анализе различий между текущим кадром и моделью фона. Упрощенно, этот подход называется *вычитанием фона* (background subtraction). Принадлежность пикселя к объекту (или к фону) определяется на основе отклонения значения (яркости) пикселя в текущем кадре от его статистической оценки в модели фона. Известно множество методов моделирования фона изображения [3]. Наиболее распространены методы бегущего гауссовского среднего (running Gaussian average) и гауссовой смеси (mixture of Gaussians):

1. Метод бегущего среднего успешно работает в стерильных условиях, когда фон остается неподвижным. В случае глобального изменчивого фона, например, движения леса, кустарников и воды, одномодальная модель не позволяет эффективно обнаруживать инородные объекты.
2. Метод гауссовской смеси является многомодальным и позволяет более точно описывать статистику изменчивого фона. Однако, при моделировании отдельных пикселей, а не их паттернов, он не обеспечивает заметного прироста в точности детектирования. При этом метод гауссовской смеси является существенно более ресурсоемким, чем бегущее среднее, и обычно не вписывается в рамки вычислительных возможностей встроенного процессора.

На этапе сегментирования, отдельные пиксели, выделенные из фона, объединяются в регионы при помощи морфологических операций [4], с. 481-495. Регионы, соответствующие заданным габаритам и форме, можно считать целевыми объектам. Как правило, именно фаза морфологического анализа составляет наибольшие вычислительные затраты. Сложность анализа нелинейно возрастает с увеличением площади регионов и их количества. Основные проблемы встроенной аналитики на базе классического подхода состоят в следующем:

1. Нелинейный рост алгоритмической сложности при увеличении пиксельного размера кадра и/или целевых объектов. Так, большинство встроенных алгоритмов работают на разрешении от 160 x 120 до 320 x 240 пикселей и практически не могут быть использованы

режиме высокой четкости 1920 x 1080 пикселей. Нелинейный рост сложности обусловлен морфологическими операциями детектора при объединении крупных регионов. Это ограничение не позволяет воспользоваться потенциалом мегапиксельных камер и увеличить дальность действия видеоаналитики.

2. Недостаточная точность детектирования объектов на изменчивом фоне в условиях слабой контрастности.
3. Высокая частота ложных срабатываний, обусловленных естественными проявлениями окружающей среды (облака, ветер, снег, дождь, птицы и насекомые).

3. НОВЫЙ АЛГОРИТМ

3.1 Нейробиологический механизм

Идея предлагаемого алгоритма заимствована у природы, где эволюция нервной системы живых существ достигла выдающихся результатов в области видеонализа [5]. Рассмотрим функциональную схему первичной зрительной коры V1 (рис. 1), развитой, в частности, у приматов и человека. Нейронная сеть состоит из клеток двух типов:

1. Простые клетки, обозначаемые буквой S от англ. simple, осуществляют линейную операцию взвешенного суммирования (**sum**), то есть двумерную свертку:

$$L_{m,n}^{out} = \sum_u \sum_v w_{u,v} L_{m-u,n-v}^{in},$$

где $L_{m,n}^{out}$ – выходной сигнал в точке (m, n) , $w_{u,v}$ – весовой коэффициент (ядро свертки) в окрестной точке (u, v) , $L_{m+u,n+v}^{in}$ – входной сигнал в точке $(m + u, n + v)$.

2. Сложные клетки, обозначаемые буквой C от англ. complex, осуществляют нелинейную операцию выбора максимального значения (**max**):

$$L_{m,n}^{out} = \max_u \max_v L_{m-u,n-v}^{in}, \quad u^2 + v^2 \leq r^2.$$

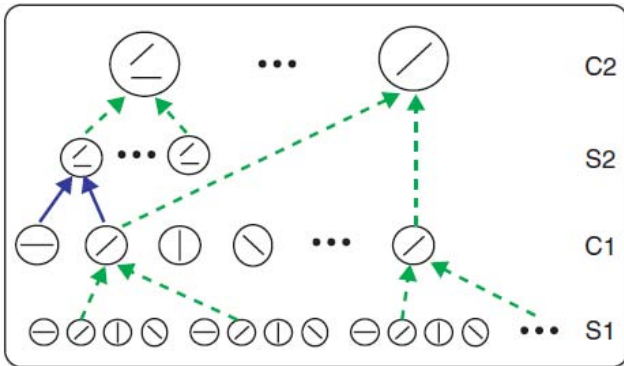


Рис 1: Обработка изображения в первичной зрительной коре: черточки – признаки, соответствующие направленным фильтрам; S1, S2 – слои простых клеток; C1, C2 – слои сложных клеток; сплошная синяя линия – взвешенное суммирование; штриховая зеленая линия – операция выбора максимума.

На входе нейронной сети исходное изображение с сетчатки глаза обрабатывается простыми клетками S1. Простые клетки реализуют направленную фильтрацию, которая выделяет границы определенных ориентаций. Задача направленного фильтра – выделить характерные признаки изображения, инвариантные к освещению. На рис. 1 показано четыре направления: горизонтальное, вертикальное и два диагональных. В области машинного зрения широко применяются аналогичные детекторы границ на базе градиента [4], с. 315-338.

На уровне сложных клеток C1 происходит группировка простых клеток S1 по каждому направлению и выделение максимального значения. Сложная клетка обладает избирательностью по признаку и обеспечивает инвариантность относительно смещения входного нейрона в окрестности группировки.

На уровне простых клеток S2 производится взвешенное суммирование выходов сложных клеток C1. В результате суммирования сигналов по различным признакам с уровня S2 получаются композитные признаки, объединяющие локальные данные по нескольким направлениям. Они похожи на признаки Хаар, но за счет предшествующего слоя сложных клеток лучше обобщают деформируемые формы.

На уровне сложных клеток C2 опять применяется нелинейная операция **max**, причем группируются не только выходы предшествующего уровня S2 (инвариантность позиции), но и выходы еще более нижнего слоя C1 (инвариантность масштаба). Таким образом, на уровне C2 простые и сложные признаки объединяются для достижения инвариантности и по смещению и по масштабу одновременно.

С другой стороны, важными особенностями клеток зрительной коры является (1) нелинейное поведение во времени и (2) нелинейная передача контраста [6]. Остановимся подробнее на особенности (2): нелинейное преобразование контраста выражается в насыщении (операция **saturate**) выходного значения признака на определенном уровне, что обеспечивает его нормирование в условиях неравномерной контрастности. Насыщение проявляется как у простых, так и сложных клеток.

Уравнение Нака-Руштона (Naka-Rushton) аппроксимирует передаточную характеристику насыщения (рис. 2):

$$L^{in}/L_{max}^{out} = L^{in}/(L^{in} + L_{sat}^{in}),$$

где L^{in} значение признака на входе клетке, L_{sat}^{in} – точка полунасыщения (на рис. $L_{sat}^{in} = 100$), L^{out} – значение на выходе клетки, L_{max}^{out} – максимальное выходное значение. Передаточную характеристику можно так же рассматривать как функцию активации нейрона.

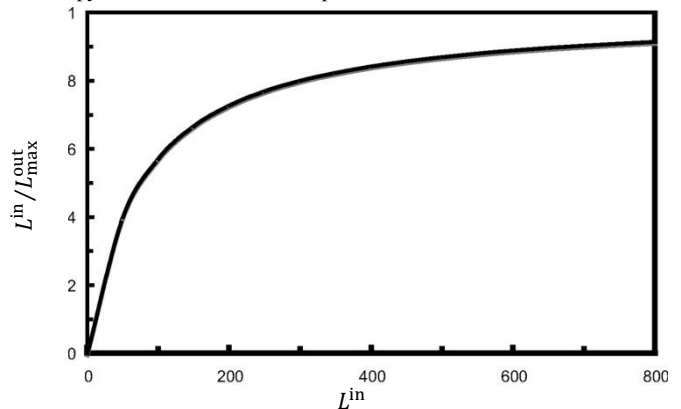


Рис 2: Передаточная характеристика клетки зрительной коры. Отметим следующие особенности зрительной коры V1 для их последующей адаптации к машинной видеоаналитике:

1. Нейронная сеть обобщает данные и понижает их размерность за счет устранения вариаций по положению и масштабу. При распространении данных от сетчатки глаза по зрительной коре понижается пространственная детализация исходного изображения, но увеличивается размерность признаков.

- Первичным признаком является не абсолютная яркость в каждой точке пространства, зарегистрированная светочувствительной клеткой, а производные признаки, полученные применением направленных фильтров к изображению.
- Простые и сложные клетки реализуют нормирование выходного сигнала за счет насыщения значения признака на некотором постоянном уровне (операция **saturate**).
- Композитные признаки получаются при помощи линейной операции взвешенного суммирования **sum** по различным признакам. Так же имеет место гауссовское суммирование сигналов, соответствующих одинаковым признакам для их обобщения.
- Инвариантность к геометрическим преобразованиям сдвига и масштабирования достигается за счет нелинейной операции **max**. При этом горизонтальное агрегирование выходов клеток одного слоя обеспечивает инвариантность относительно сдвига, а вертикальное агрегирование выходов клеток с одного или нескольких предшествующих слоев – инвариантность относительно масштаба.
- Операции **max** и **sum** применяются поочередно и итерационно. Композиция множества двух простых операций (линейной и нелинейной) порождает высокую сложность системы видеонализа в целом. Похожий прием используется в блочных алгоритмах шифрования для достижения максимальной диффузии данных внутри блока за минимальное число арифметических операций.

3.2 Многомасштабное представление

Многомасштабный подход [4], с. 125-142, уже успешно применялся для детектирования движения [7] и сегментирования сложных сцен [8]. Однако, указанные алгоритмы сегодня не пригодны для массового применения в камерах видеонаблюдения из-за их вычислительной сложности. В данной работе мы рассматриваем подходы, которые позволяют снизить ресурсоемкость алгоритмов на несколько порядков и применить алгоритмы в системах встроенного видеонализа.

Рассмотрим многомасштабное представление одного признака (простого или композитного) в виде пирамиды на рис. 3. Таких пирамид может быть несколько для каждого признака, а так же для маски сегментирования и других вспомогательных данных.

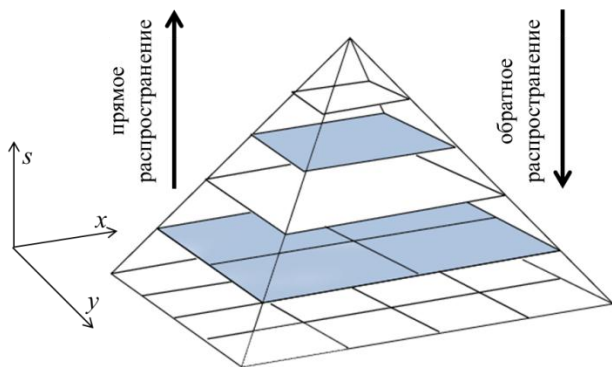


Рис 3: Две фазы многомасштабного видеонализа: x, y – пространственные координаты изображения, s – пространство масштаба.

Оптимальный алгоритм встроенной видеоналитики включает две фазы обработки многомасштабных данных:

- 1) Прямое распространение от детального к грубому представлению (обобщение).
- 2) Обратное распространение от грубого к детальному представлению (уточнение). В обратной фазе обработку можно локализовать только областью детектирования и сопровождения.

В ходе прямого распространения каждый последующий слой пирамиды вычисляется при помощи простых линейных и нелинейных операций, рассмотренных выше. Операция **sum** может выступать аналогом операции **reduce** при построении гауссовой пирамиды [4], с. 137. Операция **max** усиливает характерные признаки и не дает им «размываться» в пирамиде, а так же, как обсуждалось выше, обеспечивает инвариантность относительно сдвига и масштаба. Операция **saturate** важна для устойчивой работы детекторы в условиях неравномерной освещенности и шума.

Оптимальный набор признаков и последовательность применения операций **sum**, **max** и **saturate** зависит от конкретных функций, выполняемой встроенной аналитикой:

1. Для детектирования и сопровождения объектов необходим относительно небольшой набор признаков (1-4 признака). Обычно не требуются композитные признаки. С другой стороны, увеличение числа признаков, позволяет упростить алгоритм статистического моделирование фона и повысить чувствительность детектора на изменчивом фоне.
2. Для распознавания типов объектов и идентификации важна хорошая репрезентативность признаков. На практике, видеоаналитический алгоритм может иметь 8-64 композитных признака.

На рис. 2 представлен результат применения машинного видеодетектора с использованием трех признаков (яркость и два насыщенных градиента). Алгоритм уверенно регистрирует уток на изменчивом фоне (водная рябь с контрастными отражениями). Используются одноомодальная вероятностная модель фона, многомасштабный сегментатор, описанный ниже, и примитивный алгоритм сопровождения (связывание регионов в траекторию без построения статистической модели объекта).

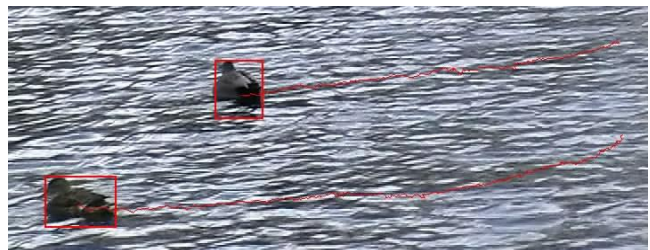


Рис 4: Сопровождение уток на изменчивом и контрастном фоне. Полное видео можно посмотреть по адресу <http://www.youtube.com/watch?v=PmJTnCIUjYw>

3.3 Сегментирование объекта

Маска объекта, рассчитываемая сегментатором, полезна для вычисления признаков объекта, для его сопровождения и для более точного моделирования фона.

Как было отмечено выше, существенным недостатком классического подхода является высокая ресурсоемкость морфологических операций на этапе определения маски регионов. Многомасштабный подход позволяет существенно повысить вычислительную эффективность видеоналитики за счет ограничения глубины сегментирования и/или использования модели формы объекта. При этом габариты

детектируемых объектов, которые могут существенно варьироваться в поле зрения камеры, не будут значительно влиять на вычислительные затраты алгоритма.

Рассмотрим подробнее предлагаемый алгоритмический подход. В ходе *обратного распространения* сигнала в пирамиде (рис. 3), формируется многомасштабная маска региона. Сегментирование производится от грубой к детальной маске, причем процесс можно прервать после достижения желаемого уровня детализации или после исчерпания квоты вычислительных ресурсов. Входными данными являются маска с предшествующего региона, разность признаков текущего кадра и модели фона на текущем слое, а так же, опционально, модель детектируемого объекта. Выходными данными является маска региона на текущем слое. На каждом пикселе производится уточнение маски на основе метода *связывания в пирамиде* [4], с. 433-436.

На рис. 5 представлен результат работы многомасштабного сегментатора на базе метода связывания в пирамиде по нескольким признакам: сверху - исходный кадр, обрабатываемый видеоаналитикой с траекторией сопровождаемого человека; ниже - четыре слоя маски сегментирования. Маски содержат незначительные ошибки в виде точек и неточных границ, обусловленные вычислительной оптимизацией морфологических операторов.

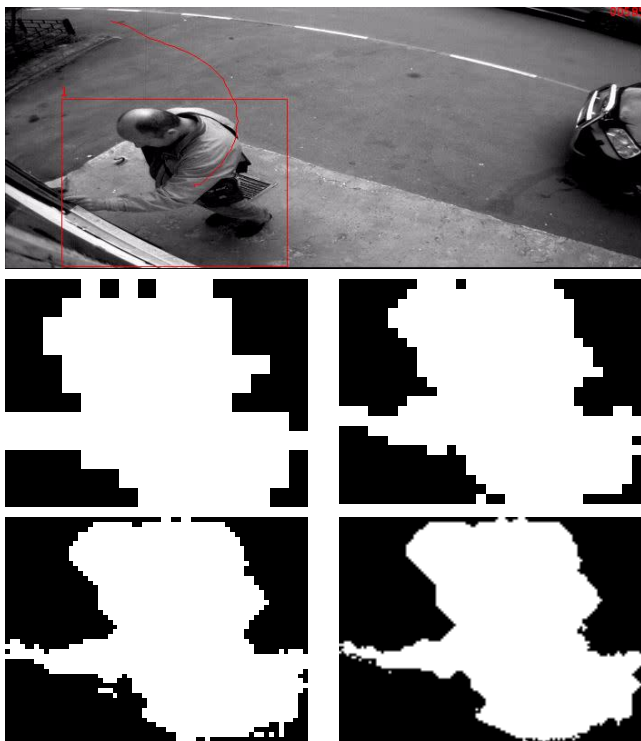


Рис 5: Многомасштабное сегментирование: исходное изображение и маски с увеличивающейся детализацией

3.4 Гибридная система сопровождения

Алгоритмы сопровождения (трассирования) позволяют получить траекторию объектов для более точного распознавания и оценки динамических признаков.

В классических реализациях встроенной аналитики популярны следующие алгоритмические подходы:

1. *Объединение регионов*, определенных детектором, во времени по множеству последовательных кадров

позволяет рассчитать искомую траекторию движения объекта. Это самый простой метод с точки зрения реализации. Основной его недостаток состоит в том, что ошибки детектора регионов приводят к нежелательным разрывам траектории. Так же, метод не позволяет сопровождать объекты в группе и сбивается при появлении ложных регионов. При медленном движении или остановке, объекты «врастают» в фон, и алгоритм сопровождения теряет цель.

2. *Корреляционные методы* предполагает построение статистической модели не только фона, но и объекта. Расчет степени схожести объекта в различных точках окрестности объекта позволяет определить наиболее вероятное его положение [4], с. 407. Преимуществом такого подхода по сравнению с первым является возможность сопровождения частично перекрывающихся объектов в группе, а так же более устойчивая работа со слабоконтрастными или медленными объектами. Основной недостаток – существенно более высокая ресурсоёмкость. Корреляционные методы неэффективны при низкой частоте кадров и сильной изменчивости сопровождаемых объектов.
3. *Оптический поток* опирается на предположение, что освещение постоянно, а форма и текстура фона и сопровождаемого объекта не меняются [4], с. 385. Метод оптического потока с вычислительной точки более эффективен, чем корреляционный метод, но уступает ему по стабильности в условиях шума и изменчивости объекта.



Рис 6: Сопровождение изменчивого объекта при помощи алгоритма связывания регионов

Более подробно методы сопровождения объектов в видеопотоке описаны в книге [4], с. 375-412.

В настоящей работе мы предлагаем гибридный метод на основе подходов (1) и (2). С одной стороны, анализ временного ряда найденных регионов позволяет эффективно сопровождать *изолированные* объекты, в том числе, которые значительно меняют форму (рис. 6). С другой стороны, корреляционный метод позволяет сопровождать объекты в группе (рис. 7) или когда детектор не обладает достаточной чувствительностью для нахождения регионов.

Объединение результатов работы алгоритмов на базе подходов (1) и (2) производится путем выбора наиболее вероятной оценки положения объекта. Для подхода (1) основой для оценки служит контрастность региона на фоне, а для подхода (2) – значения корреляции признаков по площади объекта. Подход (1) запрещается при взаимном перекрытии сопровождаемых объектов.

Многомасштабное представление маски и признаков объекта позволяет существенно повысить вычислительную эффективность корреляционного алгоритма и увеличить радиус поиска, то есть решает проблемы апертуры [4], с. 379.



Рис 7: Индивидуальное сопровождение объектов в момент встречи (наверху) и после встречи (внизу) при помощи корреляционного алгоритма. Фрагмент видео из PETS [9].

3.5 Псевдокод конвейера видеоаналитики

Ниже представлен псевдокод конвейера разработанной видеоаналитики. Он описывает алгоритм обработки одного кадра видеопоследовательности.

Псевдокод 1 Цикл обработки одного кадра

1. Получить исходное изображение I
2. Построить гауссову пирамиду P_i из I
3. Рассчитать пирамиды градиентных и композитных признаков P_{F1}, P_{F2}, \dots из P_i (раздел 3.2)
4. Вычислить пирамиду разностей P_D между пирамидами текущего кадра P_{F1}, P_{F2}, \dots и пирамидами фона P_{B1}, P_{B2}, \dots
5. Получить маски регионов P_M из P_D с помощью алгоритма сегментирования (раздел 3.3)
6. Рассчитать траектории движения объектов гибридным алгоритмом сопровождения (раздел 3.4) из P_M (метод объединения регионов) и из P_i, P_{B1}, P_{B2} и моделей объектов (метод корреляции признаков)
7. Обновить модель фона P_{B1}, P_{B2}, \dots путем расчета гауссова среднего из P_{F1}, P_{F2}, \dots , при этом подвижные объекты маскируются с помощью P_M
8. Обновить модель объектов путем расчета гауссова среднего из P_{F1}, P_{F2}, \dots по маске P_M .

Точность и ресурсоемкость алгоритма контролируются за счет частоты кадров (допускается частичный пропуск), разрешения входного кадра, глубины сегментирования, области детектирования, области сопровождения, радиуса поиска корреляционного метода и др. настройками.

4. АППАРАТНО-ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

Видеоаналитическое обеспечение для детектирования и сопровождения объектов на базе алгоритма, приведенного в разделе 3.5, реализовано на двух аппаратных платформах: (1) x86 с использованием инструкций SSE2 для тестирования и (2) на сигнальном процессоре для встраивания непосредственно в камеру или видеозаписывающее устройство (рис. 8). Отлажено серийное производство оборудования с разработанной аналитикой.

Алгоритмы работают на всех платформах в реальном масштабе времени на разрешении от 240 линий (стандартная четкость) до 1080 линий (высокая четкость).



Рис 8: Однокристалльная реализация видеоаналитики на платформе Texas Instruments DaVinci TMS320DM6467. Размер платы, 80 x 55 x 14 мм, соответствует габаритам банковской карты.

5. МЕТОДИКА И РЕЗУЛЬТАТЫ ИСПЫТАНИЙ

Внутренние испытания видеоаналитики проводилось на специальном стенде. Разработано аппаратно-программное обеспечение для автоматического тестирования видеокамер и видеосерверов со встроенной аналитикой.

В качестве исходных материалов для тестирования использовалось видео, записанное с реальной системы уличного видеонаблюдения. Набор видеосюжетов соответствует сценарию «стерильная зона» [10] и содержит:

- 432 случая нарушения периметра (движение с различной скоростью, шагом, бегом, кувырком, ползком, в маскировочных халатах, с лестницей, в группе и по нестандартным траекториям);
- около 500 ситуаций для потенциальных ложных срабатываний (резкое изменение освещенности, движение теней, дрожание камеры, мелкие млекопитающие, птицы, насекомые на объективе, плотный снег, дождь, туман).

Набор видеосюжетов состоит из фрагментов, записанных в различное время года, суток, а так же при различных погодных условиях. Суммарная продолжительность видео – около 38 часов. Источник – стандартная камера с сенсором CCD и аналоговым выходом PAL (720 x 576 x 25 кадров/с). Цифровой формат хранения видео MJPEG, поток данных избыточный – 40 Мбит/с – что максимально приблизило качество записанного и «живого» сигналов.

Табл. 1 Точность видеоаналитики по сценарию «стерильная зона»

	Весовой параметр	Чувствительность	Специфичность	Взвешенная средняя точность
Формула	α	$r = a/(a + c)$	$p = a/(a + b)$	$F_1 = \frac{(\alpha + 1)rp}{r + \alpha p}$
Оперативная тревога	0.65	1.00	1.00	1.00
Запись событий	75.00	1.00	1.00	1.00

Разметка видеосюжетов производилась независимой группой специалистов в области безопасности и видеонаблюдения. Эксперты отмечали на видео моменты появления и исчезновения нарушителя. Аналогично были отмечены ситуации для потенциальных ложных срабатываний, что позволило эффективно классифицировать ошибки в процессе отладки. Экспертная разметка или *метаданные*, были записаны для каждого видеосюжета в формате XML для гибкого программирования текстов при помощи скриптовых языков.

Настройки алгоритмов, за исключением, калибровки глубины и области интереса, для всех видеосюжетов были идентичны. Не допускалась «подгонка» параметров алгоритмов, таких как чувствительность, под конкретные видеосюжеты.

Видеодетектор должен был зарегистрировать нарушение в течение 10 секунд. Опозданием считалось пропуском нарушения, то есть ложноотрицательным срабатыванием (счетчик c).

Повторное детектирование после разрыва траектории считалось ложноположительным срабатыванием (счетчик b). Таким образом, настоящий тест позволил оценить качество не только детектора, но и системы сопровождения.

Внутреннее тестирование показало идентичные результаты на платформе x86 и на сигнальном процессоре:

- Число истинноположительных срабатываний (нет ошибки): $a = 432$
- Число ложноположительных срабатываний (ошибка I рода): $b = 0$
- Число ложноотрицательных срабатываний (ошибка II рода): $c = 0$

Точность по сценарию «стерильная зона» рассчитана по методике i-LIDS [10]. Расчет приведен в табл. 1. Значения взвешенной средней точности для ролей оперативная тревога и записи событий совпали и оказались идеальными: $F_1=1.000$.

Видеоаналитика так же прошла внешнее независимое тестирование на другом наборе видеосюжетов, неизвестном разработчикам. Значения взвешенной средней точности для ролей оперативная тревога и записи событий составили $F_1=0.997$.

Значение точности, усредненное по результатам внутреннего и внешнего тестирования, составляет $F_1=0.999$.

6. ЗАКЛЮЧЕНИЕ

В статье представлен пример успешной адаптации нейробиологического механизма живого организма к интеллектуальным устройствам охранного телевидения. Разработаны и внедрены новые алгоритмы видеоанализа для детектирования уличного движения, сегментирования гибридного сопровождения.

Отличительной особенностью разработанных алгоритмов является использование многомасштабных признаков в виде пирамиды. Применение сразу нескольких пирамид признаков позволило одновременно отказаться от ресурсоемких алгоритмов многомодального вероятностного моделирования фона и повысить точность детектора.

Точность автоматического распознавания ситуаций на оборудовании серийного производства по сценарию «стерильная зона» составляет $F_1=1.000$ по результатам внутреннего испытания и $F_1=0.997$ по результатам независимого испытания на базе методики i-LIDS.

Перспективными направлениями для дальнейшей работы являются исследование композитивных признаков и реализация на базе них более точной классификации объектов.

7. ЛИТЕРАТУРА

- [1] Fredrik Nilsson. Intelligent network video. Understanding modern video surveillance systems, CRS Press, 2009
- [2] Птицын Н.В. *Встроенная видеоаналитика: ближайшие перспективы*, Системы безопасности, №2, 2010, с.80-83, <http://www.secuteck.ru/imag/ss-2-2010/>
- [3] Massimo Piccardi, Background subtraction techniques: a review, IEEE International Conference on Systems, Man and Cybernetics, 2004, p. 3099-3104, <http://www.utsydney.cn/www-staffit/~massimo/BackgroundSubtractionReview-Piccardi.pdf>
- [4] Bernd Jähne. Digital image processing, 5th revised and extended edition, Springer, 2002, <http://books.google.com/books?id=qUeecNvfno0C&lpg=PP1&dq=Bernd%20J%C3%A4hne.%20Digital%20image%20processing&pg=PP1#v=onepage&q&f=false>
- [5] Maximilian Riesenhuber and Tomaso Poggio, *Neural mechanisms of object recognition*, Current opinion in neurobiology, 12, 2002, p. 162–168, <http://cbcl.mit.edu/projects/cbcl/publications/ps/nb120204.pdf>
- [6] Duane G. Albrecht, Wilson S. Geisler, Robert A. Frazor and Alison M. Crane, *Visual cortex neurons of monkeys and cats: temporal dynamics of the contrast response function*, Journal Neurophysiology, 88, 2002, p 888–913, <http://jn.physiology.org/cgi/content/abstract/88/2/888>
- [7] Parisa Darvish Zadeh Varcheie, Michael Sills-Lavoie and Guillaume-Alexandre Bilodeau, *A Multiscale region-based motion detection and background*, Sensors, №10, 2010, ISSN 1424-8220, <http://www.mdpi.com/1424-8220/10/2/1041/pdf>
- [8] Eitan Sharon, Meirav Galun, Dahlia Sharon, Ronen Basri and Achi Brandt, *Hierarchy and adaptivity in segmenting visual scenes*, Nature, Vol. 442, August 2006, p. 810-813,

<http://www.wisdom.weizmann.ac.il/~meirav/nature04977.pdf>

- [9] PETS: Performance evaluation of tracking and surveillance, http://www.hitech-projects.com/euprojects/cantata/datasets_cantata/dataset.html
- [10] i-LIDS User guide imagery library for intelligent detection systems, Publication №28/08 v2.0, Home Office Scientific Development Branch, p. 25-34.

Об авторе

Птицын Николай Вадимович (np@synesis.ru), выпускник МГТУ им. Н.Э. Баумана, кандидат технических наук, основатель компании «Синезис», специализирующей в области систем охранного видеонаблюдения со встроенной аналитикой.

Embedded video analytics for object detection and tracking using multiscale features

Abstract

Novel video analytics algorithms are presented enabling embedded motion detection and object tracking for CCTV systems.

The motion detection algorithm is based on the neurobiological mechanism of the primary visual cortex V1. A sequence of simple pixel operations is used including linear operators (weighted sum) and nonlinear operator (max, saturation).

The object tracking algorithm is a hybrid of two approaches: (1) time series analysis of motion detector regions and (2) space correlation between the current frame features and object model features.

Unique advantages of the present analytics pipeline include the efficiency on the high definition (HD) video stream and the ability to track low contrast overlapping objects against a dynamic background.

The embedded video analytics is implemented and deployed on different platforms including the Texas Instruments's DSP.

A comprehensive testing environment was setup used to estimate the overall performance of the video analytics implementations. A fully embedded implementation on DSP has been i-LIDS approved both as a primary detection system for operational alert use and as an event based recording system in sterile zone monitoring applications.

Keywords: *embedded video analytics, primary visual cortex V1, multiple scale features, motion detection, object tracking, i-LIDS.*

About the author

Dr Nikolai Ptitsyn (np@synesis.ru) has been graduated from Moscow State Technical University n/a N.E. Bauman and has cofounded Synesis focusing on video surveillance solutions with embedded analytics.

Распределенная система видеомониторинга для интеллектуального пространства

Ронжин Ал.Л., Прищепа М.В., Будков В.Ю., Карпов А.А., Ронжин А.Л.

Лаборатория речевых и многомодальных интерфейсов

Санкт-Петербургский институт информатики и автоматизации РАН (СПИИРАН), Санкт-Петербург, Россия
{ronzhinal, prischepa, budkov, karpov, ronzhin}@iias.spb.su

Аннотация

В статье представлена система видеомониторинга, основанная на наборе распределённых видеокамер, и предназначенная для автоматической регистрации участников различных научных мероприятий, проводимых в интеллектуальном пространстве (smart space). Проблема определения местоположений и позиций активных участников в разработанном интеллектуальном зале, а также детектирования их лиц, решается с применением широкоугольных, стационарных видеокамер и видеооборудования с функциями наклона, поворота (pan/tilt/zoom - PTZ) и масштабирования, установленных на стенах и потолке зала. В ходе экспериментов производилась оценка качества автоматического фотодокументирования участников лекций, а также точность наведение PTZ-камер на лица людей. Смещение лиц относительно центра кадра (ошибка наведения) в среднем составило около 8%, в среднем лицо занимает не менее 7% изображения видеокadra. Эксперименты показали, что среднее время автоматической многомодальной регистрации участников мероприятия возрастает с увеличением числа участников и уменьшением уровня освещённости. Дальнейшая разработка системы направлена на развитие системы поддержки проведения мероприятия, включая поддержку выступлений, дискуссии, голосования, а также иных ключевых частей мероприятий.

Ключевые слова: видеомониторинг, компьютерное зрение, интеллектуальное пространство, отслеживание

1. ВВЕДЕНИЕ

В связи с быстрым развитием широкоэшелательных сетей, достижениями в обработке и передаче мультимедийной информации, появлением Интернет приложений для телеконференций и дистанционного образования, всё более популярными в бизнесе, академических институтах и университетах при проведении распределённых мероприятий становятся так называемые системы E-meeting и E-lecture. Такие системы позволяют экономить на транспортных расходах, выбирать индивидуальный способ обучения, а также предоставляют удобные средства поиска и доступа к информации разной природы. Технологические достижения в области записи, анализа, поиска и передачи мультимедийных данных уже сейчас используются в ряде исследовательских систем для проведения телемероприятий (распределённых совещаний, лекций, конференций и т.д.). Далее попытаемся проанализировать круг проблем, сдерживающих развитие средств дистанционного обучения и совместной работы, а также методы их решения, реализованные в исследовательских телесистемах.

Распределённые мероприятия, организованные посредством специализированных веб-приложений, становятся реальной альтернативой традиционным совещаниям и лекциям, где участники находятся “лицом к лицу”. С развитием

мультимедийных технологий телемероприятия выполняют не только функции ввода/вывода аудио- и видеоданных, но также используют более сложные технологии, такие как: интеллектуальный анализ, распознавание и структурирование мультимедийных данных. Все это, несомненно, расширяет возможности участников и приводит к появлению новых способов доступа к мероприятиям в режиме реального времени, а также возможностей для обработки архивных мультимедийных записей [1]. Значительная часть мировых исследований, проводимых в данной области, посвящена развитию программно-аппаратных средств аудио- и видеообработки. Сейчас активно используются панорамные камеры, планшеты для рукописных записей, электронные доски (whiteboard), средства для мультимедийных презентаций, технологии синхронизации и объединения данных, а также интеллектуального анализа поведения и способов взаимодействия участников мероприятий.

Автоматический анализ аудиовидеозаписей совещаний или лекций осложняется тем, что такие мероприятия обычно проводятся в больших залах с большим числом участников, которые произвольно меняют свое положение. Чтобы обеспечить захват и слежение за наибольшим числом участников, применяются массивы микрофонов, панорамные камеры, PTZ-камеры с функциями наклона, поворота и масштабирования, распределённые системы видеокамер.

Так, в системе Cornell Lecture Browser [2] для видеозаписи используются два сенсора с последующей синхронизацией и объединением видеопотоков. В проекте eClass, также известном как Classroom 2000 [3], видеозаписи лекций комбинируются с рукописными записями на электронной доске. Например, система AutoAuditorium [4] обеспечивает слежение за диктором с помощью PTZ-камеры. Кроме того, система производит также автоматический выбор видеопотока от одного из трёх сенсоров, установленных в аудитории. В работе [5] предлагается также использовать пространственную локализацию источников звука (речь участников) для первичного наведения PTZ-камер и слежения за выступающими и слушателями, а также определён список правил и рекомендаций, применяющихся на практике профессиональными операторами, соблюдение которых помогает выбрать оптимальное расположение сенсоров в зале. В работе [6] управление восьмью PTZ-камерами осуществляется посредством интеллектуального анализа видеопотока, определения положения источника звука и данных от ультразвуковых датчиков движения. В системе FlySpec [7] используются одновременно PTZ- и всенаправленные камеры, управление которыми может осуществляться одновременно несколькими удалёнными участниками. При этом автоматическая система управления корректирует углы наведения PTZ-камер, чтобы удовлетворить командам максимального числа пользователей. Применение панорамных камер также позволяет вести съёмку всех событий, происходящих в зале, и отслеживать местоположение каждого участника [8].

2. СИСТЕМА ВИДЕОМОНИТОРИНГА УЧАСТНИКОВ МЕРОПРИЯТИЙ

В 2008-2010 годах в институте разрабатывается зал (натурная модель интеллектуального пространства), в котором изучаются вопросы человеко-машинного взаимодействия в различных ситуациях. Разработанный интеллектуальный зал представляет собой распределенную систему, которая содержит сеть интеллектуальных агентов (аппаратно-программных модулей), активационных устройств, мультимедийных средств и аудиовизуальных сенсоров. Основная задача зала — обеспечение участников мероприятий необходимыми сервисами на основе автоматического анализа текущей ситуации в зале и окружающей обстановки[9].

Наличие многомодального пользовательского интерфейса является отличительной характеристикой созданного интеллектуального пространства. Разработанные технологии для автоматической обработки аудиовизуальных данных были успешно внедрены в зале для обеспечения естественного взаимодействия с оборудованием. Среди наиболее важных технологий, применённых в интеллектуальном пространстве, следует отметить: автоматическое распознавание русской речи, голосовую

идентификацию диктора, локализацию источников звука, определение положения и слежение за двигающимися объектами и лицами людей, определение позы человека.

На рисунке 1 представлена технологическая сеть интеллектуального зала. Совместная работа различных технологий снабжает систему управления зала данными о текущей обстановке в помещении, о поведении пользователей, а также обеспечивает робастное распознавание голосовых команд за счет анализа пространственно-временной, ситуативной информации и предпочтений пользователей.

Для проведения телеконференций выполнена модификация расположения оборудования внутри интеллектуального зала. При этом учитывались эргономические аспекты использования мультимедийного оборудования, а также выбрано такое расположение аудио- и видеозаписывающих устройств, которое обеспечивает захват наибольшего пространства. При выборе количества камер, их месторасположения, разрешающей способности, угла обзора объективов и иных параметров системы видеомониторинга, учитывались размеры зоны охвата, максимальное число объектов, требующих одновременного слежения, степень детализации анализируемых объектов, уровень освещенности и т.д.



Рисунок 1: Технологическая сеть интеллектуального пространства с распределенной системой видеомониторинга

На рисунке 2 представлена схема расположения посадочных мест в зале; мультимедийного оборудования (ТВ, проектор), интернет-камер AXIS (PTZ-камеры Cam3, Cam5; беспроводные WiFi сенсоры Cam2, Cam4; видеокамера с

широкоугольным объективом Cam1, установленная на потолке по центру зала); 10-ти персональных веб-камер Logitech AF Sphere, установленных на конференц-стол;

четырёх групп освещения. Основное внимание в данной статье уделяется описанию разработанной и внедренной системы видеомониторинга, поэтому на рисунке не показаны четыре Т-образных массива микрофонов, расположенных на каждой из стен. Модельно-алгоритмическое обеспечение для многоканальной обработки аудиопотоков, использованное при разработке интеллектуального пространства представлено в работе [9].

Для размещения участников небольших совещаний (круглых столов до 10 человек) в левой части зала расположен конференц-стол с установленными на нем персональными

видеокамерами. В правой части зала расположены ряды кресел, на которых могут разместиться до 32 участников конференций или лекций, слежение за которыми реализовано посредством распределенной системы видеокамер и массивов микрофонов. С учетом указанного расположения оборудования, в зале выделены три непересекающиеся зоны (см. области, показанные на рисунке 2 синим цветом), требующие постоянного видеослежения: (1) зона перемещения основного докладчика; (2) зона участников, сидящих за конференц-столом; (3) зона участников, сидящих в рядах кресел.

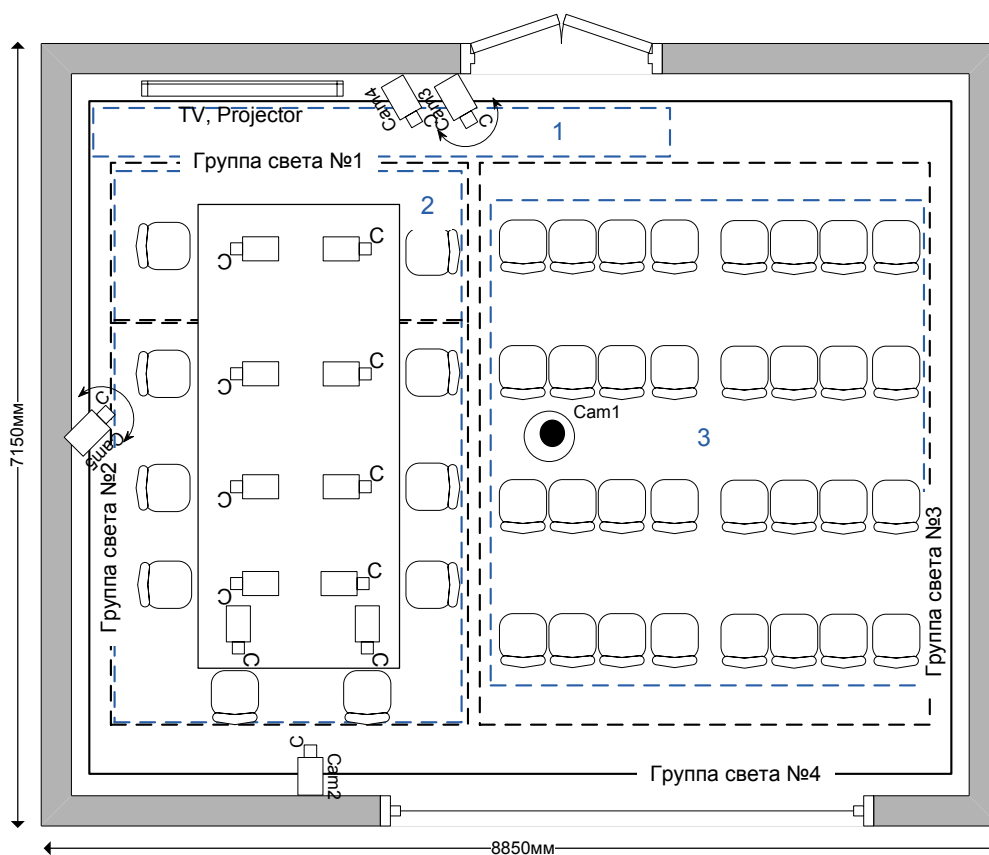


Рисунок 2: План-схема интеллектуального зала с расположением видеокамер.

Видеомониторинг всех трех зон осуществляется с применением видеокамеры Cam1, обладающей углом обзора 140 градусов и разрешающей способностью 2048x1536 пикселей. Стационарные видеокамеры Cam2 и Cam4 имеют угол обзора 74 градуса и разрешающую способность 1280x1024 пикселей. Две камеры Cam3 и Cam5 с углами обзора 52 градуса и разрешающей способностью 704x576 пикселей, оснащенные функциями наклона, поворота и масштабирования, обеспечивают наведение на любую точку пространства в зале.

Видеокамера Cam2 установлена на высоте 2 метров на противоположной от проекционного экрана стене и используется для аудиовидеозаписи хода мероприятия. Cam1 имеет широкоугольный объектив, и расположена на потолке в центре зала, поэтому она одновременно охватывает все пространство зала и позволяет определить координаты всех участников в горизонтальной плоскости. Стационарная камера Cam4 установлена слева от входной двери и

направлена таким образом, чтобы охватывать сразу всех пользователей, сидящих в зоне кресел. PTZ-камера Cam3 и стационарная видеокамера Cam2 установлены рядом и в большинстве случаев работают синхронизированно. Анализируя видеокadres, поступающие от Cam2, определяется положение наблюдаемого объекта и формируется управляющая команда для наведения Cam3 в направлении, необходимом для захвата лица определенного участника, находящегося в зоне кресел. PTZ-камера Cam5 установлена по центру левой стены зала на высоте 2 метра от уровня пола и служит, в основном, для захвата видеоизображения выступающего докладчика.

Разработанная многофункциональная система видеомониторинга интеллектуального зала на основе системы IP-видеокамер AXIS позволяет осуществлять видеозапись мероприятия, а также автоматизировать такие ключевые этапы совещания, как регистрация участников, съемка выступающего и активных участников в аудитории во

время дискуссии и т.д. Исходя из основных этапов предполагаемых мероприятий, были составлены различные режимы работы распределенной системы видеомониторинга.

В таблице 1 показаны функции каждого видеосенсора в пяти основных режимах: наблюдение, слежение за участниками, регистрация участников, выступление докладчика, дискуссия.

Камера (Cam)	Режимы работы системы				
	Наблюдение	Слежение за участниками	Регистрация участников	Сопровождение мероприятия	
				Выступление докладчика	Дискуссия
1	Слежение за изменением состояния буферной зоны	Слежение за перемещениями участников и определение сидящих людей		Слежение за перемещением выступающего и определение участников, сидящих в зоне кресел	
4	-	Поиск лиц сидящих участников			
3	-	Фотографирование лиц сидящих участников		Запись выступлений людей, сидящих в зале	
5	-	Наведение и запись перемещающегося участника		Наведение и запись выступления основного докладчика	
2	-	Запись общего вида на аудиторию			

Таблица 1: Функции видеокамер в различных режимах работы системы видеомониторинга интеллектуального зала.

В режиме «Наблюдение» выполняется обработка буферной зоны в видеокадре от Cam1 для детектирования входящих в зал людей. Как только обнаружен хотя бы один участник, автоматически запускается режим «Слежение за участниками», в котором производится обработка видеокадров от Cam1 для определения координат данного участника в осях зала и вычисления углов наклона и поворота камеры Cam5, которая ведет запись перемещения участника. Также осуществляется мониторинг зоны кресел для определения сидящих участников. Как только хотя бы один из участников сядет в любое кресло, система видеомониторинга переходит в режим «Регистрация участников», в котором задействованы сразу три видеокамеры: Cam1 осуществляет определение сидящих участников; Cam4 дополнительно проверяет наличие сидящих людей путем детекции лиц в зонах, закрепленных за номерами кресел, где выявлены участники; Cam3 фотографирует крупным планом лица обнаруженных сидящих участников.

Для основной части мероприятия определены два режима работы системы: «Выступление докладчика» и «Дискуссия». В первом режиме выполняется видеослежение за перемещением выступающего и детекция участников, сидящих в зоне кресел. При этом наведение и запись выступления основного докладчика выполняется посредством Cam5. В режиме «Дискуссия» используется Cam3 для записи выступающих, сидящих в зале. Во всех режимах работы, кроме «Наблюдение», Cam2 осуществляет запись общего вида интеллектуального зала. В таблице 1 не представлен режим «Ожидание», в котором камеры находятся в спящем состоянии, до тех пор, пока не будет включена, по крайней мере, одна группа света. Об изменении состояния активаторов, отвечающих за работу осветительного оборудования, система видеомониторинга

узнает из файла-журнала, в котором хранятся параметры, описывающие текущую ситуацию в интеллектуальном зале.

На рисунке 3 приведена укрупненная блок-схема алгоритма выбора текущего режима системы видеомониторинга. Вначале считывается файл-журнал, содержащий данные о состоянии групп света в зале, и если свет выключен, устанавливается режим «Ожидание». Напротив, при включенном свете проверяется наличие участников в зале, и если они отсутствуют, то система переходит в режим «Наблюдение». В случае нахождения участников, выполняется проверка, есть ли сидящие в зоне кресел, и если таковых нет, запускается режим «Слежение за участниками». В противном случае проверяется наличие человека в зоне выступления, и если он не обнаружен, то анализируется, в какой фазе находится мероприятие в данный момент. Если основная часть мероприятия (режимы «Выступление докладчика» и «Дискуссия») уже начата, то включается режим «Пассивная регистрация». Если же идет подготовительная стадия, то система переходит в режим «Активная регистрация». При наличии основного докладчика в зоне выступлений проверяется посредством метода аудиолокализации и массива микрофонов, есть ли говорящие среди зарегистрированных сидящих участников, и если нет, то устанавливается режим «Выступление», в обратном случае система переходит в режим «Дискуссия».

Для удобства наблюдения за режимами работы вся информация передается системе управления интеллектуальным залом, в диалоговом окне которой отображаются обнаруженные в процессе видеомониторинга объекты, в том числе участники, сидящие за конференц-столом, в зоне кресел, а также перемещающиеся в настоящий момент. Фотографии зарегистрированных участников отображаются в диалоговом окне в соответствии с занимаемыми местами в креслах.

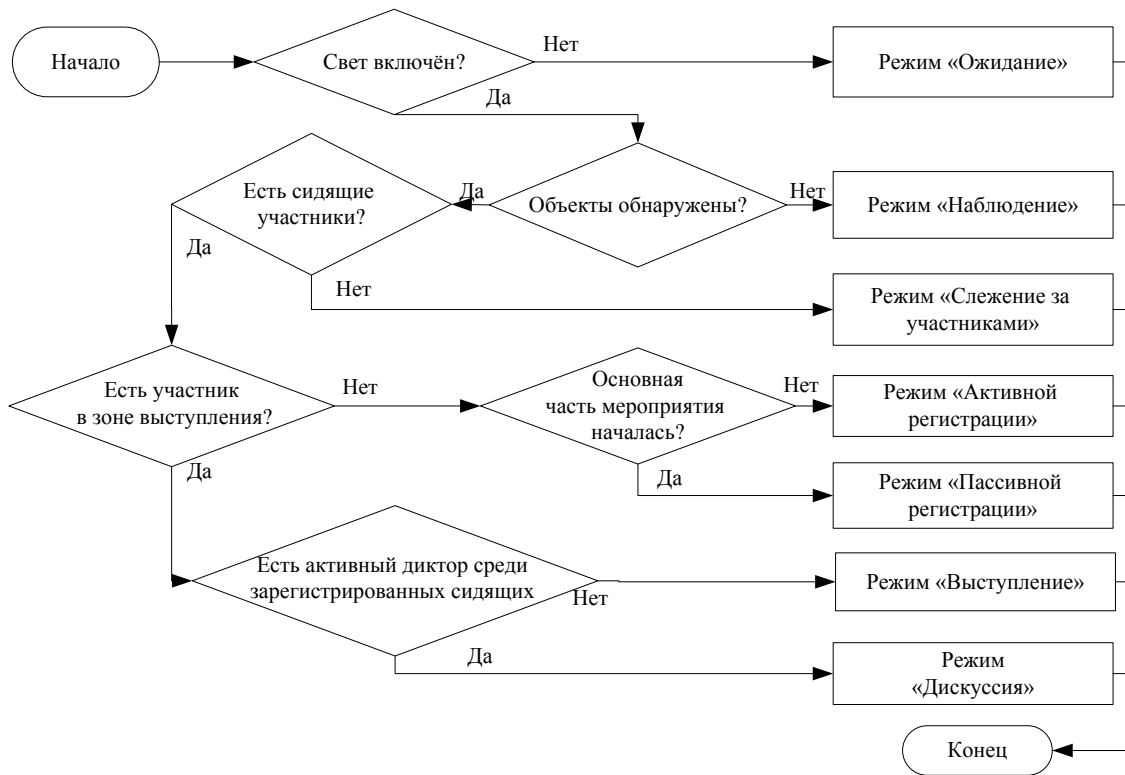


Рисунок 3: Блок-схема алгоритма выбора текущего режима работы системы видеомониторинга

3. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

В ходе серии предварительных экспериментов было организовано восемь небольших мероприятий и проведена регистрация участников, всего было зарегистрировано 106 участников. После ручной проверки было обнаружено 6 фотографий, на которых лица были детектированы неверно. Ошибки связаны, в основном, с изменениями положения головы участника в процессе фотографирования. Остальные 100 фотографий были использованы для подсчета скорости (времени) и качества фотодокументирования системы в режиме «Регистрация».

На рисунке 4 показан пример работы системы в режиме «Регистрация», где (1), (2) и (3) – кадры, выполненные видеокамерами Cam1, Cam4, Cam3, соответственно. С помощью Cam1 алгоритмом детекции движения [10] найдено 6 участников, сидящих в зоне кресел (один ложный объект был определен как участник). Номера этих кресел далее используются системой для анализа соответствующих

интересующих регионов на видеокдрах от Cam4. В данном примере анализ кадра (используется детектор лиц методом AdaBoost [11]) показал наличие четырех лиц в шести зонах, закрепленных за вышеуказанными креслами, в двух зонах лица не были найдены в связи с тем, что участники не следили за ходом мероприятия и не смотрели на лектора. Результирующий список номеров кресел затем передается на Cam3, которая последовательно устанавливается таким образом, чтобы крупным планом сфотографировать лицо каждого участника с необходимым качеством. Так как участники произвольно изменяют положения своего тела и головы, то при фотографировании производится запись ограниченной серии видеок кадров и проверяется наличие лица в нем методом [11]. Снимок сохраняется в базе данных регистрации только после обнаружения на нем графического объекта с параметрами, удовлетворяющими модели лица.

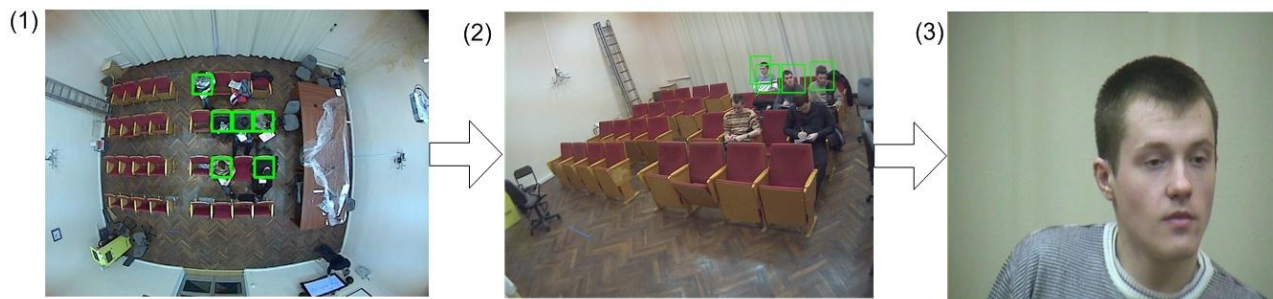


Рисунок 4: Пример работы системы в режиме «Регистрация».

Качество фотографий (кадры участников мероприятия крупным планом), сделанных системой в режиме «Регистрация», зависит, во многом, от точности наведения PTZ-камеры Cam3 и может оцениваться по двум критериям: (1) точность наведения видеокамеры на лицо; (2) отношение размера обнаруженного лица к размеру фотографии. Первый критерий показывает, насколько точно были выбраны углы поворота и наклона PTZ-камеры, а второй отвечает за правильность расчета коэффициента масштабирования. На каждой стадии обработки вычислялось относительное отклонение D центра лица от центра кадра (по двум осям). Для того чтобы избежать влияния изменения масштаба,

величина смещения определяется в процентах относительно полного размера видеокadra.

Относительный размер R обнаруженного в кадре лица вычисляется как отношение диагонали лица к диагонали кадра. В случае, если лицо в кадре не было обнаружено системой, то величина отклонения D выставляется равной 100%, а относительный размер лица - 0%. Кроме относительных размеров лица и отклонения лица, также вычислялась относительная и абсолютная площадь лица, S и A, соответственно. Кроме того, было оценено среднее время автоматической регистрации участника мероприятия. В таблице 2 приведены результаты оценки качества полученных системой фотографий.

R, %			D, %			S, %			A, Mpixels			T, c		
Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
24	47	35	0	28	11	6	23	13	0.03	0.10	0.06	0.03	11.59	1.08

Таблица 2: Количественная оценка качества фотографий участников мероприятий, сделанных системой в режиме «Регистрация»

4. ЗАКЛЮЧЕНИЕ

Слежение за подвижными объектами является сложной и важной задачей, решение которой позволит, в частности, автоматизировать проведение совещаний, телеконференций и других мероприятий. В разработанной системе видеомониторинга применены и исследуются технологии поиска и слежения за человеком, детекции позы и положения лица, позволяющие обеспечить информационное и техническое сопровождение научно-образовательных мероприятий, в том числе дистанционных. Разработанная многофункциональная система видеомониторинга работает в четырех основных режимах: (1) наблюдение, (2) слежение за участниками, (3) регистрация, (4) сопровождение мероприятия. На данный момент все режимы работы были протестированы по качественным показателям в ходе лекционных занятий и совещаний в интеллектуальном зале. Особое внимание уделено режиму автоматической регистрации участников. Было проанализировано влияние уровня освещения, числа участников и их местоположения на точность, время поиска лиц и последующее фотографирование участников. На следующем этапе исследования будут проанализированы математические и программные средства распознавания простейших жестов, выполняемых руками, и на их основе предполагается разработка подсистемы автоматического подсчета количества голосов, в мероприятиях, когда голосуют поднятием рук.

5. БЛАГОДАРНОСТИ

Данное исследование поддержано Министерством образования и науки РФ в рамках ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009-2013 годы (госконтракт № П2360), а также фондом РФФИ (гранты № 10-08-00199-а и № 09-07-91220-СТ_а).

6. СПИСОК ЛИТЕРАТУРЫ

- [1] V. Erol, Y., Li, An overview of technologies for e-meeting and e-lecture. Proc. *IEEE International Conference on Multimedia and Expo*, pp. 6–12, 2005.
- [2] S. Mukhopadhyay, B. Smith, Passive capture and structuring of lectures. *ACM Multimedia*, pp. 477–487, 1999.

- [3] J.A. Brotherton, J.R. Bhalodia, G.D. Abowd, Automated capture, integration, and visualization of multiple media streams. *IEEE Multimedia*, pp. 54–63, 1998.
- [4] M. Bianchi, AutoAuditorium: A fully automatic, multicamera system to televise auditorium presentations. *Proc. Joint DARPA/NIST Smart Spaces Tech. Workshop*, 1998.
- [5] Y. Rui, A., Gupta, J. Grudin, L. He, Automating lecture capture and broadcast: Technology and videography. *ACM Multimedia Systems Journal*, pp. 3–15, 2004.
- [6] Y. Kameda, S. Nishiguchi, M. Minoh, CARMUL: Concurrent automatic recording for multimedia lecture. *Proc. ICME*, 2003.
- [7] Q. Liu, D. Kimber, J. Foote, L. Wylcox, J. Boreczky, FLYSPEC: A multi-user video camera system with hybrid human and automatic control. *ACM Multimedia*, pp. 484–492, 2002.
- [8] D.S. Lee, B.J. Erol, Graham, J.J. Hull, N. Murata, Portable meeting recorder. *ACM Multimedia*, pp. 493–502, 2002.
- [9] P.M. Юсупов, А.Л. Ронжин. От умных приборов к интеллектуальному пространству // *Вестник Российской Академии Наук: научный и общественно-политический журнал*, Том 80, Вып. 1, С. 45–51, 2010.
- [10] J. Davis and G. Bradski, Real-time motion template gradients using Intel CVLib, Proc. *ICCV Workshop on Frame-Rate Vision*, Corfu, Greece, 1999.
- [11] P. Viola and M. J. Jones, Rapid Object Detection Using a Boosted Cascade of Simple Features, Proc. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Kauai, HI, USA, 2001.

ОБ АВТОРАХ

Ронжин Александр Леонидович – студент Санкт-Петербургского государственного университета аэрокосмического приборостроения (СПбГУАП), программист Санкт-Петербургского института информатики и автоматизации Российской академии наук (СПИИРАН), ronzhinal@iias.spb.su

Прищепа Мария Викторовна – студент Санкт-Петербургского государственного университета аэрокосмического приборостроения (СПбГУАП), программист Санкт-

Петербургского института информатики и автоматизации Российской академии наук (СПИИРАН), prischepa@iias.spb.su

Будков Виктор Юрьевич – студент Санкт-Петербургского государственного университета аэрокосмического приборостроения (СПбГУАП), программист Санкт-Петербургского института информатики и автоматизации Российской академии наук (СПИИРАН), budkov@iias.spb.su

Карпов Алексей Анатольевич – к.т.н., с.н.с. лаборатории речевых и многомодальных интерфейсов Санкт-Петербургского института информатики и автоматизации Российской академии наук (СПИИРАН), karpov@iias.spb.su

Ронжин Андрей Леонидович – к.т.н., зав. лабораторией речевых и многомодальных интерфейсов Санкт-Петербургского института информатики и автоматизации Российской академии наук (СПИИРАН), ronzhin@iias.spb.su

A DISTRIBUTED VIDEO SURVEILLANCE SYSTEM FOR THE SMART SPACE

Abstract

This paper presents the video-monitoring system based on a distributed camera system, which is intended for automation of participants registration of scientific meetings carried out in a smart space model. The problem of detecting positions of participants in the developed smart room and their faces is solved by usage of a complex of wide-angle, omni-directional and PTZ video-cameras located on the walls and the ceiling of the room, as well as application of the face detection technology for photographing zones of probable locations of participants. The quality of photographing was evaluated using the accuracy measure of pointing of PTZ video-cameras and size of a detected face in the captured image. Deviation of the face region relatively to the frame center was estimated as 8% in average and the face region occupies at least 7% of the frame area. It was observed also that mean time of the registration increases, when the number of participants increases and the illumination level decreases. Further development of the system is aimed for automatic detection of an active speaker, discussion, voting and other key phases of the scientific events.

Keywords: video surveillance, computer vision, smart space, human detection and tracking

Отслеживание контура лабораторной мыши в видеопоследовательности на основе метода ветвей и границ

Михаил Синдеев¹, Антон Конушин²

¹Институт прикладной математики им. М.В.Келдыша РАН

²Лаборатория компьютерной графики и мультимедиа ВМК МГУ им. М.В. Ломоносова

E-mail: m_sindeev@mail.ru, ktosh@graphics.cs.msu.ru

Аннотация

В данной статье рассматривается задача отслеживания контура лабораторных животных в видеопоследовательности на основе заданной модели формы и предлагается алгоритм для решения данной задачи, основанный на методе ветвей и границ.

Ключевые слова: видеорегистрация, слежение, сегментация, вычитание фона

1. ВВЕДЕНИЕ

При проведении экспериментальных исследований на лабораторных животных важной задачей является количественная оценка поведения. Для этого поведение регистрируется с помощью видеокамеры, затем по полученной видеопоследовательности отслеживается траектория движения животного и для каждого момента времени определяется ряд характеристик тела животного, такие как поворот и изгиб. На основе этих данных проводится анализ поведения, например, поведение разбивается на отдельные поведенческие акты, вычисляется общая длина пути, пройденного животным, скорость движения животного, время нахождения в выделенных функциональных зонах и т.д.

Стандартная экспериментальная установка состоит из огороженной арены и жестко закрепленной над ней видеокамеры. В рассматриваемом семействе экспериментов одновременно на арене может находиться не более одного животного.

В этой статье рассматривается задача отслеживания контура лабораторного животного по видеопоследовательности, полученной со стандартной экспериментальной установки. Для каждого кадра видеозаписи необходимо построить маску животного – бинарное изображение, в каждом пикселе которого метка «0» соответствует фону, а «1» – объекту. Также необходимо найти определенные ключевые точки ([1]), которые используются при анализе поведения животного.

Алгоритмы отслеживания лабораторных животных можно разделить на две группы – онлайн и оффлайн трекинг. При онлайн-трекинге построение маски требуется в реальном времени, т.е. скорость работы программы должна соответствовать скорости поступления кадров видео. Онлайн-трекинг требуется, например, в случае, когда программа должна управлять механизмами арены в зависимости от действий животного (например, открывать/закрывать дверцы в лабиринте, когда животное совершает нужное действие).

При оффлайн-трекинге обрабатывается заранее полученная видеопоследовательность и нет жестких ограничений на скорость работы, но требуется хранить всю видеопоследовательность на внешних запоминающих устройствах.

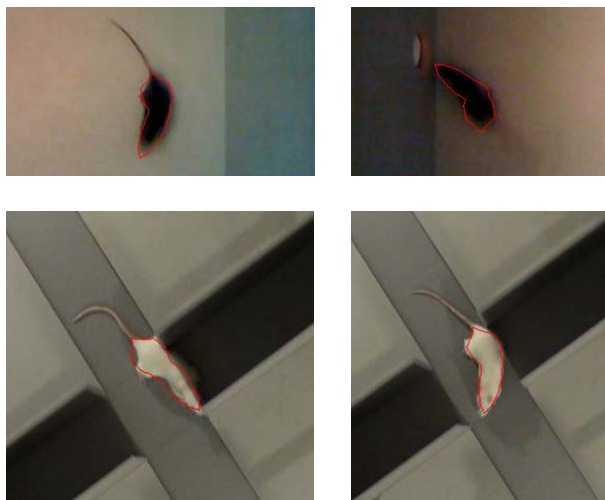


Рисунок 1. Результат работы алгоритма. Показаны фрагменты кадров видеопоследовательностей с наложением найденного контура мыши.

В данной статье предлагается алгоритм для задачи высокоточного оффлайн-трекинга. Он также может быть применен для генерации эталонных данных для тестирования других алгоритмов онлайн- и оффлайн-трекинга.

2. ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ

Стандартный подход при отслеживании лабораторных животных – попиксельное вычитание фона ([2]). Такой метод применяется в коммерческих системах [1], [3].

Существуют также расширения этого метода для динамического фона, например [4], где для повышения устойчивости к динамическому фону используется моделирование окрестности пикселя (гистограмма цветов).

Попиксельные методы дают шумную маску и не устойчивы к теням (которые, по условию задачи, алгоритм должен относить к фону, но часто не справляется с этим, так как тень имеет отличный от фона цвет). Для фильтрации которой обычно применяются эвристические методы на основе математической морфологии, которые требуют ручной подгонки радиуса фильтра для каждой видеопоследовательности (или даже для ее фрагментов).

Для нахождения опорных точек на контуре животного (нос, основание хвоста) и постобработки маски может быть использован метод на основе активных моделей формы, но он оперирует с бинарной маской, поэтому часто ошибочно присоединяет тень к контуру животного (**Error! Reference source not found.**). Аналогичные проблемы при использова-

нии кинематических моделей [6]. Был предложен ряд алгоритмов, использующих глобальную оптимизацию и учитывающих контраст (например, [7]), но они по-прежнему не устойчивы к теням, т.к. граница тени может быть довольно резкой.

Для более устойчивого трекинга необходимо на этапе построения маски использовать информацию о форме животного. Для сегментации объектов с использованием априорной информации о форме в статичных изображениях был предложен метод ветвей и границ. Этот метод уже применялся для сегментации изображений [8] при помощи разрезов графов, однако в таком варианте не подходит для видеопоследовательностей, так как работает слишком долго.

3. АЛГОРИТМ

Мышь моделируется контуром заданной формы с четырьмя степенями свободы: две компоненты сдвига (в плоскости), поворот, изгиб. Такой набор параметров соответствует реальным степеням свободы рассматриваемых животных (при этом

остальные степени свободы совершают пренебрежимо малый вклад в видимое положение животного), а также позволяет получить необходимые данные для анализа – положение центра масс, плюс ключевые точки на контуре (для мыши – точка носа и точка «крепления» хвоста).

Нахождение положения маски сводится к минимизации функционала

$$E = \iint_{\Omega} L(\omega) d\omega, \quad (1)$$

где $\Omega = \Omega(cx, cy, angle, bend)$ – внутренняя область контура при конкретных значениях параметров центра (cx, cy), угла ($angle$) и изгиба ($bend$), а $L(\omega)$ – потенциал фона в данной точке ω на изображении. Если внутри некоторой области потенциал фона минимален, то данная область, скорее всего, соответствует искомому объекту.

3.1 Вычитание фона

Для нахождения потенциала фона используется вероятностная модель. Цвет каждого пикселя на изображении фона моделируется нормальным распределением. Параметры распределения – математическое ожидание и дисперсия/матрица ковариации – оцениваются на этапе захвата фона, который осуществляется перед помещением животного на арену. В процессе трекинга фон обновляется на основе алгоритма бегущего среднего, см. [2]. Условием применимости данной модели является разделение цветов объекта и фона, а также отсутствие переходных цветов (например, теней). Для выполнения этого условия цвет арены выбирается контрастным по отношению к цвету животного. Например, черные мыши снимаются на белом фоне, а белые – на темно-сером или черном. Освещение должно быть рассеянным и равномерным с разных сторон, чтобы снизить затенение областей вокруг животного. Ввиду статичности фона более сложные модели на основе смесей нормальных распределений [2] и гистограмм [4] не используются, т.к. на этих данных дают аналогичные результаты, но с большими затратами времени и памяти.

Положив потенциал равным логарифму вероятности:

$$L = \log(p) + c, \quad (2)$$

получим в формуле (1), в предположении о независимости меток всех пикселей, плотность вероятности нахождения маски животного в заданном положении (определяемом параметрами $cx, cy, angle, bend$) при $c = 0$. Выберем константу c так, чтобы все значения L были неположительными:

$$c = -\max \log(p), L \leq 0. \quad (3)$$

3.2 Поиск положения модели животного

Глобальный минимум функционала (1), очевидно, может быть найден с заданной точностью перебором всех значений параметров на четырехмерной дискретной сетки, шаг которой определяется требуемой точностью.

Однако такой перебор имеет экспоненциальную сложность и не может быть выполнен за приемлемое время.

Локальные алгоритмы трекинга используют небольшое пространство поиска в окрестности результата, найденного на предыдущем кадре (можно считать, что на первом кадре положение животного задается пользователем). Основным недостатком таких алгоритмов является низкая стабильность: ошибки отслеживания накапливаются, а сбивание алгоритма хотя бы на одном кадре приводит к неверному результату и на всех последующих кадрах.

В данной статье предлагается использовать для глобальной оптимизации функционала (1) метод ветвей и границ.

Пусть некоторая область Ω_1 вложена в область Ω_2 . Тогда

$$E_1 = \iint_{\Omega_1} L(\omega) d\omega \geq \iint_{\Omega_2} L(\omega) d\omega = E_2 \quad (4)$$

при условии неположительности значений L (3).

Здесь E_1 и E_2 не обязательно соответствуют значениям функционала (1), т.к. области Ω_1, Ω_2 не обязательно соответствуют возможным маскам для заданной модели животного. Однако данное неравенство позволяет получать нижние оценки значений функционала (1).

Рассмотрим пространство поиска для значений $cx, cy, angle, bend$: $[0, W] \times [0, H] \times [0^\circ, 360^\circ] \times [-bend_limit, bend_limit]$, где W, H – ширина и высота изображения, а $bend_limit$ – максимальный угол изгиба животного.

Пусть Ω_2 – объединение масок для некоторого пространства поиска, вложенного в исходное или совпадающее с ним (объединение масок – маска, в которой пиксель имеет метку «1», если хотя бы в одной из объединяемых масок данный пиксель имеет эту метку, и «0» в противном случае, т.е. это попиксельная дизъюнкция масок). Тогда значение функционала (1) в каждой маске Ω_1 из этого пространства удовлетворяет неравенству (4).

Метод ветвей и границ можно применить следующим образом. Поместим исходное пространство поиска в буфер. Разбивая пространство поиска пополам вдоль одной из координат $cx, cy, angle, bend$ и вычисляя верхнюю оценку по объединению масок, будем выбирать подпространство с наименьшей нижней оценкой энергии для дальнейшего разбиения, а остальные помещать в буфер вместо разбиваемого

подпространства. Выбор подпространства осуществляется среди всех подпространств в буфере. Когда выбранное подпространство будет достаточно малым, остановим разбиение и положим результат всего алгоритма равным некоторой точке этого подпространства. Таким образом, алгоритм будет спускаться от верхней оценки значения функционала (1) к глобальному минимуму этого функционала.

На реальных данных число шагов такого алгоритма полиномиальное, против экспоненциального при полном переборе.

В отличие от алгоритма **Error! Reference source not found.**, поиск положения животного осуществляется на основе карты вероятностей L , без построения предварительной грубой маски, что дает более стабильный результат (за счет того, что эта карта придает пикселям разный вес).

3.3 Уточнение поворота

При некоторых позах рассматриваемое животное (мышь) имеет симметричную форму относительно плоскости, разделяющей переднюю и заднюю половинки. Из-за этого примерно в 15% случаев найденный угол поворота имеет ошибку 180° . При малых размерах мыши ошибка проявляется ещё чаще, так как мышь становится почти округлой. Хвост нарушает симметрию, но его обнаружение затруднено из-за малой толщины при низком качестве видео (веб-камера), сильным размытием при движении хвоста из стороны в сторону и большим числом степеней свободы в рамках рассматриваемой модели на основе масок (много возможных точек изгиба).

Для исправления таких ошибок предлагается алгоритм, основанный на том, что мышь чаще движется вперед, чем назад, и движется вперед быстрее, чем назад. Математически это можно смоделировать как максимизацию скалярного произведения вектора скорости на вектор направления мыши.

Сопоставив каждому кадру t булеву переменную $f(t)$ со значениями «0» – направление выбрано правильно и «1» – направление следует изменить на 180° (и изменить знак угла изгиба $bend$) можно сформулировать оценочную функцию

$$C = \sum_{t=1}^T V(t) \cdot D(t, f(t)) + \lambda \cdot \sum_{t=2}^T D(t-1, f(t-1)) \cdot D(t, f(t)), \quad (5)$$

где $V(t)$ – вектор скорости (вычисляемый как разность координат центра (cx, cy) в соседних кадрах), $D(t, f)$ – единичный вектор направления в кадре t , в предположении, что угол равен $angle$, если $f = 0$ и $angle + 180^\circ$, если $f = 1$. T – общее число кадров в видеопоследовательности. Второе слагаемое отвечает за согласованность направлений мыши между кадрами. Такую оценочную функцию можно максимизировать методом динамического программирования [9].

Данный алгоритм применим также и к онлайн-методам трекинга, страдающим от 180-градусных ошибок на при вычисления поворота. Суммирование в формуле (5) в этом случае осуществляется до текущего кадра, а хранение значений C только для предыдущего и текущего кадров (всего 4 значения: для $f = 0, 1$ в кадре $t - 1$ и для $f = 0, 1$ в кадре t). В онлайн-варианте возможны ошибки в течение первых нескольких кадров, но затем алгоритм стабилизируется.

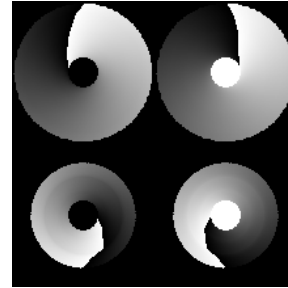


Рисунок 2. Карта поворота половинок животного (мыши). Верхний ряд: передняя часть, нижний: задняя. Карты слева определяют угол появления метки «1» в заданном пикселе, карты справа – угол исчезновения этой метки. Черный цвет в шаблонах соответствует углу 0° , белый – 360° . Черная область вне шаблонов задается специальным значением -1.

4. РЕАЛИЗАЦИЯ

В реализации пространство поиска является дискретным с шагом в 1 пиксель по координатам и 1° по углу поворота и углу изгиба. Максимальный угол изгиба $bend_limit$ равен 35° и соответствует независимому повороту верхней и нижней половинки животного в разных направлениях. Форма животного моделируется масками этих двух половинок. Маски обеих половинок имеют в точке центра круг («шарнир»), который гарантирует, что объединение половинок будет цельным объектом, достаточно гладким в месте соединения (гладкости в математическом смысле не требуется – достаточно лишь схожести с реальным контуром животного на изображении).

Разбиение пространства поиска производится по координате, имеющей наибольший диапазон. Для параметров разной размерности используется соотношение 1 пиксель $\sim 1^\circ$.

Объединение масок вычисляется сначала по углу поворота и изгиба: для каждой половинки вычисляются границы диапазона углов $[angle - bend, angle + bend]$, где $angle$ и $bend$ – также диапазоны для текущего пространства поиска. Для получения самой маски используются заранее посчитанные карты поворота каждой из половинок (Рисунок 2).

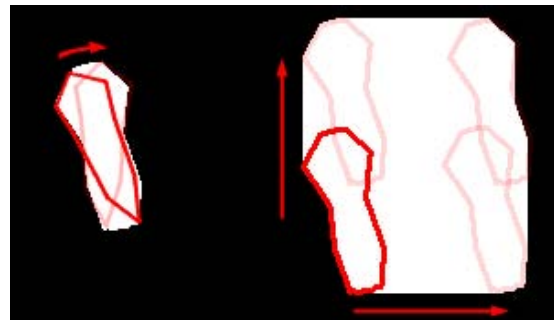


Рисунок 3. Построение объединения масок по диапазону параметров. Слева: объединение по углу, справа: последующее объединение по диапазону координат. Изгиб отсутствует ($bend = 0$).

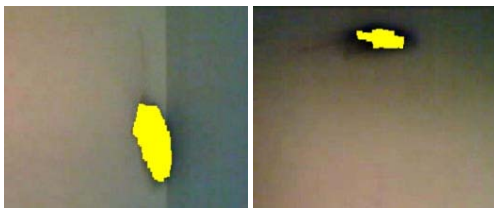


Рисунок 4. Алгоритм порога (на основе вычитания фона) может давать маску неправильной формы, т.к. в нее попадают тени и шум. Кроме того, не всегда возможно выбрать значение порога, которое давало бы правильный результат в разных областях арены и при разных условиях съемки.

Затем строится объединение маски по диапазонам координат x , y путем расположения масок, объединенных по повороту и изгибу, в углах прямоугольника и их соединения (Рисунок 3).

Предложенный алгоритм был сравнен с алгоритмом порога на основе вычитания фона (Рисунок 4) на наборе видеопоследовательностей, предоставленных ООО «НПК Открытая Наука» [10] и НИИ Нормальной физиологии им. П.К. Анохина РАМН [11]. Результат оказался более точным и стабильным, так как использует модель формы животного и находит наиболее подходящую область, в то время как пороговый алгоритм может находить область большего или меньшего размера, а также неправильной формы. Кроме того, предложенный алгоритм находит глобальный минимум функционала энергии, без какой-либо привязки к значению этого функционала. Пороговый алгоритм требует точного указания значения порога, по которому происходит классификация пикселей, однако такое значение не всегда можно выбрать оптимальным образом: например, в крестообразном лабиринте с высокими стенками вдоль двух «рукавов» лабиринта (Рисунок 5) мышь может одновременно находиться в областях разной освещенности.

5. ЗАКЛЮЧЕНИЕ

В данной статье предложен алгоритм оффлайн-трекинга лабораторных животных по видеопоследовательности, устойчивый к ошибкам обнаружения и изменению освещенности. Состояние животного в каждый момент времени описывается в виде вектора из четырех параметров: координат (x , y) центра модели животного и двух углов – поворота и изгиба. Задача сформулирована в виде поиска глобального оптимума функции и решается методом ветвей и границ.

5.1 Дальнейшая работа

Предложенный алгоритм оффлайн-трекинга на основе метода ветвей и границ планируется ускорить, сделав его применимым для онлайн-трекинга. Этого можно достичь иерархической обработкой изображения, уменьшением пространства поиска (рассматривать только небольшую окрестность результата предыдущего кадра), а также ускорением вычислительных процедур алгоритма. В частности, сумму по заданному объединению масок можно вычислять на основе интегральных изображений, учитывая, что значительную часть такого объединения составляет прямоугольник (см. Рисунок 3, справа); оставшуюся часть объединения также можно представить набором прямоугольников и горизонтальных/вертикальных отрезков.



Рисунок 5. Результаты работы предлагаемого алгоритма в сложных случаях на примере крестообразного лабиринта (показан сверху). Слева: разные условия освещенности. Справа: частичное перекрытие стенкой лабиринта.

6. БЛАГОДАРНОСТЬ

Работа выполнена при поддержке федеральной целевой программы "Научные и научно-педагогические кадры инновационной России" на 2009 - 2013 годы", проект №П264 от 23 июля 2009 г.

7. ЛИТЕРАТУРА

- [1] L. Noldus, A. Spink. EthoVision: a versatile video tracking system for automation of behavioral experiments. *Behavior Research Methods, Instruments & Computers*, 33, pp. 398–414.
- [2] M. Piccardi. Background subtraction techniques: a review. In *IEEE International Conference on Systems, Man and Cybernetics*, vol. 4. IEEE, October 2004, pp. 3099–3104.
- [3] Stoelting ANY-maze, <http://www.anymaze.com/>
- [4] T. Ko, S. Soatto, D. Estrin. Background Subtraction on Distributions. In *ECCV 2008, Part III, LNCS 5304*, pp. 276–289.
- [5] Конушин А, Ломакина-Румянцева Е, Ветров Д, Воронин П, Синдеев М. Система видеонаблюдения за поведением лабораторных животных с автоматической сегментацией на поведенческие акты. In: *GraphiCon*, 2008. pp. 199-205.
- [6] Т. Мухина, И. Зарайская, К. Анохин. Отслеживание движений животных с использованием 2D модели. *Математические методы распознавания образов (ММРО-11): Сборник докладов 11-й Всероссийской конференции*, сс. 378–380.
- [7] J. Sun, W. Zhang, X. Tang, H. Shum. Background cut. In *ECCV*, 2006, pp. 628–641.
- [8] V. Lempitsky, A. Blake, C. Rother. Image Segmentation by Branch-and-Mincut. In *ECCV 2008, Vol. 5305*, pp. 15–29.
- [9] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.
- [10] ООО «НПК Открытая Наука», <http://openscience.ru/>
- [11] НИИ Нормальной физиологии им. П.К. Анохина, РАМН, <http://nphys.ru>

К построению эффективного решения задачи пересечения отрезков

Тарас Вознюк, Василий Терешенко
Факультет кибернетики

Киевский национальный университет имени Тараса Шевченко, Киев, Украина
taarraas@gmail.com, v_ter@ukr.net

Abstract

Paper presents Balaban's algorithm modification. Most of calculations have been moved from children's nodes to parent, which gave additional performance.

Ключевые слова: пересечение отрезков, дерево алгоритма, лестница отрезков, детерминированный алгоритм.

1. ВВЕДЕНИЕ

Постановка проблемы. В работе рассматривается один из подходов решения задачи детерминированного пересечения отрезков. Результаты решения этой задачи имеют большое теоретическое и практическое значение в информатике и ряде прикладных наук. В частности, необходимость решения задач на пересечение возникает в архитектурном проектировании, компьютерной графике (удаление невидимых линий и поверхностей, построение сложных 3D изображений), в задачах сегментации изображений и оптимального раскроя. В микроэлектронике возникает необходимость проверки пересечений разных компонентов интегральных схем, которые состоят из большого количества элементов. С теоретической точки зрения, разработка эффективных алгоритмов определения пересечений позволяет исследовать сложность и глубинную структуру геометрических задач, что в свою очередь, открывает путь к поиску оптимальных решений.

Анализ последних исследований. Методы поиска пересечений отрезков разделяются на детерминированные и недетерминированные. Тривиальный детерминированный алгоритм имеет временную сложность $O(n^2)$ и суть его заключается в проверке попарного пересечения отрезков. Сложнее, но эффективнее алгоритм Бенгли-Отмена [2] с оценкой сложности $O((n+k)\log n+k)$, в основе которого лежит метод заметающей прямой. Алгоритм, предложенный Чазеле и Едельсбруннером [3], имеет лучшую оценку $O(n \log n + k)$, но в отличие от предыдущих методов требует квадратичной памяти. Оптимальный детерминированный алгоритм был предложен Балабаном [1] с временной оценкой сложности $O(n \log(n+k))$ и $O(n)$ памяти. В работе [4] акцентировано внимание на отсутствие необходимости в использовании дополнительной памяти. В предлагаемой работе реализован более эффективный алгоритм, позволяющий повысить скорость работы алгоритма Балабана в 1.5 - 2.5 раза в зависимости от количества пересечений за счет выявления некоторых пересечений еще в родительских узлах графа алгоритма на основе метода "разделяй и властвуй".

Цель работы. Оптимизировать по времени алгоритм Балабана [1] поиска пересечения отрезков.

Постановка задачи (поиск пересечения отрезков). Пусть заданное множество S состоящее с N отрезков на

плоскости. Необходимо определить полное множество всех точек попарного пересечения отрезков S .

2. АЛГОРИТМ ПОСТРОЕНИЯ РЕШЕНИЯ

2.1 Основные понятия и определения

Введем некоторые обозначения. Пусть $Int(S)$ - множество всех точек пересечения отрезков S , а $|Int(S)|$ - количество пересечений K ; через $\langle b, e \rangle$ обозначим вертикальную полосу, которая ограничена прямыми $x = b$ и $x = e$, а через s отрезок с концами абсцисс l и r . Рассмотрим взаимное расположение вертикальной полосы $\langle b, e \rangle$ и отрезка s .

Определение. Будем говорить, что отрезок s , с концами абсцисс l и r :

- содержит (span) полосу $\langle b, e \rangle$, если $l \leq b \leq e \leq r$;
- внутренний для полосы $\langle b, e \rangle$, если $b < l < r < e$;
- пересекает (strip) полосу $\langle b, e \rangle$ в других случаях.

Введем отношение порядка на множестве отрезков $s_1 \prec_b s_2$, если оба отрезка пересекают вертикальную линию $x = b$ и точка пересечения этой прямой с отрезком s_1 лежит ниже точки пересечения с s_2 .

Определение. «Лестница» D — это пара $\langle Q, \langle b, e \rangle \rangle$, в которой отрезки Q удовлетворяют следующим условиям:

- любой отрезок из Q содержит полосу $\langle b, e \rangle$;
- нет пересечений отрезков внутри лестницы;
- Q упорядочена по отношению \prec_b .

Часть отрезков лестницы внутри полосы будем называть *ступеньками*.

Определение. Будем называть лестницу D *полностью соотносимой к множеству S* , если каждый отрезок из S не пересекает полосу $\langle b, e \rangle$, или же пересекает одну из ступенек D .

Определение. Отрезки s_1 и s_2 будем называть *пересекающимися в полосе $\langle b, e \rangle$* , если абсцисса точки их пересечения находится между b и e . Обозначим $Int(S, S')$ множество пересечений множества отрезков S из S' .

Предварительная обработка и структура данных. Пусть задано множество отрезков $S = \{s_1, s_2, \dots, s_n\}$. Упорядочим множество его конечных точек $P_i = (x_i, y_i)$ по возрастанию абсциссы, а при равенстве абсцисс - по ординате. Полученное таким образом множество обозначим через L . Пусть $s(i)$ - номер отрезка, которому принадлежит точка P_i .

2.2 Алгоритм

1. Введем вспомогательную функцию *Split*, которая разделяет входное множество отрезков L , пересекающих некоторую полосу $\langle b, e \rangle$, на подмножества Q и L' так, что лестница $\langle Q,$

$\langle b, e \rangle$ полностью соотносима к множеству отрезков L' , рис. 1.

```

procedure Split $_{b,e}(L, Q, L')$ ;
{Let  $L = (s_1, \dots, s_k)$ ,  $s_i <_b s_{i+1}$ }
 $L' := \emptyset$ ;  $Q := \emptyset$ ;
For  $j = 1, \dots, k$  do
  If the segment  $s_j$  doesn't intersect
  the last segment of  $Q$  within  $\langle b, e \rangle$  and
  spans this strip then
    add  $s_j$  to the end of  $Q$ 
  else
    add  $s_j$  to the end of  $L'$ ;
  endif;
endfor;
end procedure.
    
```

Рисунок 1: Разделение множества L на подмножества Q и L' .

2. Пусть множество отрезков L содержит полосу $\langle b, e \rangle$ и отсортировано по отношению \langle_b . Определим функцию, которая находит все пересечения множества отрезков L в полосе $\langle b, e \rangle$ и формирует множество R из этих же отрезков, но отсортированных по отношению \langle_c . Все отрезки в L содержат полосу $\langle b, e \rangle$. Разобьем множество L на L_1, L_2, \dots, L_n с помощью функции $Split$ следующим образом. Пусть $L_0 = L$, тогда функция $Split(L_0, Q_0, L_1)$ дает лестницу $\langle Q_0, \langle b, e \rangle \rangle$, которая полностью соотносима с множеством отрезков L_1 . Аналогично разбивается L_1 на Q_2 и L_2 . Процесс разбиения продолжается до тех пор, пока L_i не станет равным пустому множеству, рис. 2. На следующем шаге найдем все пересечения для каждого $i = 1, \dots, n$ между лестницей $\langle L_i, \langle b, e \rangle \rangle$ и множеством L'_i полностью соотносимых отрезков, которые ее пересекают. Так как оба множества являются упорядоченными по отношению \langle_b , то данную операцию можно провести за линейное время от количества пересечений для каждого из отрезков L_i .

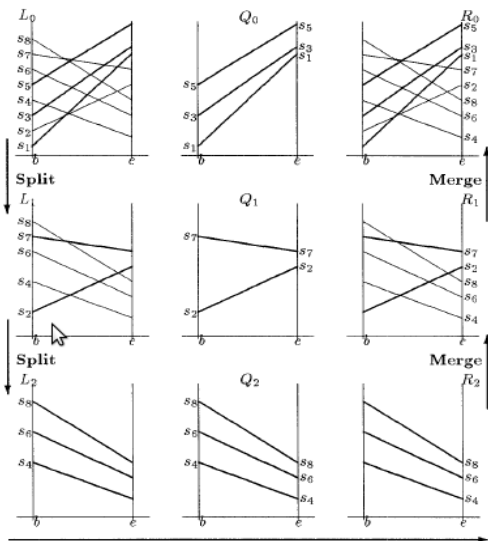


Рисунок 2: Пример работы алгоритма поиска всех пересечений отрезков в полосе.

Описанный алгоритм представлен на рис. 3. В нем используется процедура $Merge$, которая объединяет два множества S и S' , отсортированные по \langle_c .

```

procedure SearchInStrip $_{b,e}(L, R)$ 
  Split( $L, Q, L'$ );
  If  $L' = \emptyset$  then  $R := Q$ ; exit; endif;
  Find  $Int_{b,e}(Q, L')$ ;
  SearchInStrip $_{b,e}(L', R')$ ;
   $R := Merge_c(Q, R')$ ;
end procedure.
    
```

Рисунок 3: Алгоритм поиска пересечений для случая, когда все отрезки содержат некоторую полосу.

3. Общий алгоритм нахождения всех пересечений отрезков с помощью алгоритма Балабана, рис.4.

```

procedure IntersectingPairs( $S_0$ ).
  Sort the  $2N$  endpoints by abscissa and
  find  $p_i, s(i), i = 1, \dots, 2N$ ;  $S_r := S_0$ ;
  TreeSearch( $S_r, 1, 2N$ );
end procedure.

procedure TreeSearch( $S_v, b, e$ ).
1. If  $e - b = 1$  then
   $L_v := \text{sort } S_v \text{ by } \langle_b$ ; SearchInStrip $_{b,e}(L_v, R_v)$ ; exit;
  endif;
2. Split  $S_v$  into  $Q_v$  and  $S'_v$  so that staircase
   $D_v := (Q_v, \langle b, c \rangle)$  be complete relative to  $S'_v$ ;
3. Find  $Int(D_v, S'_v)$ ;
4.  $c := \lfloor (b + e) / 2 \rfloor$ ;
5. Place segments of  $S'_v$ 
   crossing the strip  $\langle b, c \rangle$  into  $S_{ls(v)}$  and
   the strip  $\langle c, e \rangle$  into  $S_{rs(v)}$ ;
6. TreeSearch( $S_{ls(v)}, b, c$ );
7. TreeSearch( $S_{rs(v)}, c, e$ );
end procedure.
    
```

Рисунок 4: Основная часть алгоритма Балабана.

На первом шаге алгоритма выполняется поиск отрезков для случая, когда все отрезки содержат полосу. Для оптимизации проверяется только случай, когда полоса проходит через две соседние вершины упорядоченного списка вершин всех отрезков. На шаге 2 выполняется процедура $Split$, которая разбивает множества всех отрезков полосы на лестницу и соотносимое множество. На шаге 3 находим пересечения между полученными множествами с помощью описанного ранее оптимального алгоритма. На шагах 4-5 множества всех отрезков разбивается на две части. Для этого разделяем полосу $\langle b, e \rangle$ на две части - $\langle b, c \rangle$ и $\langle c, e \rangle$, где c медиана упорядоченного списка вершин отрезков полосы $\langle b, e \rangle$. В полосу $\langle b, c \rangle$ войдут все внутренние и пересекающие ее отрезки. Последние шаги вызывают рекурсивно алгоритм для полос $\langle b, c \rangle$ и $\langle c, e \rangle$. Порядок обхода здесь имеет значение, так как после выполнения шага 6 получаем упорядоченное по \langle_c множество отрезков, необходимое для шага 7. Другими словами, сыновья дерева «разделяй и властвуй» связаны между собой. Эта особенность алгоритма не позволяет его распараллеливание, выполняя вычисления в сыновьях на разных процессорах, что является существенным недостатком алгоритма.

Из иллюстрации на рис.2 видно, что процедура *Split* возвращает в переменную Q_v не все отрезки, содержащие в себе полосу $\langle b, e \rangle$, и соответственно находит не все возможные пересечения между прямыми, которые содержат полосу, с пересекающими их прямыми. Эти отрезки обрабатываются в узлах сыновей, поскольку, если отрезок содержит целую полосу, то он содержит и ее части. Поэтому, для улучшения эффективности работы алгоритма предлагается ввести шаг, который при необходимости проведет дополнительное разбиение, вызвав рекурсивно функцию *TreeSearch* для этой же полосы:

```

3'.      If  $|S'v| < |Int(D_v, S'v)|$  then
          TreeSearch( $S', b, e$ ); exit;
        endif
    
```

3. ОЦЕНКА СЛОЖНОСТИ

Пусть N - это количество отрезков, K - количество пересечений. Процедура просматривает каждый из N отрезков. При этом расходуется $O(1)$ времени на просмотр отрезка, так как вставка в конец массива выполняется за константное время. Откуда и получаем необходимую оценку.

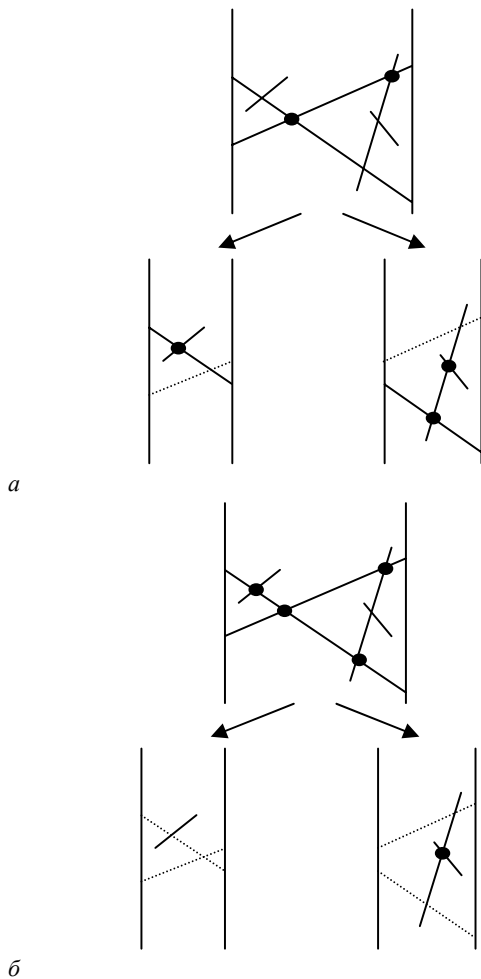


Рисунок 5: Порядок нахождения пересечений: а- до оптимизации, б- после оптимизации.

Время запуска этой процедуры равно суммарному времени ее запусков. Подсчитаем время i -того запуска. Пусть L_i, Q_i, L_i' — соответствующие множества, причем $L_0=L, L_{i+1}=L_i'$. Тогда *Split* и *Merge* выполняются за $O(|L_i|)$. Для каждого из отрезков время нахождения его пересечений пропорционально количеству пересечений, тогда время нахождения всех пересечений всех отрезков пропорционально количеству всех пересечений $|Int_{(b,e)}(Q_i, L_i')|$. В результате имеем общую сложность $O(|L_i| + |Int_{(b,e)}(Q_i, L_i')|)$

Процедура *IntersectingPairs* работает за время $O(N \log N) + K$ с использованием $O(N)$ памяти.

Полное обоснование асимптотической сложности алгоритма Балабана приведено в [1].

Предложенный новый шаг 3' позволяет найти пересечения между отрезками, которые содержат полосу $\langle b, e \rangle$ для текущей вершины дерева алгоритма "разделяй и властвуй", но которые не были отнесены к множеству Q_v , с другими отрезками в родительском узле для полосы $\langle b, e \rangle$. До этого шага 3' все такие отрезки разбивались на 2 части, и обрабатывались в узлах сыновей. В самом худшем случае такие отрезки будут обработаны в листьях, так как процедура *SearchInStrip* работает до тех пор, пока множество отрезков, которое передается ей на вход и содержащее полосу, не станет пустым. Однако, очевидно, что выполнять одну и ту же работу многократно не эффективно. Лучше, если это можно сделать один раз. С другой стороны, возможен случай, когда количество найденных новых отрезков, содержащих полосу, будет небольшим. Тогда время их поиска будет больше времени, сэкономленного вышеприведенной оптимизацией. Поэтому было принято решение выполнять разделение множества до тех пор, пока количество найденных пересечений превышает количество отрезков, которые пересекают полосу.

На рис. 5 приведен пример сравнения работы классического алгоритма Балабана и модифицированного алгоритма. Пунктиром обозначены не обрабатываемые на данном шаге отрезки, а жирными точками — найденные на шаге пересечения.

В первом случае на первом шаге была удалена только одна прямая и найдено 2 точки пересечения, а в узлах сыновей обработано 7 отрезков. Тогда как во втором случае сразу было найдено 4 вершины. В левом поддереве алгоритма «разделяй и властвуй» остался только один отрезок, для которого не требуется дополнительных вычислений, а в правом – 2 отрезка, которые дают еще одно пересечение. Таким образом, обработано 3 отрезка в узлах сыновей с использованием оптимизации, что в 2 раза меньше, чем без нее.

4. РЕАЛИЗАЦИЯ

Реализация модифицированного алгоритма проводилась на языке программирования Java с использованием библиотеки *Swing* для создания интерфейса программы. Для проверки работы программы использовалось юнит-тестирование отдельных модулей, и непосредственно сверка результатов алгоритма Балабана с тривиальным алгоритмом, проверяющим попарно все пересечения за $O(N^2)$, на больших наборах случайных данных. Результат работы алгоритма представлено на рис. 6.

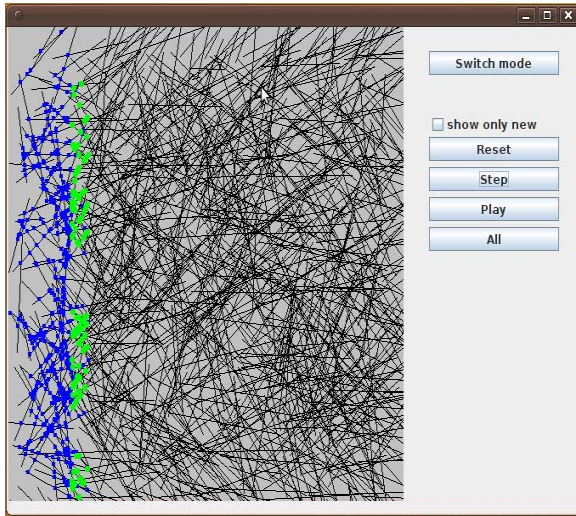


Рисунок 6: Пример работы программы алгоритма

5. ЗАКЛЮЧЕНИЕ

В работе проведен сравнительный анализ существующих алгоритмов поиска пересечения отрезков [6] и предложена эффективная модификация лучшего из них – алгоритма Балабана [1]. Из таблицы 1 видно, что уже при количестве отрезков равным 2000 и большому количеству пересечений целесообразно использовать алгоритм Балабана. Однако в результате громоздкости и высокой сложности реализации алгоритма в большинстве практических задач используется алгоритм заметающей прямой Бентли-Отмена [2].

n	V	S	M	T	B	BM
2000	4007	1,14	1,74	15,13	1,94	1,05
2000	4026	1,18	2,25	14,87	2,07	1,08
2000	4136	1,25	2,91	15,26	2,17	1,09
2000	4428	1,39	3,44	15,06	2,33	1,18
2000	5857	1,81	4,44	15,31	2,48	1,27
2000	10954	3,03	5,93	15,41	2,74	1,4
2000	29683	7,57	9,71	16,01	3,22	1,63
2000	91789	22,84	20,04	16,62	5,38	2,59
2000	267048	70,24	48,96	18,42	11,54	5,76

Таблица 1: Время выполнения алгоритмов поиска пересечений отрезков в секундах для одной и той же тестовой машины.

Здесь n - количество сегментов, V - количество граней на которые отрезки разбивают плоскость, S - алгоритм

Бентли-Отмена, M - недетерминированный алгоритм Mulmuley [8], B - алгоритм Балабана, T - тривиальный алгоритм, BM - улучшенный алгоритм Балабана.

На рис. 7 показано сравнение эффективности работы модифицированного алгоритма и классического. Дальнейшее усовершенствование алгоритма должно заключаться в возможности запуска его на многопроцессорных системах.

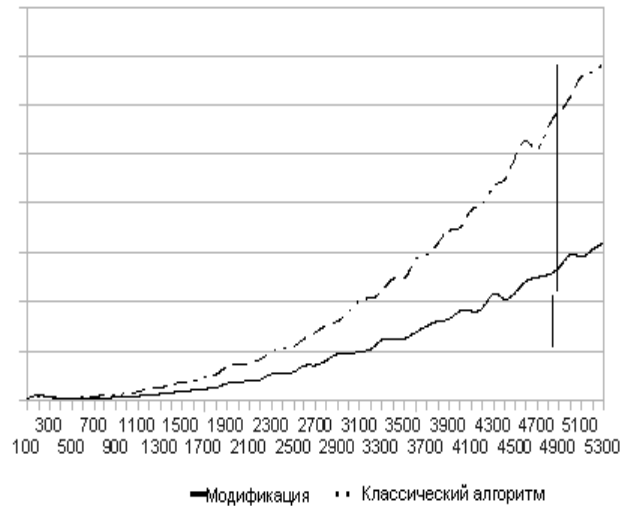


Рисунок 7: Зависимость времени выполнения модифицированного и классического алгоритмов от количества отрезков.

6. ЛИТЕРАТУРА

- [1] I.J. Balaban. An optimal algorithm for finding segments intersections. *In Proceedings of the Eleventh Annual Symposium on Computational Geometry*, ACM Press, New York, 1995. - pp. 211–219.
- [2] B. Chazelle. Intersecting is easier than sorting. *In Proceedings of the 16th Annual ACM Symposium on Theory of Computing*, 1984. - pp. 125–134.
- [3] B. Chazelle, H. Edelsbrunner. An optimal algorithm for intersecting line segments in the plane. *Journal of the ACM*, 39 (1), 1992. - pp. 1–54.
- [4] Jan Vahrenhold. Line-segment intersection made in-place. *Computational Geometry*, 38, 2007. – pp. 213–230.
- [5] Goodman J.E., O'Rourke J. Handbook of discrete and computational geometry (2ed., CRC, 2004). - pp. 551-554.
- [6] <http://www.algorithmic-solutions.com/leda/>
- [7] Kurt Mehlhorn and Stefan Näher. LEDA—a platform for combinatorial and geometric computing. *Cambridge University Press*, Cambridge, UK, 2009.- pp.733-735.
- [8] K. Mulmuley. A fast planar partition algorithm, I. *Journal of Symbolic Computation*, 10(3/4), 1990. -pp.:253-280.

About the author

Vasyl Tereshchenko is an associate professor at National Taras Shevchenko University of Kyiv, Faculty of Cybernetics. His contact email is v_ter@ukr.net.

Taras Voznyuk is a student at National Taras Shevchenko University of Kyiv, Faculty of Cybernetics. His contact email is taarraas@gmail.com

Скелет многоугольной фигуры – представление плоским прямолинейным графом

Леонид Местецкий

Кафедра математических методов прогнозирования,

Московский Государственный университет имени М.В. Ломоносова, Ленинские горы, Москва, Россия

l.mest@ru.net

Аннотация:

Предлагается новый метод представления скелета многоугольной фигуры в виде плоского графа, рёбрами которого являются кривые Безье первого и второго порядка. Приводится описание радиальной функции скелета сплайнами Безье. Такое представление позволяет описать криволинейный в общем случае скелет многоугольной фигуры с помощью так называемого прямолинейного контрольного графа, составленного из характеристических многоугольников кривых Безье. Скелет вместе с радиальной функцией предоставляет альтернативный способ описания фигуры в виде объединения семейства кругов.

Ключевые слова: многоугольная фигура, скелет, диаграмма Вороного, радиальная функция, параболические рёбра, кривые Безье, контрольный граф.

1. ВВЕДЕНИЕ

Многоугольная фигура – это замкнутая область на плоскости, граница которой состоит из конечного числа простых непересекающихся многоугольников. Скелет многоугольной фигуры – это серединные оси фигуры, множество точек, имеющих не менее двух ближайших граничных точек фигуры. Многоугольные фигуры и их скелеты широко используются в компьютерной графике, анализе и распознавании формы изображений (Siddiqi, Pizer, 2008, Местецкий, 2009).

Методы построения скелета многоугольной фигуры обычно используют концепцию диаграммы Вороного линейных сегментов. Для множества отрезков, составляющих границу фигуры, строится диаграмма Вороного, из которой далее выделяется скелет. Скелет многоугольной фигуры, имеющей n вершин, может быть получен за время $O(n)$ из диаграммы Вороного. В свою очередь, известны эффективные $O(n \log n)$ алгоритмы построения диаграммы Вороного для множества линейных сегментов общего вида (Fortune, 1987, Yip, 1987), а также для сегментов, образованных сторонами простого многоугольника (Lee, 1982) или многосвязной многоугольной фигуры (Mestetskiy, 2009).

Геометрическая конструкция скелета многоугольной фигуры выглядит достаточно просто: она имеет вид плоского графа, рёбра которого состоят из отрезков прямых и парабол (рис.1).

Из-за присутствия в скелете параболических рёбер возникают некоторые технические неудобства при построении скелета, а также при его хранении и использовании для анализа и распознавания изображений.

Неявные уравнения парабол включают квадратичные формы общего вида, что является неудобным для рисования и

анализа. При хранении данных о скелете приходится запоминать фокусы и директриссы параболических рёбер для вычисления уравнений в ходе обработки. Кроме того, неявное уравнение параболы задаёт всю бесконечную кривую, а в скелет входит лишь её конечный фрагмент. Выделение этого фрагмента также создаёт определённые трудности.

Эти неудобства порождают стремление создать альтернативный скелет без параболических рёбер. Такой подход реализован в концепции «прямолинейного скелета»

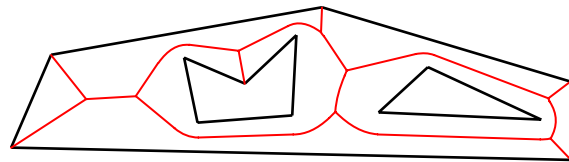


Рис.1. Многоугольная фигура и её скелет.

(Aichholzer, Aurenhammer, 1996). Но прямолинейный скелет имеет свои недостатки: сложность математического определения, низкая эффективность алгоритмов построения, сложность регуляризации при наличии шумовых эффектов. Эти недостатки явно превосходят по своим масштабам проблему вычисления парабол в обычном скелете.

В настоящей статье предлагается другой способ описания скелета в виде прямолинейного графа, не требующий получения уравнений параболических рёбер ни на этапе построения диаграммы Вороного, ни при хранении, рисовании и обработке. Достигается это следующим образом.

1. Скелет многоугольной фигуры представляется объединением некоторого множества элементарных кривых Безье первой и второй степени. Это объединение мы называем составной кривой Безье.
2. Составная кривая Безье определяется своим контрольным графом, который образуется из характеристических многоугольников элементарных кривых Безье. Контрольный граф имеет прямолинейные рёбра.

Таким образом, описание скелета, которое используется для его хранения и обработки, представляет собой

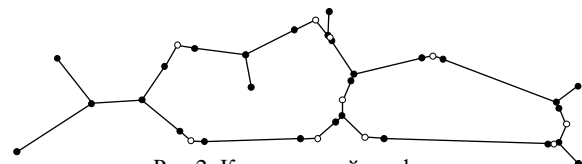


Рис.2. Контрольный граф скелета.

прямолинейный граф. На рис.2 изображён контрольный граф скелета фигуры из рис.1. Множество вершин контрольного графа состоит из двух подмножеств. Первое подмножество (чёрные точки) – это вершины скелета многоугольной

фигуры. А второе подмножество (белые точки) – это управляющие вершины кривых Безье второго порядка.

Для того, чтобы построить такое представление скелета, мы покажем, что параболические сегменты скелета описываются кривыми Безье второй степени и опишем алгоритм получения этого представления. Далее мы покажем, что радиальная функция скелета вычисляется также с помощью сплайнов Безье.

Работа выполнена при поддержке Российского фонда фундаментальных исследований (грант 08-01-00670).

2. СТРУКТУРА СКЕЛЕТА

Пусть задана многоугольная фигура M на плоскости R^2 с евклидовым расстоянием $d(p, q)$, $p, q \in R^2$. Граница фигуры ∂M состоит из нескольких простых многоугольников.

Пустым кругом фигуры M с центром в точке p и радиусом $r \geq 0$ называется замкнутое множество точек $K_r(p) = \{q : q \in R^2, d(p, q) \leq r\}$ такое, что $K_r(p) \subset M$.

Максимальным пустым кругом (или вписанным кругом) $K_r^{\max}(p)$ фигуры M называется пустой круг, который не содержится ни в каком другом пустом круге. Очевидно, что не всякая точка фигуры M является центром максимального пустого круга.

Скелетом S фигуры M называется множество всех точек центров её максимальных пустых кругов

$$S = \{p : p \in M, K_r^{\max}(p) \neq \emptyset\}.$$

Это определение скелета является более точным, чем то, что приведено выше во введении (через понятие серединных осей), поскольку оно включает в скелет все выпуклые вершины фигуры. Согласно определению, данному выше, каждую точку фигуры можно рассматривать как пустой круг нулевого радиуса. При этом вырожденные круги с центрами в выпуклых вершинах фигуры являются максимальными пустыми кругами, поскольку не содержатся в других пустых кругах. Следовательно, в скелет многоугольной фигуры входят точки, которые совпадают с выпуклыми вершинами многоугольной фигуры.

В каждой точке скелета определена радиальная функция, равная радиусу вписанного круга с центром в этой точке. Радиальная функция задаёт «ширину» фигуры относительно точек скелета.

Пусть S – скелет многоугольной фигуры M . Общее число точек, составляющих множество S , бесконечно. Тем не менее, оказывается, что все эти точки лежат на конечном множестве отрезков прямых линий и квадратичных парабол. Пусть $s \in S$ – точка скелета, а g_1 и g_2 – две ближайшие к ней граничные точки фигуры, $g_1, g_2 \in \partial M$. Точки g_1 и g_2 могут занимать различное положение на границе фигуры. Будем называть граничную точку угловой, если она является вершиной многоугольной фигуры, и простой в

остальных случаях. Возможны три варианта сочетания типов точек g_1 и g_2 : пара угловых, пара простых или пара из угловой и простой точек.

В случае, когда обе точки g_1 и g_2 угловые, точка $s \in S$, равноудалённая от g_1 и g_2 , обязательно должна лежать на прямой, являющейся серединным перпендикуляром для отрезка $[g_1, g_2]$ (рис.3a).

Если обе точки g_1 и g_2 простые, то точка s равноудалена от сторон многоугольников, содержащих g_1 и g_2 . Значит, точка s лежит на биссектрисе угла, образованного этими

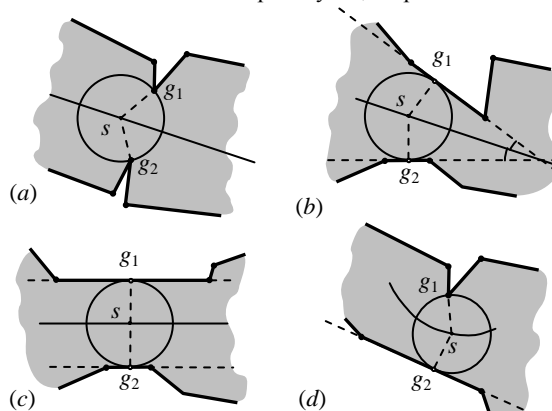


Рис. 3: Бисекторы многоугольной фигуры.

сторонами (рис.3b). Если же стороны параллельны, то s лежит на прямой линии, равноудалённой от этих сторон (рис.3c).

Наконец в случае, когда пара g_1, g_2 состоит из угловой и простой точек, точка s равноудалена от угловой точки и от стороны многоугольника, содержащей простую точку. Это значит, что s лежит на параболе, для которой фокусом является угловая точка, а директрисой – сторона многоугольника, на которой лежит точка g_2 (рис.3d).

Таким образом, все точки скелета лежат на указанных линиях трёх перечисленных типов, определяемых парами «вершина-вершина», «сторона-сторона» и «вершина-сторона» фигуры.

Для общности будем называть вершины и стороны многоугольной фигуры сайтами. Будем также называть бисектором максимальное связанное подмножество точек скелета, равноудалённых от одной и той же пары сайтов. В зависимости от пары определяющих сайтов бисекторы будем называть: vv -бисектор для пары «вершина-вершина», ss -бисектор для пары «сторона-сторона» и vs -бисектор для пары «вершина-сторона».

Скелет многоугольной фигуры можно рассматривать как плоский граф, рёбрами которого являются бисекторы, а вершинами – концевые точки бисекторов. Вершины графа – это точки соединения бисекторов между собой и выпуклые вершины фигуры.

3. ВЕРШИНЫ СКЕЛЕТА

Вершинами скелета являются центры максимальных пустых кругов, касающихся трёх или более сайтов. Поэтому задача поиска вершин может быть решена путем построения

касательных окружностей для троек сайтов. Вычисление таких кругов приводит к следующим геометрическим задачам (рис. 4).

Возможные комбинации сайтов:

- 1) три сайта-точки (рис.4a);
- 2) два сайта-точки и один сайт-сегмент (рис.4b,c);

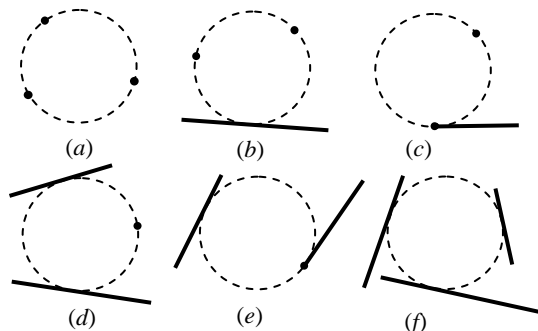


Рис.4. Касательные окружности для различных троек сайтов.

- 3) два сайта-сегмента и один сайт-точка. (рис.4d,e);
- 4) три сайта-сегмента (рис.4f).

Во второй и третьей комбинациях возможны два варианта, в зависимости от того, совпадают ли точки с концами сегментов.

Если последовательность точек касания задана и касательная окружность существует, то она единственна.

Непосредственное вычисление центра касательной окружности t трёх сайтов s_1, s_2, s_3 осуществляется на основе решения системы из двух уравнений, составленной из следующих условий:

$$\begin{cases} d^2(t, s_1) = d^2(t, s_2) \\ d^2(t, s_1) = d^2(t, s_3) \end{cases}$$

В ситуациях, изображённых на рис.4a,c,e,f оба уравнения являются линейными. А в случаях на рис.4b,d одно уравнение линейное, а другое имеет вторую степень. Выразив одну координату точки t через другую из линейного уравнения, можно привести второе уравнение к обычному квадратному уравнению. Полученное решение должно удовлетворять двум дополнительным условиям, которые легко проверяются. Первое условие состоит в том, что проекции найденной точки t на сайты-сегменты должны лежать на самих сегментах. Второе условие выражается в том, что касательная окружность должна лежать внутри фигуры и, следовательно, центр её t лежит слева от сайта-сегмента.

Таким образом, вычисление вершин скелета для многоугольной фигуры сводится к комбинаторному перебору различных троек сайтов и вычислению для них касательных окружностей. Для решения этой задачи известны как простые наивные алгоритмы, имеющие высокую вычислительную сложность, так и более сложные алгоритмы, осуществляющие направленный перебор троек весьма эффективно, обеспечивая вычислительную сложность $O(n \log n)$.

4. РЁБРА СКЕЛЕТА

Как уже отмечалось выше, непосредственное вычисление параболических рёбер в виде неявного уравнения второй степени не очень удобно для рисования её средствами компьютерной графики и для анализа скелета при решении задач преобразования и распознавания изображений.

Для работы с параболическими рёбрами скелета гораздо более удобным представляется их описание в виде параметрических кривых вида $V(t) = (x(t), y(t))$, $t \in [0,1]$, где точки $V(0)$ и $V(1)$ – это вершины скелета, являющиеся концами ребра. Такое параметрическое описание предлагается получить с помощью кривых Безье.

Кривая Безье степени m – это параметрическая линия вида

$$V(t) = \sum_{j=0}^m V_j B_j^m(t), \text{ с параметром } t \in [0,1], \text{ где}$$

$V = \{V_0, V_1, \dots, V_m\}$ упорядоченное множество опорных точек, называемое *характеристическим многоугольником*

кривой Безье, а $B_j^m(t) = \frac{m!}{(m-j)!j!} (1-t)^{m-j} t^j$,

$j = 0, \dots, m$ – полиномы Бернштейна. Точки V_0, V_m называются концевыми, а точки V_1, \dots, V_{m-1} управляющими.

Идея предлагаемого нами решения состоит в представлении линейных рёбер скелета кривыми Безье первого порядка $V(t) = V_0 B_0^1(t) + V_1 B_1^1(t)$, $t \in [0,1]$, а параболических рёбер – кривыми Безье второго порядка $V(t) = V_0 B_0^2(t) + V_1 B_1^2(t) + V_2 B_2^2(t)$, $t \in [0,1]$.

Характеристический многоугольник кривой Безье первого порядка имеет две вершины V_0 и V_1 , являющиеся вершинами скелета. Характеристический многоугольник кривой второго порядка имеет три вершины $V = \{V_0, V_1, V_2\}$. При этом концевые точки ребра V_0 и V_2

– это вершины скелета, а точка V_1 – управляющая (или виртуальная) вершина кривой Безье. Такой способ описания рёбер является компактным, так как дополнительно к концевым точкам всех рёбер, являющихся вершинами скелета, нужно для каждого параболического ребра запоминать лишь одну управляющую точку. Кроме того, существенно упрощается рисование и обработка рёбер, поскольку для кривых Безье имеются эффективные алгоритмы решения этих задач.

Для реализации предлагаемой идеи необходимо уметь вычислять управляющие точки параболических рёбер.

Параболическое ребро скелета представляет собой *vs*-бисектор. Пусть A и B – пара сайтов, задающих этот бисектор, причём A – это вершина, а B – сторона многоугольной фигуры. Сторона B – это отрезок, соединяющий вершины фигуры B_1 и B_2 . Мы будем обозначать B_1B_2 сам отрезок и прямую, на которой он лежит. Не нарушая общности, предположим, что многоугольная фигура находится слева от стороны B_1B_2 . Опустим перпендикуляр AD из точки A на прямую B_1B_2 и выберем точку O как середину AD . С парой сайтов A и B свяжем прямоугольную декартову систему координат с центром O , осью абсцисс параллельной B_1B_2 и осью ординат DA и (рис.5).

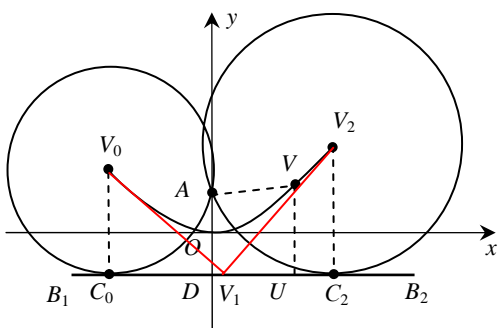


Рис.5. Параболическая кривая *vs*-бисектора.

Бисектор, определяемый парой сайтов A и B , является линией центров кругов, проходящих через точку A , для которых прямая B_1B_2 является касательной (рис.5). Обозначим концевые точки этого бисектора V_0 и V_2 . Пусть C_0 и C_2 проекции V_0 и V_2 на прямую B_1B_2 .

Пусть V точка на бисекторе и U её ортогональная проекция на $[B_1, B_2]$. Тогда длины отрезков AV и UV равны. Если координаты точки $V = (x, y)$, а координаты точки $A = (0, p)$, то из условия $AV^2 = UV^2$ получаем $x^2 + (y - p)^2 = (y + p)^2$. Отсюда получаем уравнение параболы для бисектора $y = \frac{1}{4p}x^2$.

Рассмотрим касательные параболы в точках $V_0 = (x_0, y_0)$ и $V_2 = (x_2, y_2)$.

Как известно, уравнение касательной к кривой $F(x, y) = 0$, в точке (\hat{x}, \hat{y}) имеет вид

$$F'_x(\hat{x}, \hat{y}) \cdot (x - \hat{x}) + F'_y(\hat{x}, \hat{y}) \cdot (y - \hat{y}) = 0.$$

В нашем случае $F(x, y) = x^2 - 4py$, поэтому уравнения касательных к кривой $x^2 - 4py = 0$ в точках V_0 и V_2 есть

$$2x_0 \cdot (x - x_0) - 4p \cdot (y - y_0) = 0, \quad (1)$$

$$2x_2 \cdot (x - x_2) - 4p \cdot (y - y_2) = 0. \quad (2)$$

Следовательно, точка пересечения этих касательных может быть найдена из системы уравнений:

$$\begin{cases} x_0x - 2py = x_0^2 - 2py_0 \\ x_2x - 2py = x_2^2 - 2py_2 \end{cases}$$

С учётом того, что

$$y_0 = \frac{1}{4p}x_0^2, \quad y_2 = \frac{1}{4p}x_2^2, \quad (3)$$

получаем решение этой системы

$$x_1 = \frac{1}{2}(x_0 + x_2), \quad (4)$$

$$y_1 = \frac{1}{4p}x_0x_2. \quad (5)$$

Таким образом, мы получили точку пересечения касательных $V_1 = (x_1, y_1)$ (рис.5).

Рассмотрим кривую Безье второго порядка

$V(t) = V_0B_0^2(t) + V_1B_1^2(t) + V_2B_2^2(t)$, $t \in [0, 1]$, заданную характеристическим треугольником с вершинами $\{V_0, V_1, V_2\}$. Полиномы Бернштейна $B_j^m(t)$ при $m = 2$

есть $B_0^2(t) = (1 - t)^2$, $B_1^2(t) = 2t(1 - t)$, $B_2^2(t) = t^2$.

Тогда параметрические уравнения для кривой Безье $V(t)$ имеют следующий вид:

$$x(t) = (x_0 - 2x_1 + x_2)t^2 - 2(x_0 - x_1)t + x_0, \quad (6)$$

$$y(t) = (y_0 - 2y_1 + y_2)t^2 - 2(y_0 - y_1)t + y_0, \quad (7)$$

$$t \in [0, 1].$$

Из (4) и (6) следует

$$x(t) = x_0 + (x_2 - x_0) \cdot t. \quad (8)$$

А из (7), (3) и (5) получаем

$$\begin{aligned} y(t) &= \frac{1}{4p}[(x_0^2 - 2x_0x_2 + x_2^2)t^2 - 2(x_0^2 - x_0x_2)t + x_0^2] = \\ &= \frac{1}{4p}[(x_0 - x_2)^2 \cdot t^2 - 2x_0(x_0 - x_2)t + x_0^2] = \\ &= \frac{1}{4p}[(x_0 - x_2)t - x_0]^2. \end{aligned} \quad (9)$$

Из (8) и (9) следует, что координаты точки кривой Безье связаны уравнением $y(t) = \frac{1}{4p} [x(t)]^2$. А это совпадает с уравнением параболы, задающим бисектор.

Таким образом, получена искомая простая форма описания бисектора в виде кривой Безье. Это кривая второго порядка, которая задаётся характеристическим треугольником. Две его вершины – это концевые точки бисектора, а третья вершина – это точка пересечения касательных к бисектору в концевых точках.

Следовательно, для того, чтобы задать бисектор в виде кривой Безье, необходимо вычислить касательные в концевых точках бисектора и найти их пересечение. Рассмотрим решение этой задачи.

5. ХАРАКТЕРИСТИЧЕСКИЙ ТРЕУГОЛЬНИК ПАРАБОЛИЧЕСКОГО РЕБРА

Сначала покажем, что касательная к параболе в любой её точке ортогональна вектору, направленному из фокуса параболы в проекцию этой точки на директрису (рис.6).

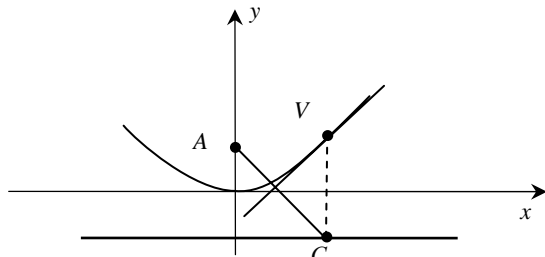


Рис.6. Ортогональность касательной и направления из фокуса в точку проекции.

Уравнение касательной к параболе $x^2 - 4py = 0$ в точке $V = (\hat{x}, \hat{y})$ есть:

$$2\hat{x} \cdot (x - \hat{x}) - 4p \cdot (y - \hat{y}) = 0.$$

А вектор из фокуса параболы $A = (0, p)$ в точку $C = (\hat{x}, -p)$, являющуюся проекцией точки $V = (\hat{x}, \hat{y})$ на директрису параболы $y = -p$, есть $\overrightarrow{AC} = (\hat{x}, -2p)$. Нетрудно заметить, что вектор \overrightarrow{AC} и направляющий вектор касательной $(2\hat{x}, -4p)$ являются коллинеарными.

Это свойство позволяет вычислить касательные в концевых точках V_0 и V_2 параболического ребра скелета. Для этого нужно сначала найти проекции C_0 и C_2 точек V_0 и V_2 на прямую B_1B_2 . А затем вычислить нормали соответствующих касательных – вектора $\overrightarrow{AC_0}$ и $\overrightarrow{AC_2}$.

Исходной информацией для определения касательных к бисектору в его концевых точках служат следующие данные. Известны концевые точки бисектора $V_0 = (x_0, y_0)$ и $V_2 = (x_2, y_2)$, а также пара сайтов A и B , причём один

сайт – это вершина, а другой – сторона многоугольной фигуры. Для определённости будем считать, что A это вершина, а B сторона, имеющая вершины B_1 и B_2 , причём многоугольник лежит слева от этой стороны. Задача состоит в том, чтобы найти виртуальную вершину V_1 характеристического треугольника $\{V_0, V_1, V_2\}$.

В описании алгоритма приняты следующие обозначения:

\overrightarrow{PQ} - вектор с началом P и концом Q ;

$[\overrightarrow{P_1Q_1} \times \overrightarrow{P_2Q_2}]$ - векторное произведение;

$(\overrightarrow{P_1Q_1}, \overrightarrow{P_2Q_2})$ - скалярное произведение;

$V + \overrightarrow{PQ}$ - сдвиг точки V на вектор \overrightarrow{PQ} ;

$|\overrightarrow{PQ}|$ - модуль вектора.

Алгоритм решения этой задачи следующий.

1. Вычислить параметр p параболы:

$$p = \frac{[\overrightarrow{B_1B_2} \times \overrightarrow{B_1A}]}{2 \cdot |\overrightarrow{B_1B_2}|}.$$

2. Вычислить точки C_0 и C_2 - проекции точек V_0 и V_2 на B :

$$C_0 = B_1 + \overrightarrow{B_1B_2} \cdot \frac{(\overrightarrow{B_1B_2}, \overrightarrow{B_1V_0})}{|\overrightarrow{B_1B_2}|},$$

$$C_2 = B_1 + \overrightarrow{B_1B_2} \cdot \frac{(\overrightarrow{B_1B_2}, \overrightarrow{B_1V_2})}{|\overrightarrow{B_1B_2}|}.$$

3. Вычислить вектора $\overrightarrow{AC_0}$ и $\overrightarrow{AC_2}$:

$$\overrightarrow{AC_0} = (a, b), \quad \overrightarrow{AC_2} = (c, d).$$

(a, b) - координаты вектора $\overrightarrow{AC_0}$,

(c, d) - координаты вектора $\overrightarrow{AC_2}$.

4. Решить систему уравнений

$$\begin{cases} a \cdot (x - x_0) + b \cdot (y - y_0) = 0 \\ c \cdot (x - x_2) + d \cdot (y - y_2) = 0 \end{cases}$$

5. Решение системы даёт координаты виртуальной точки кривой Безье $V_1 = (x_1, y_1)$, описывающей ребро скелета.

Таким образом, характеристический граф кривой Безье, представляющей параболическое ребро скелета, описывается тройкой $\{V_0, V_1, V_2\}$, условно называемой характеристическим треугольником.

6. СКЕЛЕТНЫЙ ГРАФ ИЗ КРИВЫХ БЕЗЬЕ

Мы показали, что весь скелет многоугольной фигуры есть объединение кривых Безье первого и второго порядка. Это объединение будем называть составной кривой Безье. Такой термин используется в шрифтовом дизайне. Там составные кривые Безье описывают замкнутые контуры символов шрифта. В нашем случае кривые описывают более сложную структуру – связный плоский граф.

Характеристические многоугольники кривых Безье имеют прямолинейные рёбра. Объединение характеристических многоугольников всех рёбер скелета будем называть *контрольным графом* полученной составной кривой, или контрольным графом скелета многоугольной фигуры. Пример контрольного графа представлен на рис.2.

Важным свойством контрольного графа скелета является его планарность, которая доказывается следующим образом. Рассмотрим образующие сайты параболического ребра: сайт-точку A и сайт-сегмент B .

Если V_0 и V_2 лежат по одну сторону от оси Y , т.е. x_0 и

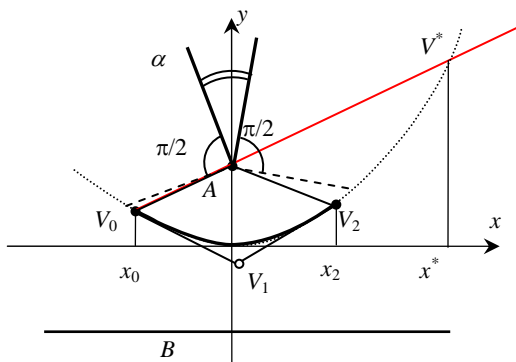


Рис 7: Планарность характеристического графа

x_2 имеют одинаковый знак, то из (5) следует, что $y_1 \geq 0$ и тогда точка V_1 лежит выше сегмента B .

Предположим теперь, что x_0 и x_2 имеют разные знаки (рис.7). Поскольку фокус параболы A – это вогнутая вершина многоугольной фигуры, то угол α , образованный инцидентными этой вершине сторонами, лежит в диапазоне $\pi < \alpha < 2\pi$. Рассмотрим угол $\angle V_0AV_2$ между векторами $\overrightarrow{AV_0}$ и $\overrightarrow{AV_2}$. Очевидно, что

$$\angle V_0AV_2 \leq 2\pi - \left(\alpha + 2 \cdot \frac{\pi}{2} \right) = \pi - \alpha < \pi.$$

Следовательно, существует прямая, проходящая через фокус A , такая, что точки V_0 и V_2 лежат ниже неё. Она пересекает параболу в точках $P = (x_p, y_p)$ и $Q = (x_q, y_q)$, причём

$$x_p < x_0, \quad x_q > x_2. \quad (10)$$

Уравнение прямой PQ имеет вид $y = p + a \cdot x$, где a – угловой коэффициент. Точки пересечения прямой с параболой определяются из уравнения $p + ax = \frac{1}{4p}x^2$. Это квадратное уравнение имеет два корня:

$$x_p = 2p \cdot (a - \sqrt{a^2 + 1}), \quad x_q = 2p \cdot (a + \sqrt{a^2 + 1}).$$

Точка пересечения касательных V_1 имеет ординату y_1 . Из уравнения (5) и условий (10) получаем следующую оценку

$$\begin{aligned} y_1 &= \frac{1}{4p}x_0x_2 < \frac{1}{4p}x_px_q^* = \frac{1}{4p}x_0x_2 = \\ &= \frac{1}{4p} \left[2p \cdot (a - \sqrt{a^2 + 1}) \right] \cdot \left[2p \cdot (a + \sqrt{a^2 + 1}) \right] = \\ &= \frac{1}{4p} 4p^2 \cdot [a^2 - (a^2 + 1)] = -p. \end{aligned}$$

Отсюда следует, что точка V_1 лежит слева от сайта-сегмента B , с той же стороны, что и сайт A и точки V_0 и V_2 , т.е. $\overrightarrow{AV_0}$ и $\overrightarrow{AV_2}$ не пересекают B .

Таким образом, получаем, что характеристический треугольник параболического ребра не имеет точек пересечения со своим определяющим сайтом-сегментом и лежит внутри объединения пустых кругов с центрами на параболическом сегменте. Следовательно, стороны

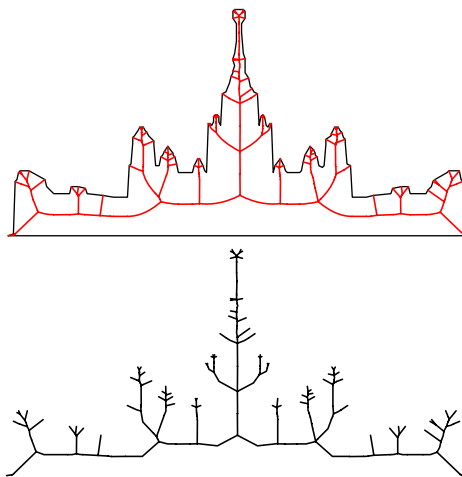


Рис.8: Многоугольная фигура и её скелет.

характеристического треугольника не имеют пересечений и с

остальными рёбрами контрольного графа. А это значит, что контрольный граф скелета является планарным.

Пример контрольного графа фигуры приведен на рис. 8. Фигура представляет собой 75-угольник, её скелет имеет 147 рёбер, а контрольный граф имеет 190 вершин, в том числе 43 виртуальных.

7. РАДИАЛЬНАЯ ФУНКЦИЯ СКЕЛЕТА

Радиальная функция скелета фигуры ставит в соответствие каждой точке скелета радиус вписанного пустого круга с центром в этой точке. Рассмотрим, каким образом можно представить радиальную функцию в случае, когда скелетный граф описан в виде составной кривой Безье.

С концевыми точками V_0 и V_1 линейного ss -бисектора связаны максимальные пустые круги, радиусы которых r_0 и r_1 известны. Очевидно, что радиус $r(t)$ пустого круга с центром в точке $V(t) = V_0 \cdot (1-t) + V_1 \cdot t$, лежащей на ребре V_0V_1 , определяется как

$$r(t) = r_0 \cdot (1-t) + r_1 \cdot t. \quad (10)$$

Таким образом, формула для вычисления радиуса круга имеет тот же самый вид, что и формула для вычисления точки на бисекторе, являющейся центром круга.

Рассмотрим случай vs -бисектора. Здесь мы тоже хотим для точки $V(t)$ получить выражение

$$r(t) = r_0 B_0^2(t) + r_1 B_1^2(t) + r_2 B_2^2(t), \quad (11)$$

где r_0 и r_2 - радиусы максимальных пустых кругов с центрами в концевых точках бисектора V_0 и V_2 , а r_1 радиус «виртуального круга», который мы хотели бы приписать виртуальной точке кривой Безье V_1 .

В выбранной нами местной системе координат для пары сайтов A и B имеет место простое соотношение между радиусами кругов и ординатами точек бисектора $r(t) = y(t) + p$ (рис.7). Поскольку для полиномов Бернштейна справедливо тождество $B_0^2(t) + B_1^2(t) + B_2^2(t) = 1$, на основе этого получаем:

$$\begin{aligned} r(t) &= y_0 B_0^2(t) + y_1 B_1^2(t) + y_2 B_2^2(t) + p = \\ &= (y_0 + p) \cdot B_0^2(t) + (y_1 + p) \cdot B_1^2(t) + (y_2 + p) \cdot B_2^2(t) = \\ &= r_0 \cdot B_0^2(t) + (y_1 + p) \cdot B_1^2(t) + r_2 \cdot B_2^2(t). \end{aligned} \quad (12)$$

Сравнивая выражения (11) и (12) для $r(t)$, имеем $r_1 = y_1 + p$.

Поскольку $r_1 = y_1 + p = y_1 - (-p)$, из геометрических соображений получаем, что r_1 есть расстояние от точки V_1 до сайта-сегмента, взятое со знаком. В терминах алгоритма вычисления управляющей точки кривой Безье, описанного в разделе 5, получаем:

$$r_1 = \frac{[\overrightarrow{B_1 B_2} \times \overrightarrow{B_1 V_1}]}{|\overrightarrow{B_1 B_2}|}. \quad (13)$$

Таким образом, для радиуса круга с центром на параболическом бисекторе также получена простая формула такого же типа, как и для вычисления самой точки на бисекторе.

Как видно из (10), (11), для того, чтобы обеспечить возможность вычисления радиальной функции для любой точки ss -бисектора и vs -бисектора, в структуре данных скелета достаточно сохранить радиусы всех кругов с центрами в вершинах контрольного графа. При этом радиусы кругов с центрами в контрольных вершинах параболических рёбер вычисляются по формуле (13).

Однако для бисекторов третьего типа (vv -бисекторов) представление радиальной функции скелета в форме сплайнов Безье получено быть не может. Несмотря на это найденное описание ss -бисекторов и vs -бисекторов в виде (10)-(11) оказывается достаточным для весьма полезного представления фигуры в виде объединения множества пустых кругов.

Действительно, объединение всех максимальных пустых кругов с центрами на ss - и vs -бисекторах и радиусами, определяемыми их радиальными функциями, покрывают полностью фигуру. Это значит, что зная скелет фигуры и её радиальную функцию для ss - и vs -бисекторов, можно восстановить фигуру путём построения огибающей этого семейства кругов.

Пример на рис.9 демонстрирует многоугольную фигуру и её скелет (а), а также контрольный граф скелета с

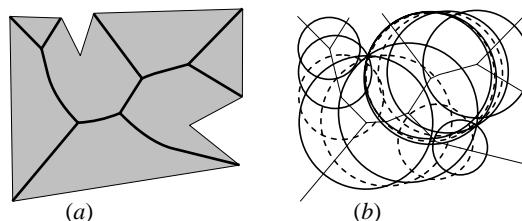


Рис 9: Многоугольная фигура и её скелет (а), контрольный граф и контрольные диски радиальной функции (b).

контрольными дисками (b).

8. ВИЗУАЛИЗАЦИЯ СКЕЛЕТА

Полученное представление скелета многоугольной фигуры в виде плоского графа, составленного из кривых Безье первой и второй степени является удобным средством для визуализации, хранения и анализа формы изображений.

Визуализация скелета может быть выполнена с помощью стандартных графических программ рисования линейных отрезков и кривых Безье. Обычно в графических библиотеках используется рисование кривых Безье третьей степени, как наиболее распространённого типа кривых. Для того, чтобы воспользоваться такими программами при изображении кривой второй степени, нужно применить известное преобразование их характеристических многоугольников. Кривая Безье второй степени с характеристическим треугольником $\{V_0, V_1, V_2\}$ совпадает с кривой Безье

третьей степени с характеристическим четырёхугольником $\{W_0, W_1, W_2, W_3\}$, у которого вершины определены следующим образом:

$$W_0 = V_0, W_1 = V_0 \cdot \frac{1}{3} + V_1 \cdot \frac{2}{3}, W_2 = V_1 \cdot \frac{2}{3} + V_2 \cdot \frac{1}{3}, W_3 = V_2.$$

Получив четырёхугольник $\{W_0, W_1, W_2, W_3\}$, можно изобразить кривую Безье третьего порядка $W(t) = W_0 B_0^3(t) + W_1 B_1^3(t) + W_2 B_2^3(t) + W_3 B_3^3(t)$, совпадающую с кривой второго порядка $V(t) = V_0 B_0^2(t) + V_1 B_1^2(t) + V_2 B_2^2(t)$.

Использование представления многоугольной фигуры в виде объединения семейства всех её максимальных пустых кругов позволяет рассматривать фигуру в качестве гибкого объекта и осуществлять над ней некоторые полезные преобразования по изменению формы. Например, можно изменить фигуру, увеличить или уменьшить её ширину за счёт изменения радиусов и положения контрольных кругов скелетного графа. Пример такого преобразования фигуры (силуэт быка переходит в силуэты жирафа и осла) представлен на рис.10.

Изменение формы фигуры за счёт коррекции скелета и радиальной функции может быть использовано в компьютерной графике для создания анимации, а также в

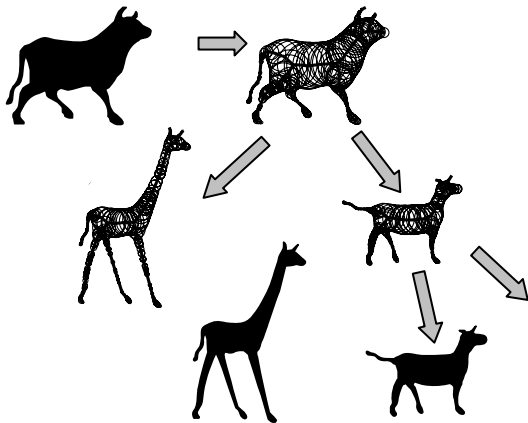


Рис.10. Преобразование формы изображения путём изменения контрольного графа скелета.

распознавании изображений для измерения сходства гибких объектов путём подгонки с использованием допустимого множества деформаций (Mestetskiy, 2009).

9. ЗАКЛЮЧЕНИЕ

Предложенный способ описания скелета в виде составной кривой Безье второго порядка даёт простой и наглядный инструмент для представления скелетов многоугольных фигур. Скелеты любой сложности теперь можно описать путём задания плоского прямолинейного графа. А полученное представление радиальной функции в форме сплайнов Безье обеспечивает возможность практического использования «циркулярного» представления многоугольных фигур как объединения вписанных в фигуру кругов. Такое представлено открывает широкие возможности для анализа и преобразования формы объектов в компьютерной графике. Интересно отметить также, что предложенный подход неожиданно связывает между собой два давно и хорошо

изученных объекта – скелет многоугольной фигуры и составную кривую Безье.

Литература

- Aichholzer O., Aurenhammer F., 1996. Straight Skeletons for General Polygonal Figures in the Plane. Lecture Notes in Computer Science. Vol. 1090. Springer-Verlag (1996), 117-126.
- Fortune S., 1987. A sweepline algorithm for Voronoi diagrams. *Algorithmica*, 2 (1987), 153-174.
- Lee, D., 1982. Medial axis transformation of a planar shape. *IEEE Trans. Pat. Anal. Mach. Int.* PAMI-4(4): 363-369, 1982.
- Siddiqi K., Pizer S.M., 2008. Medial representations: Mathematics, Algorithms and Applications. Springer, 2008.
- Yap C., 1987. An $O(n \log n)$ algorithm for the Voronoi diagram of the set of simple curve segments. *Discrete Comput. Geom.*, 2(1987), 365-393.
- Местецкий Л.М., 2009. Непрерывная морфология бинарных изображений: фигуры, скелеты, циркуляры. – М.: ФИЗМАТЛИТ, 2009.

Автор

Местецкий Леонид Моисеевич, доктор технических наук, профессор кафедры математических методов прогнозирования Московского государственного университета имени М.В. Ломоносова.

E-mail: l.mest@ru.net

Skeleton of polygonal figure – representation by the a planar linear graph

Leonid Mestetskiy
Moscow State University, Russian Federation

Abstract

The new method of the polygonal figure skeleton representation is proposed. The skeleton is a planar graph, edges of which are linear and quadratic Bezier curves. And the radial function of skeleton is represented by Bezier splines too. This approach makes possible to describe the non-linear skeleton of polygonal figure by the so-called linear control graph comprised of vertices and sides of the control polygons of Bezier curves. Skeleton with the radial function allows the alternative method for representing the figure as a union of the family of circles.

Keyword: polygonal figure, skeleton, Voronoi diagram, radial function, parabolic edges, Bezier curves, the control graph

About the author

Leonid Mestetskiy is a professor at Moscow State University, Department of Mathematical methods of Forecast. His contact email is l.mest@ru.net.

Топологическая модель многоуровневой улично-дорожной сети на основе скелета

Дмитрий Задонский¹, Елена Макарова², Иван Мехедов³
^{1,2}Факультет вычислительной математики и кибернетики

Московский государственный университет им. М. В. Ломоносова

³Вычислительный центр им. А. А. Дородницына Российской академии наук

¹jk32k@yandex.ru

²luarsoll@yandex.ru

³i.mehedov@mappl.ru

Аннотация

В работе рассматривается подход к построению топологической модели улично-дорожной сети векторной карты на основе скелетизации многоугольных фигур. Исходные данные задачи представлены в виде векторного описания контуров обочин. Требуется получить связный граф осевых линий дорог, мостов и тоннелей – модель улично-дорожной сети. Сложность задачи обусловлена наличием перекрывающихся многоугольных фигур в области дорожных развязок на карте. Способ построения такой модели опирается на идею склейки скелетов смежных многоугольных фигур. Операция склейки скелетов, описанная в статье, является новой операцией над скелетами плоских фигур.

Ключевые слова: модель улично-дорожной сети, скелет многоугольной фигуры, склейка скелетов.

1. ВВЕДЕНИЕ

В основе геоинформационных систем (ГИС) лежат различные модели данных, являющиеся отражением реальных существностей на местности, отношений между ними и других дополнительных знаний, имеющих пространственную привязку. Каждая модель данных ГИС включает в себя различные отдельные пространственные объекты, связанные между собой дополнительными топологическими отношениями.

Выбор той или иной модели в геоинформационных системах обусловлен характером решаемых задач. Широко применяется распространенная *нетопологическая модель*. В ней все объекты являются полностью независимыми друг от друга и могут произвольно размещаться в пространстве. В модели различают точечные (1-мерные), линейные (1-мерные) и площадные (2-мерные) объекты. В случае, когда данные модели определены на плоскости, точечные объекты характеризуются 2 координатами (x, y) , линейные – последовательностью точек, соединенных отрезками, площадные – несколькими (не менее одного) контурами, заданными в виде последовательности замкнутых непересекающихся линий. Такая модель хорошо подходит для описания совокупностей отдельных объектов, например, границ рек, озер, контуров зданий, границ дорог.

В то же время, такая модель не позволяет описать схему транспортных коммуникаций для последующего сетевого анализа. Для решения транспортных задач используется другая модель – модель транспортной сети. Транспортная сеть содержит два основных типа объектов – узлы и дуги. Узлы являются обычными точечными объектами, заданными координатами (x, y) на плоскости, а дуги – линейными объектами, ломаными, соединяющими пару узлов и проходящими через последовательность промежуточных точек. Каждая дуга характеризуется длиной или временем движения по ней, разрешенными направлениями движения, классом дороги или пропускной способностью и др.

В случае, когда используемая в геоинформационной системе модель данных не позволяет решать некоторые задачи (как в приведенном выше примере с нетопологической моделью данных и транспортной задачей), возникает вопрос: можно ли путем разработки внешних программных расширений ГИС перейти от одной модели к другой? Ответ на этот вопрос заключается в том, что все зависит от поставленной задачи, и в некоторых случаях это сделать возможно.

В работе рассматривается задача автоматического построения *топологической модели* многоуровневой улично-дорожной сети в виде связного графа осевых линий дорог, мостов и тоннелей. Исходные данные представлены площадными объектами, организованными в рамках нетопологической модели, но при этом для решения задачи требуется интегральный анализ формы объектов, т.е. такой анализ формы, при котором объекты рассматриваются как единое целое, а не как отдельные независимые единицы.

Данная задача является актуальной в связи с развитием навигационных систем, требующих топологического описания дорожной сети, и наличием большой накопленной базы векторных карт, не содержащих информации о топологии.

Алгоритмы автоматического построения топологической модели одноуровневой улично-дорожной сети были предложены в [3,4,6]. Описанный подход основывается на объединении всех площадных объектов в одну многоугольную фигуру с большим числом вершин и компонент связности, построение скелета такой фигуры (линии, равноудаленной от границ фигуры), и преобразовании его к модели (рис. 1).

Однако, данный подход неприменим к обработке картографических данных, содержащих дорожные развязки.

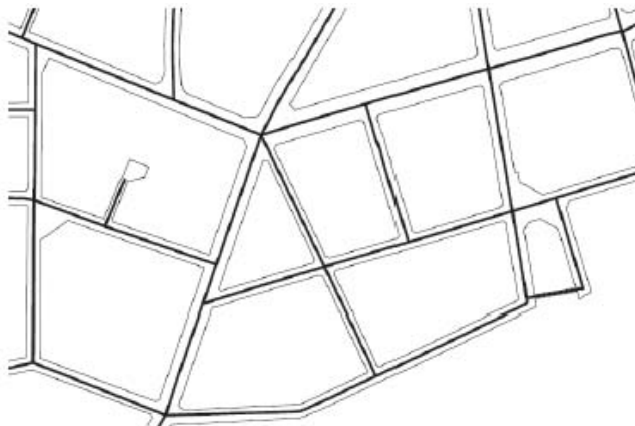


Рисунок 1. Топологическая модель одноуровневой улично-дорожной сети на основе скелетизации многоугольных фигур.

В отличие от одноуровневой дорожной сети, в нетопологической модели многоуровневой улично-дорожной сети присутствуют перекрывающиеся многоугольные фигуры (рис. 2), соответствующие пересечению проезжих частей на разных уровнях (например, мост над дорогой и тоннель под ней).

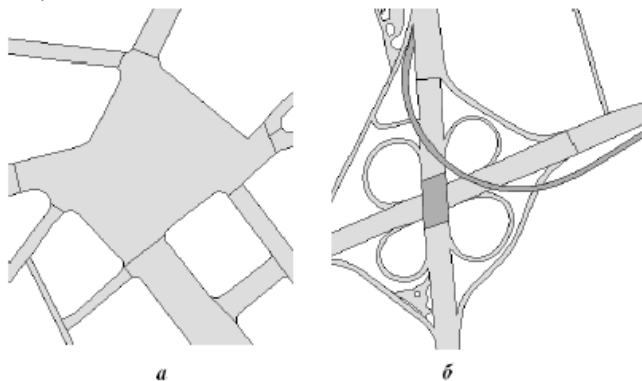


Рисунок 2. Представление одноуровневой (а) и многоуровневой (б) улично-дорожной сети в нетопологической модели данных.

Такие многоугольные фигуры не могут быть объединены как множества точек без потери информации о топологии.

В данной работе предлагается другой подход к построению модели улично-дорожной сети, основанный на скелетизации отдельных многоугольных фигур и склейке скелетов многоугольных фигур, имеющих общий участок границы (или, смежных многоугольных фигур). Такой подход дает возможность автоматического построения модели. Настоящая работа, таким образом, развивает идеи, предложенные в [3,4,6] и служит логическим продолжением этих работ.

Основное внимание в статье уделено алгоритму склейки скелетов смежных многоугольных фигур. Для удобства читателя, в следующем разделе будут приведены основные понятия, взятые из [2] и используемые далее в тексте.

2. ОСНОВНЫЕ ПОНЯТИЯ

Определение 1. Многоугольной фигурой называется часть плоскости, ограниченная конечным числом простых замкнутых непересекающихся ломаных.

Определение 2. Сайтом многоугольной фигуры называется ее вершина (сайт-точка) или сторона (сайт-сегмент).

Определение 3. Скелетом многоугольной фигуры называется множество центров максимальных пустых кругов фигуры.

Определение 4. Сайт-сегмент и сайт-точка, имеющие непустое пересечение, называются соседними сайтами.

Определение 5. Сайт-точка называется ближайшим сайтом для внутренней точки фигуры, если этот сайт является для нее ближайшей граничной точкой. Сайт-сегмент называется ближайшим сайтом для внутренней точки фигуры, если ближайшая граничная точка лежит на этом сайте или является ортогональной проекцией внутренней точки на прямую, содержащую этот сайт.

Определение 6. Ячейкой Вороного сайта называется геометрической место точек фигуры, для которых этот сайт является ближайшим.

Определение 7. Сайты называются смежными, если их ячейки Вороного имеют невырожденную границу.

Определение 8. Бисектором пары сайтов называется линия, являющаяся общей границей их ячеек Вороного.

Определение 9. Диаграммой Вороного многоугольной фигуры называется объединение бисекторов всех ее сайтов.

Определение 10. Графом смежности многоугольной фигуры называется граф (V, E) , у которого множество вершин V состоит из сайтов фигуры, а множество ребер E содержит все пары смежных сайтов из V .

Определение 11. Два многоугольника из множества граничных многоугольников многоугольной фигуры называются смежными, если в них существует пара смежных сайтов, принадлежащая разным многоугольникам.

Определение 12. Будем говорить, что для ребра триангуляции выполнено условие Делоне, если существует окружность, проходящая через вершины ребра, внутри которой нет других вершин триангуляции.

Заданное в определении 11 отношение смежности граничных многоугольников определяет граф смежности граничных контуров фигуры. Этот граф является связным, и его остов является деревом, называемым деревом смежности граничных контуров фигуры.

Заметим, что скелет многоугольной фигуры является подмножеством ее диаграммы Вороного и может быть получен из диаграммы Вороного за линейное время (по числу сайтов).

Также заметим, что граф смежности многоугольной фигуры является структурой, двойственной ее диаграмме Вороного. Обе структуры могут быть получены одна из другой за линейное время (по числу сайтов).

3. ПОСТАНОВКА И ОБЩАЯ СХЕМА РЕШЕНИЯ ЗАДАЧИ

Рассмотрим множество многоугольных фигур на плоскости, каждая из которых описывает некоторый пространственный объект на карте, соответствующий участку улично-дорожной сети. Каждому такому объекту, кроме его координат, приписана атрибутивная характеристика уровня, на основе которой все объекты делятся на дороги, мосты и тоннели (возможно и более подробное деление на мосты 1-го уровня, мосты 2-го уровня и т. д.).

Любые два объекта либо не имеют общих точек, либо имеют общий участок границы (тогда они являются составными элементами одной проезжей части), либо перекрываются (тогда они имеют разные значения характеристики уровня и входят в состав развязки). В последнем случае объекты могут как иметь общий участок границы, так и не иметь его.

Дорожную сеть в такой нетопологической модели будем называть *многоуровневой* (проезжие части могут пересекаться на разных уровнях).

Необходимо построить топологическую модель многоуровневой улично-дорожной сети в виде связанного графа осевых линий дорог, мостов и тоннелей, топологически соответствующего конфигурации улично-дорожной сети и лежащего внутри границ обочин на равном удалении от них.

Предлагается следующая схема решения задачи:

1. Построить граф смежности объектов улично-дорожной сети (два объекта будем называть *смежными*, если в их пересечении присутствуют невырожденные отрезки);
2. Построить граф смежности и скелет каждой многоугольной фигуры в отдельности;
3. Склеить скелеты многоугольных фигур, смежных в графе смежности элементов улично-дорожной сети;
4. Преобразовать полученную геометрическую конструкцию к модели улично-дорожной сети.

Пункт 1 этой схемы легко реализуется при применении алгоритма Бентли-Оттмана, описанного в [5]. Для реализации пункта 2 достаточно воспользоваться алгоритмом скелетизации многосвязных многоугольных фигур, описанном в [1]. А алгоритм преобразования скелетного графа к модели улично-дорожной сети (стрижка тупиковых ветвей, склейка близких вершин), достаточно подробно изложенный в [4], хорошо подходит для пункта 3.

Основной задачей, требующей решения, является задача склейки скелетов многоугольных фигур, имеющих общие участки границы – *смежных многоугольных фигур*, или, более строго, многоугольных фигур, в пересечении которых присутствуют невырожденные отрезки.

4. СКЛЕЙКА СКЕЛЕТОВ СМЕЖНЫХ МНОГУГОЛЬНЫХ ФИГУР

Пусть P_1 и P_2 – две многоугольные фигуры, имеющие общий участок границы, $S(P_1)$ и $S(P_2)$ – их скелеты.

В случае, если пересечение фигур совпадает с пересечением их границ, фигуры могут быть легко объединены. В этом

случае под склейкой скелетов $S(P_1)$ и $S(P_2)$ будем понимать скелет объединения фигур – $S(P_1 \cup P_2)$.

Если P_1 и P_2 пересекаются не только по границе (рис. 3а), то *склейкой скелетов* будем называть совокупность частей скелета каждой из фигур, точки которых не являются центрами максимальных пустых кругов, касающихся сайтов общего участка границы и нового участка (рис. 3б).

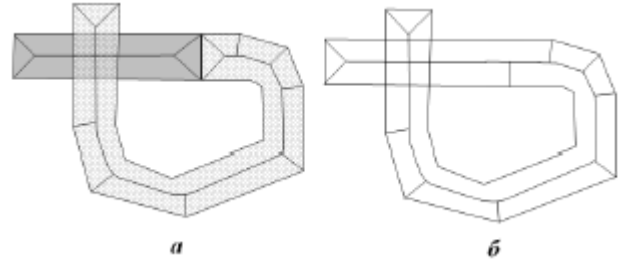


Рисунок 3. Две перекрывающиеся многоугольные фигуры и их скелеты (а) и склейка их скелетов (б).

Заметим, что идея слияния графов смежности не нова – она была предложена в **Error! Reference source not found.** для скелетизации многосвязной многоугольной фигуры, то есть для случая, когда граница фигуры состоит из нескольких контуров, причем сайты разных контуров не образуют соседних пар. В этом случае необходимо сначала построить граф смежности для каждого контура (рис. 4а – здесь и далее на всех рисунках кружками обозначены сайты-точки, а квадратами – сайты-сегменты), а затем слить эти графы смежности, для чего нужно найти начальный стежок «шва» и края сшиваемых графов смежности. Начальный стежок соединяет пару сайтов разных контуров, удовлетворяющую условию Делоне; край графа смежности внешнего граничного многоугольника определяется в результате расчистки – разделения одного из ребер на два (рис. 4б) начальный стежок (12, 9), край графа смежности внешнего контура (1, 9) и (9, 1), внутреннего – (10, 11), (11, 12), (12, 13), (13, 14), (14, 15), (15, 10)).

Далее сшивка происходит по обычному алгоритму сшивки графов смежности ломаных.

Однако рассматриваемая нами задача имеет свои особенности, заключающиеся в том, что множество общих точек границ фигуры не пусто и состоит из отрезков, поэтому множество сайтов при объединении графов смежности меняется

Алгоритм слияния графов смежности можно разделить на следующие подзадачи:

- Определение линии склейки фигур.
- Удаление из графов смежности каждой из фигур ребер, которых не будет в объединенном графе смежности («расчистка»)
- Построение объединенного графа смежности («сшивка»)

4.1 Определение линии склейки фигур

Если многоугольные фигуры имеют ровно один общий отрезок, возможны следующие основные случаи :

- Общий отрезок АВ является целой стороной каждой из фигур (рис. 5а)
- Общий отрезок DE является целой стороной одной из фигур и частью стороны другой фигуры (рис. 5б);
- Общий отрезок НК является частью стороны каждой из фигур (рис. 5в).

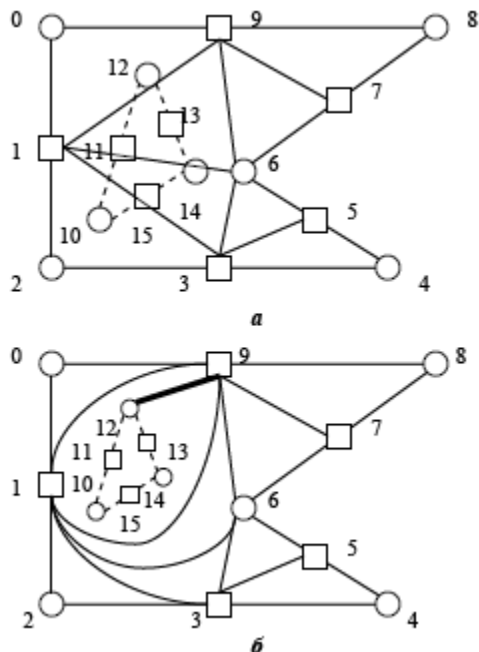


Рисунок 4. Построение графа смежности многосвязной многоугольной фигуры.

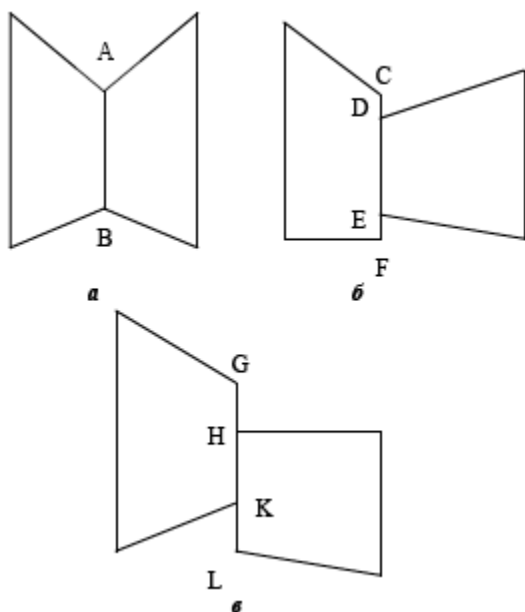


Рисунок 5. Варианты взаимного расположения фигур, пересечение которых является отрезком.

Если неизвестно множество пересекающихся сайтов (для сайтов-вершин – совпадающих, для сайтов-отрезков – имеющих общий отрезок), найдем их, воспользовавшись, например, алгоритмом плоского заметания, описанным в [5]. Этот поиск займет $O(N \cdot \log(N))$, где N – общее число сайтов.

Затем проведем сортировку пересекающихся сайтов для того, чтобы определить отдельные линии склейки. Кроме того упорядочим их таким образом, чтобы при обходе фигура, которой принадлежит сайт, оставалась справа от него. Эта сортировка займет время $O(K \cdot \log(K))$, где K – число пересекающихся сайтов.

4.2 «Расчистка» графов смежности

На этапе «расчистки» графов смежности необходимо определить множество сайтов объединенной фигуры и множество ребер графов смежности, которые сохраняются в графе смежности объединенной фигуры.

4.2.1 Сохраняющиеся сайты

Очевидно, что все сайты линий склейки, кроме конечных, будут удалены. Определим, что произойдет с начальными и конечными сайтами цепочек. Пусть у фигур F и G есть ровно одна цепочка склейки. Возможны следующие варианты:

- Цепочка склейки фигуры F начинается с сайта-точки, а цепочка фигуры G заканчивается сайтом-точкой (рис. 6а). Тогда объединим начальный сайт цепочки F и конечный сайт цепочки G в один сайт-точку, соединенный ребрами графа смежности со всеми сайтами, с которыми были соединены объединяемые сайты (рис. 6б);
- Пусть цепочка фигуры F начинается с сайта-сегмента, а цепочка фигуры G заканчивается сайтом-точкой (рис. 7а, сайты f_1 и g_5). Тогда удаляем сайт-сегмент и оставляем сайт-точку. Строим новый сайт-сегмент таким образом, чтобы начало составляющего его отрезка лежало на сайте-точке фигуры F (соседнем с удаленным конечным сайтом-сегментом цепочки), а конец – на сайте-точке фигуры g_5 (рис. 7б). Поместим новый сайт в список сайтов фигуры F .

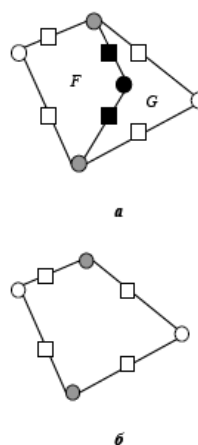


Рисунок 6. Расчистка графов смежности (цепочка начинается с сайта-точки).

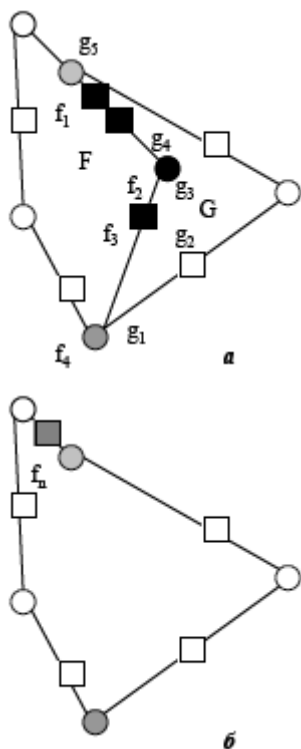


Рисунок 7. Расчистка графов смежности (цепочка начинается с сайта-сегмента).

Таким образом, построено множество сайтов объединенной фигуры. Сложность построения $O(K)$. Назовем *контрольным* сайтом сайт-точку, полученный объединением двух сайтов-точек, или новый сайт.

4.2.2 Сохраняющиеся ребра графа смежности

Утверждение. Все ребра графа смежности, не инцидентные удаляемым сайтам, сохраняются в объединенном графе смежности.

Доказательство. Рассмотрим сайты s_1 и s_2 , принадлежащие фигуре F , сохранившиеся в объединении фигур F и G и соединенные ребром графа смежности. Это значит, что существует пустой круг, касающийся только сайтов s_1 и s_2 . Этот круг, за исключением точек касания с s_1 и s_2 , лежит внутри фигуры F .

По определению смежности многоугольных фигур $\text{int}(F) \cap \text{int}(G) = \emptyset$. Значит, ни одна точка границы фигуры G не лежит внутри фигуры F . Значит, граница фигуры G не имеет общих точек с пустым кругом, касающимся s_1 и s_2 (за исключением, возможно, точки касания круга с s_1 или s_2 , если один из этих сайтов является контрольным). Новые сайты также не могут попасть в этот пустой круг, так как являются частями старых. Значит, ребро графа смежности, инцидентное сайтам s_1 и s_2 , сохранится в объединенном графе смежности.

Так как мы рассматривали любое ребро, не инцидентное удаляемым сайтам, то все ребра, не инцидентные удаляемым сайтам сохраняются в объединенном графе смежности. *Утверждение доказано.*

На рис. 8 черным выделен удаляемый сайт, серым – контрольные сайты, пунктиром – удаляемые ребра графа смежности.

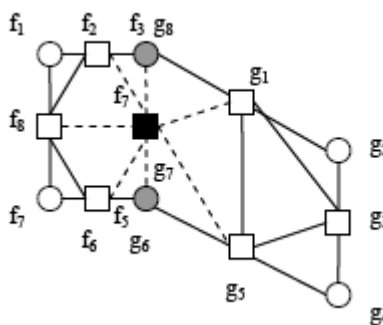


Рисунок 8. Расчистка графа смежности (сохраняющиеся сайты и сшиваемые сайты).

4.2.3 «Сшиваемые» сайты

Определим множество сайтов, которым могут быть инцидентны достраиваемые ребра графов смежности.

Так как граф смежности является планарным графом, можно говорить о гранях графа смежности. Будем называть вершиной грани графа смежности сайт (как сайт-вершину, так и сайт-сторону), инцидентный какому-либо из ребер, ограничивающих грань. Несколько сайтов образуют грань, если они и только они являются вершинами этой грани.

Утверждение. В множество сшиваемых сайтов входят все новые сайты и все сайты, принадлежащие тем же граням, что и удаляемые сайты.

Доказательство. Вхождение новых сайтов в множество сшиваемых сайтов очевидно, так как до этапа сшивки они связаны ребрами графа смежности только со своими соседними сайтами.

Не теряя общности, можно предположить, что все грани графа смежности являются треугольными, однако возможны и грани, образованные большим количеством вершин (например, если четыре и более сайта-вершины, соответствующих вогнутой вершине фигуры, окажутся на одной окружности).

Рассмотрим грань нерасчищенного графа смежности, которой не принадлежит ни один удаляемый сайт (на рис. 8 это, например, грань, образованная сайтами g_1 , g_3 и g_5). Тогда эта грань сохранится в расчищенном графе смежности. Если эта грань треугольная, она не может быть разбита новым ребром на две новые грани. Если эта грань образована более, чем тремя сайтами, существует пара сайтов, принадлежащая этой грани, но не связанная ребром графа смежности. Предположим, что на этапе «сшивки» это ребро будет построено. Это означает, что в первоначальных многоугольных фигурах не существовало пустого круга, касавшегося этих двух сайтов, а в объединении – существует. Значит, существованию этого пустого круга «мешал» какой-то из удаляемых сайтов, значит, этот сайт был смежен с концами предполагаемого нового ребра. Но это означает, что рассматриваемой грани в первоначальном графе смежности не существовало. Значит, рассматриваемая грань также не может быть разбита новым ребром на новые грани.

Рассмотрим сайт g_3 (рис. 8). Ни один удаляемый сайт в первоначальном графе смежности многоугольника G не принадлежал граням, которым принадлежит сайт g_3 . Новые ребра графа смежности не могут связать g_3 с сайтами, которые принадлежат тем же граням, которым принадлежит g_3 . Кроме того, так как граф смежности многоугольной фигуры планарен и полностью расположен внутри фигуры, сайт g_3 не может оказаться связан ребрами графа смежности с сайтами, не принадлежащими этим граням. Значит такой сайт не входит в множество сшиваемых сайтов. *Утверждение доказано.*

Поэтому внесем в множество сшиваемых сайтов только сайты, принадлежащие тем же граням, что и удаляемые сайты. На рис. 8 это сайты $f_2, f_3, g_8, g_1, g_5, g_6(f_5)$ и f_6 .

Утверждение. Каждый сайт из множества сшиваемых сайтов, связан ребрами расширенного графа смежности ровно с двумя другими сайтами из множества сшиваемых сайтов.

Доказательство. Пусть на рис. 8 это сайт f_8 . Он смежен с удаляемым сайтом f_4 и принадлежит граням (f_8, f_2, f_4) и (f_8, f_5, f_4) . При он инцидентен ровно трем ребрам, инцидентным этим граням. Ребро (f_8, f_4) при «расчистке» удалится, ребра (f_8, f_2) и (f_8, f_6) сохранятся. Аналогично для любого другого сайта, смежного хотя бы с одним удаляемым. Если же рассматриваемый сайт принадлежит только одной грани (образованной более, чем тремя сайтами), которой принадлежит и удаляемый сайт, то он связан ровно с двумя сшиваемыми сайтами, принадлежащими той же грани.

Если это новый сайт (на рис. 7 сайт f_n), то он является соседним для сайтов, соответствующих его концам, а значит, и смежным с ними. Других ребер, инцидентных этому сайту, в расширенном графе смежности не существует. Кроме того, вершина, являющаяся концом ровно одного общего отрезка (на рис. 5 сайты $f_3(g_8)$ и $g_6(f_3)$), является соседней ровно для одного сайта каждой фигуры либо соединена ровно с одним сшиваемым сайтом в каждой фигуре. *Утверждение доказано.*

Значит, множество сшиваемых сайтов можно представить в виде одной или нескольких цепочек, упорядоченных таким образом, что каждый сайт смежен с предыдущим и следующим.

Строить множество сшиваемых сайтов лучше всего параллельно с этапом «расчистки», то есть при удалении ребер, инцидентных удаляемым сайтам, вписывать в множество сшиваемых сайтов все неудаляемые сайты, принадлежащие граням, инцидентным этим ребрам. Заметим, что, если хранится информация о том, какие грани инцидентны каким ребрам и наоборот, то при этом возможно построить упорядоченные замкнутые цепочки.

Заметим, что расчистка графа смежности и построение цепочек сшиваемых сайтов по сложности не превышает $O(m \cdot k)$, где k – число удаляемых сайтов, а m – число сшиваемых сайтов.

4.3 «Сшивка» графов смежности

Даны «расчищенные» графы смежности многоугольных фигур, являющиеся подмножествами объединенного графа смежности. Необходимо достроить объединенный граф смежности.

Так как в большинстве случаев число удаляемых сайтов и число сайтов, смежных с одним конкретным сайтом невелики

по сравнению с числом всех сайтов фигуры, то и число сшиваемых сайтов оказывается невелико по сравнению с числом всех сайтов. Поэтому для «сшивки» можно применить «жадный» алгоритм (для диаграмм Вороного описан в [2], для графов смежности аналогичен). Время сшивки по этому алгоритму будет $O(m^2)$, где m – число сшиваемых сайтов.

Однако в данной задаче возможно также использовать алгоритм, основанный на принципе «разделяй и властвуй», имеющий сложность $O(m \cdot \log m)$, где m – число сшиваемых сайтов.

Предположим сначала, что при расчистке графа смежности образовалась ровно одна замкнутая цепочка сшиваемых сайтов (на рис. 9а сайты замкнутой цепочки выделены серым).

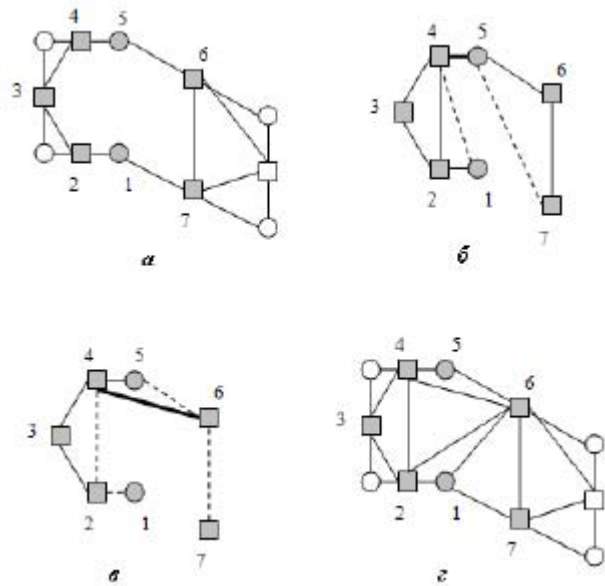


Рисунок 9. Сшивка графов смежности.

Будем разбивать цепочку примерно пополам и рекурсивно строить граф смежности каждой из частей цепочки, а затем – сливать эти графы смежности.

Для того, чтобы воспользоваться этим методом решения задачи, необходимо уметь сливать графы смежности цепочек (например, сайты фигуры на рис. 9а можно разбить на цепочки 1, 2, 3, 4 и 5, 6, 7).

Пусть построены графы смежности двух цепочек и определены их внутренние края (если ребро является ребром расчищенного графа смежности, то оно считается принадлежащим внутреннему краю цепочки, если не инцидентно ни одной грани, образованной сайтами только рассматриваемой цепочки; если ребро построено на этапе «сшивки», то оно принадлежит внутреннему краю цепочки, если инцидентно ровно одной грани, образованной сайтами только рассматриваемой цепочки; на рис. 9б внутренние края цепочек изображены пунктиром)

Так как сшиваются цепочки, последний сайт одной из которых в расчищенном графе смежности связан с первым сайтом другой, то ребро, соответствующее этой связи, будем считать начальным стежком (например, ребро (4, 5) на рис. 9б).

Один конец стежка всегда принадлежит одной цепочке, другой – другому.

Сначала производится проверка условия Делоне для ребер внутреннего края каждой из цепочек, смежных с текущим стежком. Условие Делоне выполняется, если ребро графа смежности одной из цепочек существует с учетом всех сайтов другой. Если для ребра оно не выполняется, то данное ребро разрушается, во внутренний край на его место помещаются ребра грани, разрушенной при удалении этого ребра. Удаление происходит до тех пор, пока не выполнится условие Делоне. На рис. 9б удаляются ребра (4, 1) и (5, 7); ребрами внутреннего края становятся ребра (4, 2), (1, 2), (5, 6) и (6, 7).

Далее, для каждого из концов стежка и сайта, связанного ребром внутреннего края с другим концом стежка проверяются условия Делоне (то есть на рис. 9в при стежке (4, 5) для пар (5, 1) и (4, 6)). Условие может быть выполнено не более, чем для одной пары; если не выполняется ни для одной – алгоритм завершает свою работу. Если выполняется, то ребро внутреннего края, ставшее ребром новой грани, перестает быть ребром внутреннего края, а новое ребро становится текущим стежком и алгоритм продолжает работу. Очевидно, что на рис. 9в необходимо соединить сайты (4, 6), затем (6, 2), (6, 1) и (1, 7).

После завершения работы алгоритма необходимо выполнить операцию дополнительной расчистки – проверку условия Делоне построенных ребер графа смежности с сайтами внешнего края цепочек. Если условие не выполняется – ребра разрушаются. В [2] показано, что этот алгоритм корректен.

Для случая существования только одной цепочки сшиваемых сайтов построение графа смежности завершено (рис. 9г).

Алгоритмическая сложность этого этапа $O(m \cdot \log m)$, где m – число сшиваемых сайтов.

Если замкнутых цепочек сшиваемых сайтов оказывается более одной, то существует два основных случая (для двух цепочек):

- Одна цепочка состоит из сшиваемых сайтов обеих фигур; другая состоит из сшиваемых сайтов только одной фигуры (рис. 10а, 10б). В этом случае все сшиваемые сайты второй цепочки будут смежными только с сайтами первой, поэтому можно воспользоваться алгоритмом слияния графов смежности на основе дерева смежности многоугольной фигуры, описанным в [1].
- Если обе цепочки сшиваемых сайтов содержат сайты как одной, так и другой фигур и образуют в расчищенном графе смежности две грани (рис. 10в, 10г), с каждой из этих граней можно работать независимо, то есть применить алгоритм сначала к одной, потом к другой цепочке.

5. ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ

Для проведения вычислительного эксперимента были использованы пространственные данные Единой государственной картографической основы г. Москвы [7] и геоинформационная система «ГИС Mappl» [8].

Был реализован встраиваемый в геоинформационную систему компонент, осуществляющий склейку скелетов смежных многоугольных фигур (слияние их графов смежности).

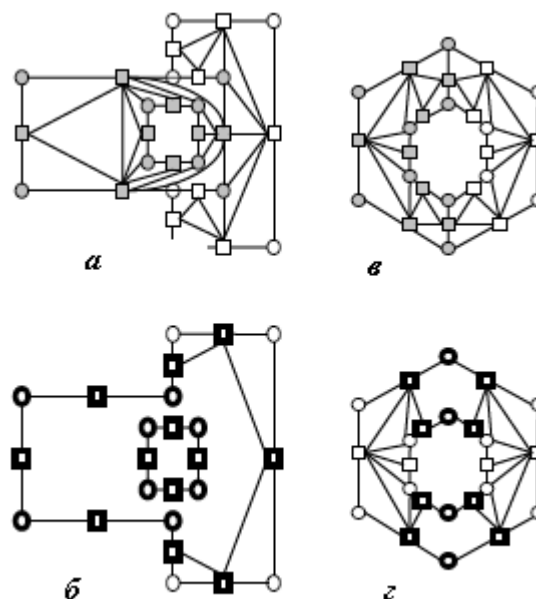


Рисунок 10. Сшивка графов смежности (в случае, если замкнутых цепочек сшиваемых сайтов более одной).

Для проведения эксперимента был выбран участок карты, содержащий развязку (рис. 11а) (несколько многоугольных фигур, соответствующих участкам проезжих частей, входящих в состав развязки).

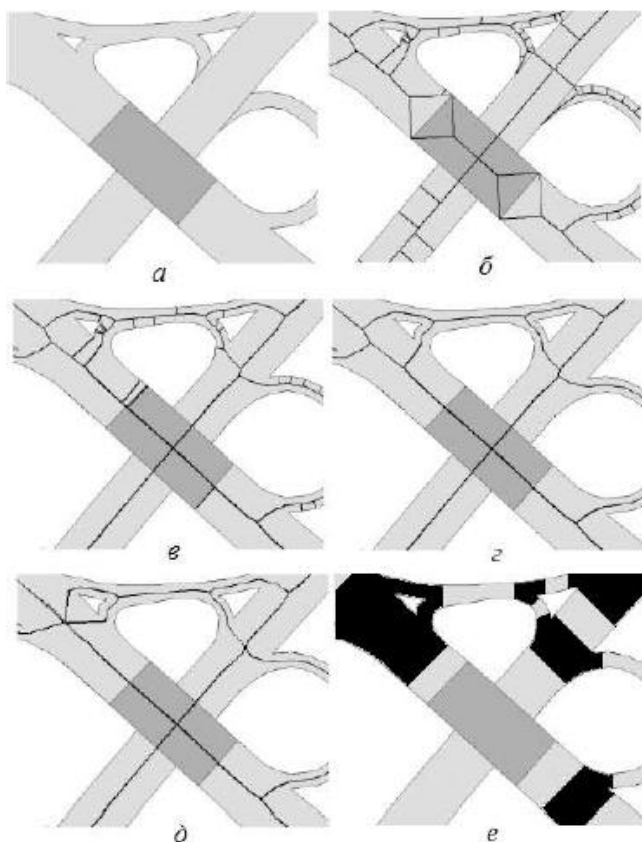
В автоматическом режиме была построена топологическая модель участка (рис. 11д), а также выделены перекрестки (рис. 11е). Задача автоматического выделения перекрестков с помощью топологической модели улично-дорожной сети описана в [3].

Результаты вычислительного эксперимента представлены на рис. 11.

6. ВЫВОДЫ

Таким образом, авторами были получены следующие результаты:

1. Предложен, теоретически обоснован и программно реализован метод склейки скелетов многоугольных фигур, имеющих общий участок границы. Вычислительная сложность алгоритма – $O(N \cdot \log N)$, где N – общее число вершин в обеих многоугольных фигурах;
2. Разработан алгоритм автоматического построения топологической модели многоуровневой улично-дорожной сети. Данная задача геоинформатики может считаться полностью решенной.



7. СПИСОК ЛИТЕРАТУРЫ

- [1] Местецкий Л.М. *Скелетизация многоугольной многосвязной фигурой на основе дерева смежности ее границы*, Сибирский журнал вычислительной математики, Т.9, № 3, сс. 299 – 314, 2006.
- [2] Местецкий Л.М. *Непрерывная морфология бинарных изображений: фигуры, скелеты, циркуляры*, – М.: ФИЗМАТЛИТ, 2009.
- [3] Мехедов И.С. *Поиск шаблонов перекрестков на векторной карте городской улично-дорожной сети*, В сборнике докладов 14-й Всероссийской конференции «Математические методы распознавания образов», М.: МАКС Пресс, сс. 414 – 417, 2009.
- [4] Мехедов И.С., Козлов А.В. *Модель улично-дорожной сети на основе скелета*, В сборнике докладов 19-й международной конференции «Графикон-2009», М.: МАКС Пресс, сс. 356 – 359, 2009.
- [5] Препарата Ф., Шеймос М. *Вычислительная геометрия*, - М.: Мир, 1989.
- [6] Haunert J.-H., Sester M. *Area Collapse and Road Centerlines Base on Straight Skeletons*, Geoinformatica, Vol. 12, № 2, pp. 169 – 191, 2008.
- [7] <http://www.egko.ru>
- [8] <http://www.mappl.ru>

ОБ АВТОРАХ

Иван Мехедов является аспирантом Вычислительного центра РАН и работает в ООО «Маппл Групп» (геоинформационные технологии)

Елена Макарова и Дмитрий Задонский являются студентами факультета Вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова.

ABSTRACT

A method for a graph-based representation of street networks is developed. The source data are presented by the set of polygonal figures that are geographic map objects describing traffic area boundaries. These polygonal objects may overlap in the plain (corresponding to the different-level parts of roads, for example, a tunnel and a bridge) and, therefore, classical skeletonization methods are not appropriate for automatic generating graph-based representation of street networks. The approach proposed in the paper is based on a novel fusing technique which is used to sew local skeletons together.

Keywords: *topological model of street network, skeleton of a polygonal figure, fusion of skeletons.*

Верификация алгоритмов фотореалистического рендеринга кристаллов

В.А. Дебелов[†], Д.С. Козлов^{*}

[†]Институт вычислительной математики и математической геофизики СО РАН,

^{*}Новосибирский государственный университет, Новосибирск, Россия

{debelov | kozlov}@oapmg.sccc.ru

Аннотация

В данной работе предлагается подход для верификации алгоритмов фотореалистического рендеринга изотропных и анизотропных кристаллов, аналогичный подходу, с успехом применяемому в вычислительной математике, например, Matrix Market¹. В области глобальной освещенности таким примером является классический Cornell Box [1]. Подход предполагает наличие некоторого расширяемого множества тестов, основанных на известных законах распространения лучей на границах сред, спецификациях тестовых сцен, фотографиях этих сцен (если это возможно) и изображений тестовых сцен, рассчитанных различными алгоритмами рендеринга.

Отметим, что наличие результатов работы различных алгоритмов (как в Cornell Box²) позволяет прикладному программисту выбрать тест наиболее подходящий для его целей.

Ключевые слова: Фотореалистический рендеринг, кристаллы, верификация, двойное лучепреломление, поляризация, дисперсия, коническая рефракция, интерференция, тестирование алгоритмов рендеринга.

1. ВВЕДЕНИЕ

В последнее время в трудах конференций по компьютерной графике стали чаще появляться статьи, посвященные фотореалистической визуализации кристаллов, например, [2–6]. Однако, только в одной из них [3] сделана попытка верифицировать физическую корректность рассчитанных изображений. Авторы приводят рассчитанное изображение одноосного прозрачного кристалла в сравнении с соответствующей фотографией кристалла. Приводятся чёрно-белые изображения и выбрана очень специфичная сцена, поскольку представленный в работе алгоритм не позволяет рассчитать изображение кристалла в общем случае. В общем же случае задача верификации алгоритмов фотореалистического рендеринга кристаллов не решена.

Для проверки алгоритмов фотореалистического рендеринга сцен, содержащих только непрозрачные диффузные поверхности, в работе [1] была предложена сцена, известная как Cornell Box, которая с тех пор стала де-факто стандартом для тестирования таких алгоритмов. Позднее были предложены вариации сцены, включающие полупрозрачные объекты, см. [7] и Global Illumination Test Scenes³.

Верификация методом сравнения с фотографией часто оказывается очень сложной задачей: необходимо подобрать геометрию и структуру кристалла, ракурс и параметры камеры, геометрию и спектры источников света. Например, тестирование алгоритмов излучательности показало, что при попиксельном сравнении рассчитанных изображений сцены Cornell

Box с фотографией очень сильно проявляются неровности реальных поверхностей, игнорируемые при расчёте изображений сцены, несовпадение геометрий источников и т.д., что приводит к тому, что изображения совпадают не полностью. Это очень сильно затрудняет проведение такого сравнения.

В кристаллах может проявляться ряд оптических эффектов, связанных с поляризацией света, двойное лучепреломление и коническая рефракция, для проверки расчёта корректности которых должны создаваться специфические сцены, например, для верификации расчёта двойного лучепреломления необходимо иметь набор контрастных текстур, а для явления конической рефракции необходимо создавать специфический источник света и очень специфичное расположение кристалла.

Удобно иметь некоторую систему более простых тестов, позволяющих хотя бы частично верифицировать корректность алгоритма, не прибегая к сравнению с фотографией. Это позволит на ранних стадиях разработки и отладки алгоритма проверить правильность решения частных задач. Пример теста: неполяризованный луч падает на границу между вакуумом и изотропным кристаллом под углом Брюстера, рассчитать отраженный и преломленный лучи и их состояние поляризации.

Авторы предлагают инициировать данную работу, начиная с рассмотренной ниже системы тестов и специальных сцен, которые позволяют проверить корректность алгоритмов расчета фотореалистических изображений полупрозрачных изотропных, одноосных и двуосных кристаллов.

2. СВЕДЕНИЯ ИЗ КРИСТАЛЛООПТИКИ

Полупрозрачные кристаллы по оптическим свойствам можно разделить на: изотропные, одноосные и двуосные кристаллы [8–10]. Последние два типа кристаллов являются анизотропными средами, в них проявляются эффекты не свойственные изотропным средам. Рассмотрим каждый из эффектов в объёме необходимом для дальнейшего изложения. Более подробное описание можно найти в любой книге по кристаллооптике, например, в [9, 10].

2.1 Основные понятия

1. Световой луч. Световым лучом называется набор электромагнитных волн, распространяющихся в одном направлении. Каждая из волн может иметь различную плоскость колебаний электрического и магнитного поля и разный сдвиг фаз. Если все волны имеют одну и ту же частоту, то такой луч называется монохроматическим, в противном случае луч называется полихроматическим, и имеет смысл говорить о спектре луча.

2. Поляризация света. Луч называется частично поляризованным, если существует преимущественная плоскость колебаний векторов электрического поля. Если такая плоскость единственная, то луч *линейно поляризован* в этой плоскости –

¹ <http://math.nist.gov/MatrixMarket>

² <http://www.graphics.cornell.edu/online/box>

³ <http://www.cs.utah.edu/~bes/papers/scenes>

плоскости поляризации. Различают также круговую и эллиптическую поляризации. Если луч не поляризован, говорят, что он обладает естественной поляризацией.

3. Оптическая ось. В анизотропных кристаллах направление распространения фронта волны (направление распространения фазы) в общем случае не совпадает с направлением распространения луча (направлением распространения энергии). При любом заданном направлении распространения фазы в кристалле могут распространяться две волны с различными скоростями, плоскостями поляризации и направлениями распространения лучей. Исключение составляет два направления оптических осей I рода или *бинормалей*, которым соответствует по одному направлению луча. Аналогично, каждому направлению распространения луча соответствует два направления распространения фазы, за исключением двух направлений оптических осей II рода или *бирадиалей* [10]. В одноосных кристаллах оптическая ось является одновременно бинормалью и бирадиалью. В зависимости от наличия и числа оптических осей кристаллы делятся на изотропные (нет оптических осей), одноосные (одна ось) и двуосные (две *бирадиали*, две *бинормали*).

2.2 Оптические эффекты в кристаллах

1. Дисперсия света. При преломлении на границе сред с разными коэффициентами преломления полихроматический луч расщепляется на набор своих монохроматических составляющих. Дисперсия в изотропных кристаллах широко известна как явление разложения света на спектр.

2. Двойное лучепреломление. При преломлении монохроматического луча на границе анизотропных кристаллов образуется два преломлённых луча, поляризованных во взаимно перпендикулярных направлениях. Такое явление называется двойным лучепреломлением. При падении луча на границу между двумя анизотропными кристаллами образуется до двух отражённых и двух преломлённых лучей. В одноосных кристаллах один из лучей называется обыкновенным, а второй – необыкновенным, в двуосных – быстрый и медленный лучи. На рис. 1 приведена фотография кальцита – одноосного кристалла с ярко выраженным двойным лучепреломлением.



Рис. 1. Двойное лучепреломление в кристалле кальцита⁴

3. Анизотропное поглощение или плеохроизм. Коэффициент поглощения в анизотропных кристаллах зависит от направления распространения луча и его состояния поляризации, что приводит к наблюдению нескольких выраженных окрасок по различным направлениям наблюдения кристалла. В одноосных кристаллах можно обнаружить две выраженные окраски (дихроизм), а двуосных – три окраски (плеохроизм).

4. Интерференция света. Сложение интенсивностей двух или нескольких волн, сопровождающееся чередованием в пространстве максимумов и минимумов интенсивности. Результат интерференции (интерференционная картина) зависит от разности фаз накладывающихся волн.

Примером интерференционных картин являются коноскопические фигуры (рис. 2), получаемые в тонких кристаллических пластинках с помощью петрографического микроскопа.

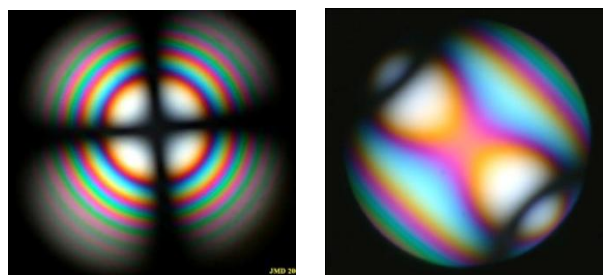


Рис. 2. Примеры коноскопических фигур для одноосного кристалла кальцита (слева) и двуосного кристалла мусковита⁵

5. Коническая рефракция. Явление конической рефракции наблюдается только в двуосных кристаллах. При определённых условиях луч света, преломлённый на границе кристалла, порождает полный конус лучей. Различают внешнюю и внутреннюю коническую рефракцию (рис. 3).

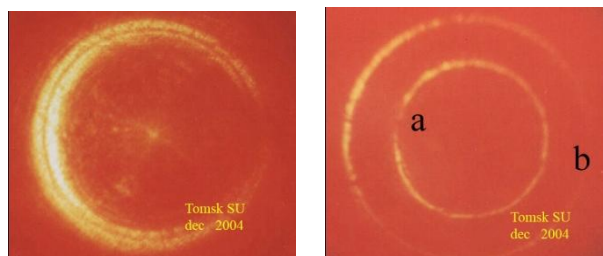


Рис. 3. Фотографии внутренней (слева) и внешней конической рефракции⁶

3. ЧИСЛЕННЫЕ ТЕСТЫ

Для расчёта фотореалистических изображений кристалла, как правило, используется трассировка лучей, которая подразумевает наличие локальной модели, т.е. алгоритма расчета взаимодействия луча с границей сред (например, вакуум-кристалл, кристалл-кристалл). Данная глава предназначена для тестирования корректности этой части алгоритма рендеринга.

Здесь мы рассматриваем алгоритм как процедуру, принимающую на вход направление падающего луча, его интенсивность и состояние поляризации, нормаль к плоскости раздела сред и оптические характеристики каждой среды, результатом вычислений которой является набор направлений лучей, их интенсивностей и состояний поляризации.

Отметим, что хотя и существует несколько математических представлений состояния поляризации луча, например, вектора Стокса и матрицы Мюллера, матрицы когерентности и модификаторы матриц когерентности и т.д., см. [9], предлагаемый набор тестов не зависит от такого представления.

Численные тесты выбраны таким образом, чтобы проверка корректности их расчёта была достаточно простой и не приводила к появлению дополнительных ошибок.

3.1 Закон Снеллиуса

При падении луча на границу между двумя **изотропными** средами угол падения равен углу отражения, а преломление описывается законом Снеллиуса: падающий, отражённый и

⁴ http://ru.wikipedia.org/wiki/Двойное_лучепреломление

⁵ <http://jm-derochette.be/conoscopy.htm>

⁶ <http://www.demophys.tsu.ru/Original/Hamilton/Hamilton.html>

преломлённый лучи, а также нормаль к поверхности лежат в одной плоскости – *плоскости падения*, а угол падения света на поверхность связан с углом преломления соотношением:

$$n_1 \sin \alpha_1 = n_2 \sin \alpha_2, \quad (1)$$

где n_1, n_2 – коэффициенты преломления сред, из которой свет падает на границу раздела и на которую падает свет, а α_1, α_2 – углы падения и преломления, соответственно (рис. 4).

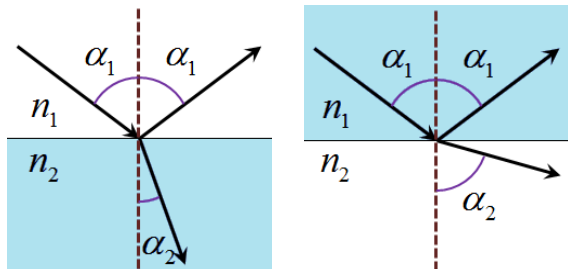


Рис. 4. Преломление и отражение луча света на границе двух изотропных сред: при падении из менее плотной среды в более плотную (слева) и наоборот. Пунктирной линией показана нормаль к границе раздела в точке падения

Отметим, при падении луча из более *плотной* среды (среды с большим коэффициентом преломления) в менее плотную, при больших углах падения преломленный луч может отсутствовать. Это происходит, если

$$\alpha_1 > \arcsin(n_1/n_2), \quad (2)$$

тогда α_2 оказывается комплексным и луч не существует. Это явление называется *полным внутренним отражением*. При этом отражённый луч имеет интенсивность равную интенсивности падающего луча и поляризован эллиптически [10].

Таким образом, в процессе верификации необходимо проверить, что все три луча и нормаль лежат в одной плоскости, угол падения равен углу отражения и выполняется условие (1). При выполнении условия (2) необходимо проверить отсутствие преломлённого луча, равенство интенсивностей падающего и отражённого лучей, а также то, что отражённый луч поляризован эллиптически.

3.2 Угол Брюстера

При падении луча на границу раздела двух диэлектрических сред происходит частичная поляризация преломлённого и отражённого лучей, за исключением случая перпендикулярного падения луча.

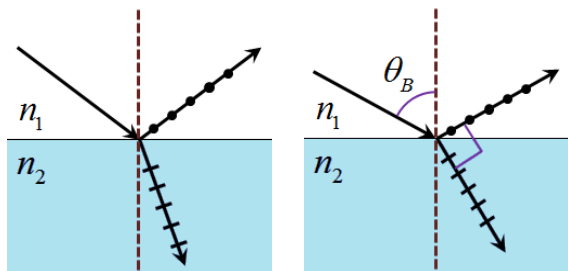


Рис. 5. Частичная поляризация при падении луча на границу сред (слева). Полная поляризация отражённого луча при падении луча под углом Брюстера

Рассмотрим случай падения луча на границу двух **изотропных** сред. Колебания вектора электрического поля в преломлённом луче лежат в плоскости падения (луч поляризован в плоскости падения), а в отражённом – перпендикулярны ей

[11]. На рис. 5 обозначены колебания вектора электрического поля в плоскости рисунка, а точками в плоскости, перпендикулярной плоскости рисунка.

При падении естественно поляризованного луча под углом Брюстера

$$\theta_B = \arctan(n_1/n_2), \quad (3)$$

отражённый луч полностью поляризован, а отражённый и преломлённый лучи образуют прямой угол (рис. 5, справа).

При преломлении и отражении луча *степень поляризации* луча (относительное количество поляризованного света) может только увеличиться, т.е. при падении луча поляризованного в плоскости падения, отражённый луч будет отсутствовать. Если же падающий луч поляризован перпендикулярно плоскости падения, то будет отсутствовать отражённый луч [11]. В обоих случаях интенсивность второго луча равна интенсивности падающего.

Таким образом, верно следующее:

- 1) При падении неполяризованного луча и выполнении условия (3) отражённый луч полностью поляризован перпендикулярно плоскости падения, а угол между отражённым и преломлённым лучом прямой.
- 2) При падении луча поляризованного в плоскости падения отражённый луч отсутствует, а интенсивность преломлённого луча равна интенсивности падающего.
- 3) При падении луча поляризованного перпендикулярно плоскости падения преломлённый луч отсутствует, а интенсивность отражённого луча равна интенсивности падающего.

3.3 Двойное лучепреломление

Для верификации корректности расчета двойного лучепреломления рассмотрим наиболее простой и часто приводимый в учебниках по оптике случай: нормальное падение неполяризованного луча на **одноосный** кристалл из вакуума.

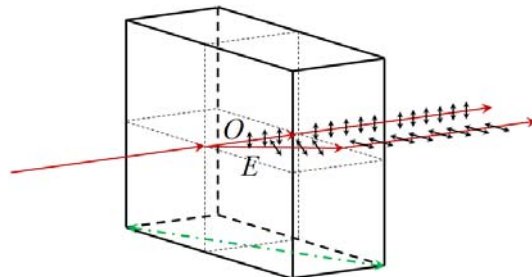


Рис. 6. Направление и поляризация отражённых и преломленных лучей при двойном лучепреломлении на границе одноосного кристалла

Для одноосного кристалла определяется два главных коэффициента преломления: для обыкновенного луча n_o и необыкновенного луча n_e . Реальный коэффициент преломления необыкновенного луча лежит в интервале между n_o и n_e и зависит от угла падения и направления оптической оси.

При преломлении в кристалле образуются два взаимно перпендикулярно линейно поляризованных луча (рис. 6). Луч, поляризованный перпендикулярно *главной плоскости падения* (плоскости, образованной нормалью к границе и направлением оптической оси), называется обыкновенным (обозначен O на рисунке). Второй луч E , поляризованный в *главной плоскости падения*, называется необыкновенным. Короткими стрелками обозначены колебания вектора электрического поля. Направление оптической оси обозначено штрих пунктирной стрелкой снизу.

Падающий, обыкновенный и необыкновенный лучи, нормаль к границе раздела и оптическая ось лежат в *главной плоскости падения* [10]. Отражённый и падающий луч противоположны по направлениям.

После выхода из кристалла лучи становятся параллельными падающему лучу, обыкновенный луч поляризован перпендикулярно главной плоскости падения, а необыкновенный в самой плоскости. Это верно для любого угла падения.

Для любого угла падения для обыкновенного луча выполняется закон Снеллиуса, при этом в качестве коэффициента n_2 необходимо использовать главный коэффициент преломления для обыкновенного луча n_o .

Таким образом, справедливы следующие утверждения:

- 1) При нормальном падении. Падающий, обыкновенный и необыкновенный лучи, лежат в *главной плоскости падения*. Отражённый и падающий луч противоположны по направлениям. Обыкновенный луч поляризован перпендикулярно главной плоскости падения, а необыкновенный – в плоскости.
- 2) Для любого угла падения. Соотношение углов падения и преломления для обыкновенного луча удовлетворяет уравнению (1). После преломления на второй границе обыкновенный и необыкновенный лучи параллельны падающему, при этом необыкновенный луч поляризован в главной плоскости падения, а обыкновенный – перпендикулярно ей.

4. ВИЗУАЛЬНЫЕ ТЕСТЫ

Хотя сравнение с фотографией является в общем случае очень сложной задачей, в некоторых сценах оно оказывается осуществимым. В данной главе мы рассмотрим несколько примеров таких сцен. Этот метод тестирования подходит для всех алгоритмов фотореалистического рендеринга кристаллов. Алгоритм рассматривается как процедура, принимающая на вход геометрию и оптические характеристики объектов и источников сцен, а также параметры и ракурс камеры. Результатом работы является рассчитанное RGB-изображение.

4.1 Рендеринг конической рефракции

Для наблюдения внутренней конической рефракции из двуслоного кристалла вырезают кристаллическую пластинку **C** толщиной **d**, перпендикулярной одной из его бинормалей. Вторая бинормаль χ при этом отклонена в сторону. Сверху на пластинку накладывается фильтр **F** толщиной **h** с малым отверстием радиуса **R** (рис. 7, слева). Снизу под пластинкой на расстоянии **D** от нижней плоскости кристалла параллельно ей располагается прямоугольный экран **S**.

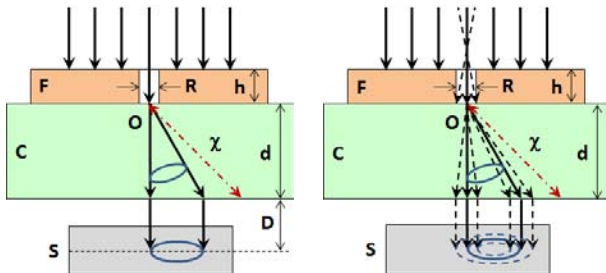


Рис. 7. Идеальная (слева) и реальная схема распространения лучей в случае внутренней конической рефракции

При освещении этой конструкции сверху неполяризованным светом, падающим под прямым углом, через отверстие **O** проходит тонкий пучок света. Пучок, преломляясь на первой границе, образует полый конус преломлённых лучей. После второго преломления конус превращается в полый цилиндр.

На практике через отверстие **O** в экране проходит не только луч перпендикулярный пластинке, но и лучи близкие к нему по направлению (рис. 7, справа). Эти лучи испытывают двойное лучепреломление, а не коническую рефракцию. Расходимость падающего на пластинку пучка определяется отношением **R** к **h**. В результате вместо одного светлого кольца образуется двойное кольцо, с тёмным кольцом внутри (рис. 3). Светлые кольца образуются из-за двойного лучепреломления, а тёмное – конической рефракцией.

Отметим, что если повернуть направление второй оптической оси на некоторый угол (т.е. повернуть пластинку **C**), то двойное кольцо, образованное на экране **S**, также повернётся на этот же угол.

Таким образом, предлагаются следующие тесты:

- 1) Рассчитать изображение реальной сцены, формирующиеся на экране **S**, проверить, что образуется двойное кольцо конической рефракции.
- 2) При повороте второй оптической оси на некоторый угол, двойное кольцо на экране **S** поворачивается на этот же угол.

4.2 Рендеринг петрографического шлифа

Петрографический шлиф представляет собой пластинку минерала или горной породы толщиной порядка 30 мкм, наклеенной на стеклянную подложку при помощи канадского бальзама (смола с коэффициентом преломления близким к коэффициенту преломления стекла) (рис. 8, слева). Изучение таких шлифов происходит с помощью петрографического микроскопа.

Если шлиф содержит анизотропные минералы, то при рассмотрении через скрещенные поляризаторы его различные части окрашиваются в зависимости от присутствующего в той или иной части шлифа того или иного минерала. Окраска образуется из-за интерференционных эффектов, возникающих при таком изучении.

Существует два способа изучения шлифа: ортоскопия и коноскопия. В первом случае на шлиф падает параллельный пучок лучей, а во втором сходящийся пучок лучей с фокусом внутри кристаллической пластинки. Подробнее см. [12, 13].



Рис. 8. Структура петрографического шлифа (слева) и схема распространения лучей при изучении шлифа методом коноскопии

На рис. 8 справа приведена **общая** схема распространения лучей в шлифе в случае коноскопии. Неполяризованные лучи испускаются источником света (верхняя жирная линия на рисунке), затем они поляризуются, проходя через поляризатор **6**. Далее лучи фокусируются, проходя через линзу **8**. В шлифе лучи проходят последовательно: покровное стекло **1**, слой канадского бальзама **2**, кристаллическую пластинку **3**, второй слой канадского бальзама **4** и стеклянную подложку **5**. При прохождении верхней границы кристаллической пластинки из-за двойного лучепреломления образуется два луча,

распространяющихся в разных направлениях и с различной скоростью. После преломления на второй границе лучи приобретают исходное направление, но имеют разность фаз, зависящую от толщины пластинки и оптических характеристик минерала в этой части шлифа.

Проходя через линзу **9** лучи вновь становятся параллельными. Затем, проходя через второй поляризатор (анализатор) **7**, все лучи обретают одинаковую поляризацию и интерферируют, образуя на экране (нижняя жирная линия) коноскопические изображения (рис. 2).

Отметим, что слои **1**, **2**, **4** и **5**, являясь изотропными, не вносят дополнительной разности фаз и оказывают малое влияние на получаемые коноскопические изображения, таким образом, при расчёте изображения их можно не учитывать.

Предлагаются следующие тесты:

- 1) Рассчитанные коноскопические изображения, образованные на экране, должны быть визуально похожи на фотографии, получаемые с помощью петрографического микроскопа. Коллекцию таких фотографий и краткое описание параметров их получения можно найти, например, здесь⁷.
- 2) При повороте оптических осей на некоторый угол коноскопическая картина поворачивается на этот же угол.

4.3 Рендеринг трёхмерной сцены

В данной главе мы рассмотрим две связанные сцены: виртуальную сцену, используемую для расчета изображения, и соответствующую ей реальную сцену, фотография которой используется для сравнения с рассчитанным изображением, а также метод их сопоставления. Отметим, что предложенный метод сопоставления виртуальной и реальной сцены позволяет это сделать, используя только легкодоступные материалы и оборудование, и не требует наличия спектрометра для измерения спектров источников и спектров отражения материалов в реальной сцене, а также крайне труднодоступных данных о спектральной чувствительности матрицы фотоаппарата и прозрачности используемого объектива.

Проверка корректности рендеринга этой сцены проводится на **заключительном** этапе тестирования. Критерием правильности расчёта данной сцены, является то, что рассчитанное изображение визуально очень похоже на фотографию.

4.3.1 Виртуальная сцена

Виртуальная сцена – это бокс с односторонними стенками: прозрачными снаружи, но непрозрачными внутри. Стенки являются источниками, излучающими свет внутрь бокса (рис. 9). Излучение каждой точки стенки бокса неполяризованное и равномерно по всем направлениям. Спектр излучения определяется наложенной на стенку бокса текстурой.

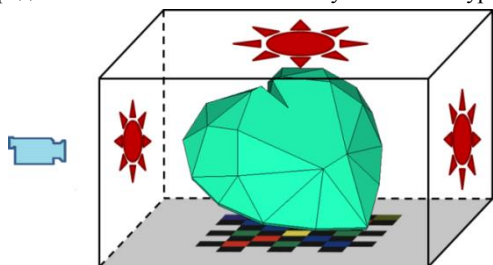


Рис. 9. Схема виртуальной сцены

Использование спектральных текстур не является удобным в процессе сравнения с фотографией, поэтому можно использовать RGB-текстуры (см. ниже).

Для сравнения все источники-стенки за исключением нижнего выключаются, т.е. на них накладывается чёрная неизлучающая текстура. На нижнюю стенку бокса накладывается текстура, показанная на рис. 10 слева. Текстура состоит из разноцветных квадратиков, предназначенных для корректного тестирования абсорбции и дисперсии и серой каймы.

Внутри сцены располагается кристалл (рис. 9). Оптические характеристики, которого соответствуют монокристаллу кальцита. Главные коэффициенты преломления кальцита для каждой частоты волны обычно вычисляются по формуле Селлмейера, коэффициенты для которой могут быть, например, здесь⁸. На правом изображении на рис. 10 пунктирной стрелкой показано направление оптической оси (она соединяет два тупых телесных угла кристалла). Если тестируемый кристалл не обладает естественной огранкой кальцита, то направление оптической оси должно быть предварительно определено каким-либо способом. Геометрия виртуального кристалла совпадает с геометрией реального кристалла.

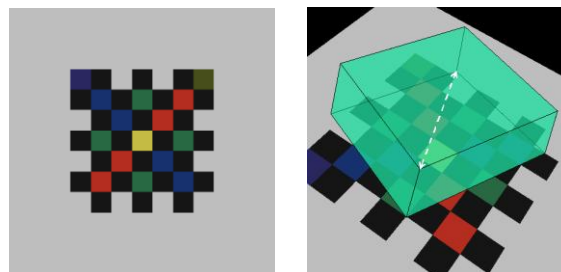


Рис. 10. RGB-текстура, наложенная на нижнюю стенку бокса (слева). Вид сцены из камеры до рендеринга

Нижняя грань кристалла расположена параллельно нижней стенке бокса, на минимальном расстоянии, при котором не возникает ошибок вычисления пересечения луча с нижней гранью кристалла и нижней стенки бокса. Эта величина определяется точностью геометрических вычислений, используемых в алгоритме. Положения реального и виртуального кристаллов совпадают.

4.3.2 Реальная сцена

На рис. 11 приведена схема реальной сцены. Сцена включает в себя три грани бокса, источник света и кристалл кальцита. На нижней грани бокса расположена текстура, напечатанная на листе белой бумаги, белые зоны на схеме на гранях бокса являются чёрными. Они изготовлены из чёрной бумаги и предварительно покрашены чёрной гуашью. Дополнительное чернение позволило существенно понизить альбедо бумаги, которое при скользящем падении света (на вертикальные грани бокса) оказывается достаточно высоким. Размеры выражены в миллиметрах. Ширина каймы текстуры равна ширине трёх цветных клеток.

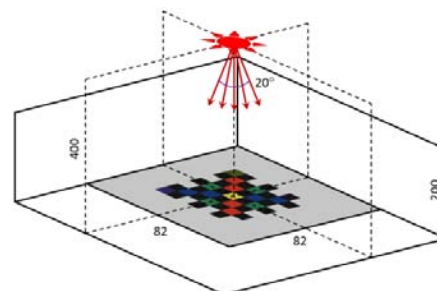


Рис. 11. Схема реальной сцены

⁷ <http://jm-derochette.be/conoscopy.htm>

⁸ <http://www.redoptronics.com/Calcite-crystal.html>

Источник света выполнен в виде конусообразной лампы с углом расхождения порядка 20 градусов. Он расположен над центром текстуры на высоте около 40 см и светит вертикально вниз (рис. 11).

Кристалл кальцита лежит на нижней грани бокса и выровнен горизонтально относительно клетчатой текстуры в соответствии с положением кристалла в виртуальной сцене.



Рис. 12. Фотография реальной сцены

Камера расположена рядом со сценой и зафиксирована на штативе. Камера направлена на сцену со стороны отсутствующих граней бокса. На рис. 12 приведена фотография сцены и расположения камеры относительно неё.

4.3.3 Сопоставление спектров источников

Источником света в виртуальной сцене является стенка бокса, с наложенной на него RGB-текстурой. В реальной сцене эту роль выполняет направленный источник света расположенный над сценой (рис. 11). Свет от источника проходит через кристалл и падает на бумагу, с распечатанной на ней клетчатой текстурой. Если бумага достаточно матовая, то можно считать, что при падении на неё свет полностью деполаризуется и отражается равномерно во всех направлениях, т.е. выступает **вторичным** источником.

В процессе рендеринга виртуальной сцены RGB-текстура преобразуется в спектральную с помощью обратного преобразования CIE. Такое преобразование не является однозначным, т.е. одинаковым RGB-цветам может соответствовать несколько различных спектров, которые называются метамерами. Спектр излучения реальной текстуры (как вторичного источника) определяется спектром излучения источника, спектром отражения красителя и спектром отражения бумаги.

Другими факторами, влияющими на RGB-цвета сфотографированной реальной текстуры, являются искажение цветов при печати текстуры, а также различные характеристики спектральной чувствительности матрицы фотоаппарата и прозрачности используемого объектива.

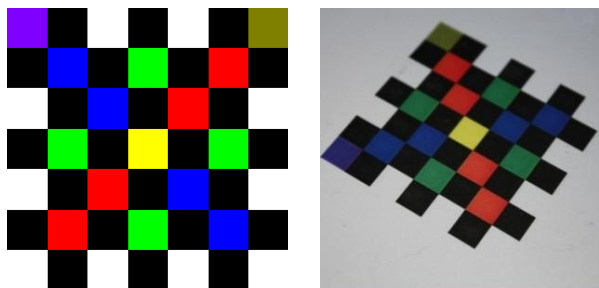


Рис. 13. Текстура, использованная для печати, и сфотографированное изображение отпечатанной текстуры

Все перечисленные факторы приводят к тому, что спектры излучения виртуальной и реальной текстуры не совпадают, и RGB-цвета виртуальной и сфотографированной текстуры различаются (рис. 13).

Для того чтобы частично устранить различия между спектрами излучения виртуальной и реальной текстуры необходимо сфотографировать реальную текстуру в сцене без кристалла, а затем по полученным RGB-цветам на фотографии изменить цвета виртуальной текстуры. Фотографирование необходимо производить с того же ракурса и при тех же настройках съёмки, при которых делается фотография сцены для сравнения. Для устранения шумов на фотографии и усреднения цвета внутри каждой клетки необходимо немного расфокусировать изображение перед съёмкой. В результате такой коррекции RGB-цвета непокрытой кристаллом части текстуры на рассчитанном изображении и фотографии будут практически идентичны (рис. 17, ниже).

Отметим, что фотографирование должно производиться при отсутствии других внешних источников освещения, т.к. реальная сцена содержит только три стенки бокса из шести.

4.3.4 Параметры съёмки и расчёта изображения

При расчёте изображения виртуальной сцены в качестве камеры используется камера-обскура (рис. 14, слева). Характеристики камеры-обскуры определяются шириной sw , высотой sh и длиной zn , а также диаметром входного отверстия R . Разрешение изображения определяется размером матрицы $M \times N$, расположенной на задней стенке бокса. Отношение M к N равно отношению sw к sh . Диаметр R равен нулю для алгоритмов, основанных на обратной лучевой трассировке, и ненулевой для алгоритмов, основанных на прямой лучевой трассировке. Фокусное расстояние такой камеры равно zn , а диафрагменное число D равно zn / R .

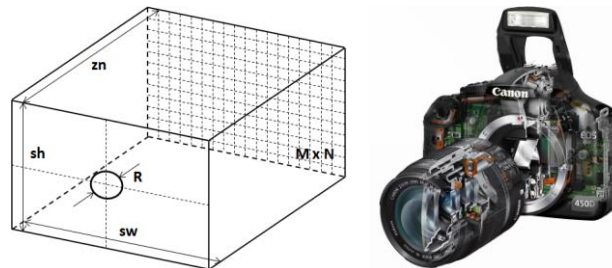


Рис. 14. Схема виртуальной сцены (слева) и устройство реальной камеры

В качестве реальной камеры наиболее удобно использовать зеркальную цифровую фотокамеру, т.к. она позволяет выставить все настройки съёмки вручную. Справа на рис. 14 видно, что устройство реальной фотокамеры намного сложнее, в частности, она содержит сложную систему линз, предназначенную, однако, для того убрать различные aberrации. При съёмке необходимо выставить диафрагменное число равное D и фокусное расстояние zn .

Камера фиксируется на штативе. Чувствительность камеры выставляется минимальной для исключения возникновения шумов на изображении. Если R виртуальной камеры равно нулю, то на реальной камере необходимо максимально уменьшить размер диафрагмы, т.е. выставить максимально возможно диафрагменное число.

Так как спектр, используемого источника света может быть нестандартным, необходимо откалибровать баланс белого по расфокусированной фотографии белого листа бумаги без отпечатанной на нём текстуры, в свете этого источника света.

Для совмещения ракурсов виртуальной и реальной камер удобнее подбирать ракурс реальной камеры, т.к. фотографирование происходит существенно быстрее расчёта изображения. Контроль правильности совмещения ракурсов съёмки осуществляется визуально. Для удобства контроля можно, если это позволяет камера, отображать на экране монитора изображение с камеры в режиме реального времени. Ракурсы совмещаются достаточно легко, если ориентироваться на положение не лежащих на бумаге вершин кристалла относительно чёрных граней бокса в плоскости фотографии.

5. АПРОБАЦИЯ ПОДХОДА

Предложенный метод верификации алгоритмов фотореалистического рендеринга кристаллов был опробован на авторском алгоритме. На первом этапе алгоритм был отлажен на системе элементарных тестов. Процесс такой «специализированной» отладки позволил выявить и устранить множество ошибок, а также ускорил отладку алгоритма в процессе разработки по сравнению с «обычной» отладкой.

Вторым этапом проверки корректности работы алгоритма стал рендеринг тестовых сцен и визуальное сравнение результатов рендеринга с тестовыми фотографиями. Результаты сравнения приведены ниже.

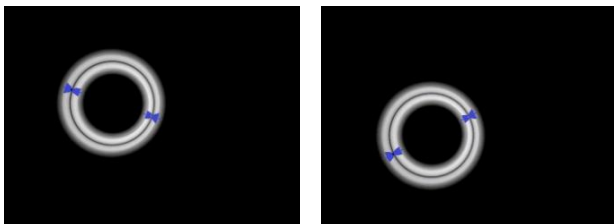


Рис. 15. Рассчитанные изображения внутренней конической рефракции при различных положениях второй бинормали

На рис. 15 приведены рассчитанные изображения внутренней конической рефракции. На изображениях отчётливо видны светлые кольца, образованные двойным преломлением, и тёмные кольца, образованные конической рефракцией. Неперпендикулярная к поверхности бинормаль двусного кристалла в одной сцене повернута относительно перпендикулярной бинормали в другой сцене на 30 градусов. В результате двойные кольца на изображениях, также повернуты на 30 градусов относительно друг друга и центров изображений.

На рисунках на кольцах отмечены диаметрально противоположные места, где изображение рассчитано некорректно, а именно, те места, где сумма интенсивностей двух порождённых лучей больше интенсивности падающего. Мы проанализировали возникшую ошибку и пришли к выводу, что её причиной является почти вертикальное падение луча на поверхность кристаллической пластинки (рис. 7, справа). В результате направления лучей вычисляются верно, а при вычислении их интенсивностей и состояний поляризации возникают ошибки точности вычислений.

Таким образом, авторам и потребителям алгоритмов фотореалистического рендеринга двусных кристаллов следует проверять и корректность расчёта этого случая. Это также говорит о том, что зачастую классические формулы не работают корректно в некоторых особых случаях, а значит, авторы алгоритмов должны находить такие случаи и по возможности устранять или хотя бы документировать их.

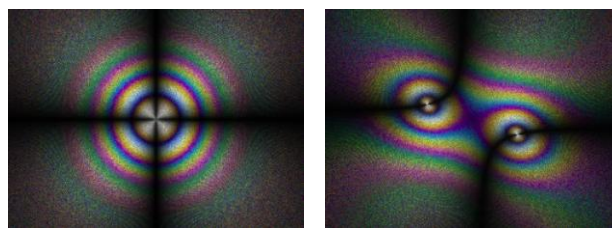


Рис. 16. Рассчитанные коноскопические изображения одноосного кристалла (слева) и двусного кристалла

На рис. 16 приведены рассчитанные коноскопические изображения для одноосного кристалла (кальцит) и двусного кристалла (гипотетический кристалл с главными коэффициентами преломления: 1.3, 1.4, 1.5). Толщина шлифа – 60 мкм. Изображения очень похожи на фотографии на рис. 2.

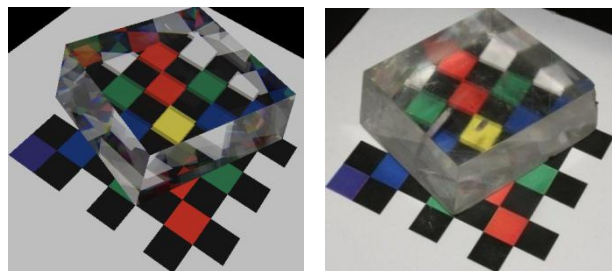


Рис. 17. Сравнение рассчитанного изображения кристалла кальцита (слева) с фотографией (справа)

На рис. 17 приведено рассчитанное изображение вышеописанной виртуальной трёхмерной сцены и фотография реальной сцены. Для рассчитанного изображения: оригинальное разрешение 480 на 435, глубина трассировки – 15, число сэмплов в спектре – 21, время расчёта 14.5 часов на 8 ядрах Intel Xeon X7350 2.93 ГГц. Для расчёта использовался алгоритм, основанный на **обратной** лучевой трассировке, т.е. R равно 0. Для фотографии: фотоаппарат – Canon 450D, объектив – EF-S18-55 мм F:3.5-5.6 IS, чувствительность – ISO100, выдержка – 1/25 секунды, диафрагменное число – 32 (максимально возможное для этого объектива при выбранном фокусном расстоянии), режим ручной фокусировки, ручной баланс белого. Отчётливо видно совпадение удвоенной текстуры видимое, через верхнюю грань: совпадает направление и степень двойения изображения, а также совпадает изображение текстуры непокрытой кристаллом.

К сожалению, части изображения, соответствующие боковым граням кристалла, не совпадают. Возможными, причинами этого являются:

- 1) Реальный кристалл является не монокристаллом, а сростком из трёх кристаллов, хотя их направления оптических осей достаточно близки. При этом на гранях срастания происходят дополнительные преломления, не учитываемые при расчёте изображения виртуальной сцены.
- 2) Хотя при фотографировании выбран минимальный размер диафрагмы, он всё же отличается от нуля, а следовательно, фотография имеет конечную глубину резкости. При многократном отражении света внутри кристалла эта глубина достигается и мозаичное цветное изображение, видимое через боковые стенки на рассчитанном изображении, сливается в равномерный серый цвет на фотографии.
- 3) Небольшое отличие ракурсов камер.

6. ЗАКЛЮЧЕНИЕ

В данной работе предложен метод верификации алгоритмов фотореалистического рендеринга кристаллов. Метод основан на проверке ряда численных и визуальных тестов. Система тестов была апробирована на алгоритме, разработанном авторами. Проверка тестов помогла выявить ряд существенных ошибок и частично исправить их.

Данная система тестов не претендует на окончательную завершенность. Этой работой мы хотели бы инициировать коллективное наполнение сайта, в котором участвовали бы разработчики аналогичных алгоритмов и программ, а также другие исследователи, которые могли бы представить дополнительные тесты. Очевидно, что очень важны тесты, результаты которых можно вывести теоретически, но исключительную ценность имеют спецификации сцен, особенно при наличии фотографий. Несомненно, что детальное описание отдельного теста вполне по силам одному коллективу, как в Cornell Box, однако, хотелось бы иметь более мощный набор, как это сделано для задач вычислительной математики. Начальный вид сайта можно посмотреть по адресу⁹.

На сайте для всех визуальных тестов даны конфигурационные файлы (и описание их формата), содержащие все оптические и геометрические характеристики приведенных сцен, все параметры необходимые для их расчёта и рассчитанные изображения сцен, а также соответствующие им фотографии, если они приводятся в тексте. Таким образом, авторы алгоритмов смогут полностью воспроизвести сцены, приводимые в статье, и сравнить изображения, размещённые на сайте, с изображениями, рассчитанными с помощью их алгоритмов, и фотографиями.

7. БЛАГОДАРНОСТИ

Работа выполнена при поддержке РФФИ по гранту № 09-07-00237а.

8. БИБЛИОГРАФИЯ

- [1] Greenberg D. et al. *Modelling the Interaction of Light Between Diffuse Surfaces* // *Computer Graphics, 1984 - Volume 18, Number, p. 213-222.*
- [2] Guy S., Soler C. *Fast and Physically-Based Rendering of Gemstones.* // *Comp. Graphics (SIGGRAPH '03 Proceedings), ACM Press.*
- [3] 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.*
- [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] 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).*
- [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] Farnsworth M., Erbacher R. F. *Global Illumination: Efficient Renderer Design and Architecture* // *Proceedings of the International Conference on Geometric Modeling, Visualization & Graphics 2005, pp. 1691-1695.*
- [8] Дебелов В.А., Саттаров М.А. *Модели и алгоритмы для фотореалистической визуализации сцен с кристаллами.* // *Тр. 15-й международной конференции по компьютерной графике и зрению ГрафиКон-2005, – Новосибирск, 2005. С. 348-354.*
- [9] Борн М., Вольф Э. *Основы Оптики.* Москва, “Наука”, 1973.
- [10] Фёдоров Ф.И., Филиппов В.В. *Отражение и преломление света прозрачными кристаллами.* Минск, “Наука и техника”, 1976.
- [11] Шубников М.В. *Оптическая кристаллография.* М.: Изд. АН СССР –1950. – 274 с.
- [12] Дебелов В.А., Рубцова А.Ю., Смирнов С.З. *Компьютерная модель петрографического микроскопа* // *Труды XVI международной конфер. по компьютерной графике и ее приложениям ГрафиКон-2006 – 2006. – С. 293–297.*
- [13] Козлов Д.С. *Алгоритм рендеринга по методу Монте-Карло петрографических шлифов одноосных кристаллов* // *Труды XIX Международной конференции по компьютерной графике и зрению ГрафиКон-2009, – М., 2009, – С. 368-371.*

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.

E-mail: kozlov@oapmg.sccc.ru.

Verification of algorithms of photorealistic rendering of crystals

In this paper, an approach of verification of algorithms of photorealistic rendering of isotropic and anisotropic crystals is suggested, which is similar to approach successfully used in computational mathematics, e.g. site Matrix Market. A classical example for global illumination area is Cornell Box project [1]. The paper exposes an extensible set of tests based on well-known laws of light rays propagation at interfaces between media, specifications of test scenes, photos of the scenes (if it is possible), and images of the scenes calculated by various rendering algorithms.

Note that availability of results of work of various algorithms (such as in the case of Cornell Box) allows an application programmer to select one which is the most suitable for his purposes.

Keywords: *Photorealistic rendering, crystals, verification, birefringence, polarization, optical dispersion, conical refraction, interference, rendering algorithm testing.*

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 Novosibirsk State University. His contact email is debelov@oapmg.sccc.ru.

Dmitry S. Kozlov is a Ph.D. student of the Novosibirsk State University. His contact email is kozlov@oapmg.sccc.ru.

⁹ http://oapmg.sccc.ru/crystal_tests

Анализ алгоритмов оптимизации времени отображения электронных карт в формате HP-GL.

Ю.Л. Кетков
НИИ ПМК ННГУ
Нижний Новгород, Россия
ket@unn.ru

З.А. Матвеев
Intel Corporation
Нижний Новгород, Россия
zakhar.a.matveev@intel.com

Abstract

The visualization performance becomes critical when checking the digital maps completeness and quality; especially since typical map size is about 10-100 Mb of vector graphical data. In this paper two base performance improvement methods proposed. They are: data pre-processing technique and spatial indices-like data structure usage.

Generalization, hierarchical indexing, spatial selection criteria approaches; local offset optimization, special polyline 2-dimensional scissoring and task parallelism-based modification of algorithms introduced and analyzed as well.

Experimental results illustrate computational efficiency of proposed approaches in comparison with other visualization software.

Keywords: *Visualization, Indices, Digital map, Cartography, Scissoring.*

1. ВВЕДЕНИЕ

Бурное развитие цифровой картографии за последние годы привело к появлению большого числа ГИС различного назначения, в том числе автоматических и автоматизированных картографических систем (АКС). Одной из задач, решаемых с помощью профессиональных АКС, является осуществление контроля полноты и качества цифровых моделей топографических карт (ЦТК).

В большинстве случаев исходным материалом для создания ЦТК являются стандартные тиражные оттиски, при подготовке которых существует довольно много отступлений от регулярных правил. Например, допускается изменять характеристики шрифтов, отступать от типовых размеров дискретных знаков, варьировать шаг расположения заполнителей линейных знаков, выносить надписи в местах большой насыщенности документа и т.п. Кроме того в процессе оцифровки, автоматического распознавания карты возникают самые разные ошибки, связанные с погрешностью позиционирования, деформацией исходных документов, проблемами классификации картографических объектов и их пространственно-логических связей.

В результате электронная версия картографического документа может существенно отличаться от оригинала. Поэтому как на этапе первоначального наполнения, так и при

оперативном обновлении содержания, ЦТК обязательно должны контролироваться с помощью автоматических средств или путем визуального просмотра. Это особенно необходимо в случае подготовки издательских оригиналов цифровых карт, т.к. твердые (бумажные) копии еще не потеряли своей актуальности.

Требования, предъявляемые руководствами по изданию карт, достаточно высоки и для соблюдения установленных правил редактору-картографу приходится тщательно исследовать каждый фрагмент изображения. С учетом размеров листов карт (габариты морских карт могут достигать порядка 1000 мм) и габаритов просматриваемых объектов (минимальная высота некоторых надписей составляет 1–1.2 мм) на экране дисплея приходится отображать фрагменты электронной карты с увеличением в 4-8 раз. Когда размер файла с графическим изображением электронной карты достигает нескольких десятков и сотен Мбайт, быстродействие системы отображения становится решающим фактором, напрямую влияющим на эффективность внедрения АКС. Однако оперативность многих современных графических систем, работающих как с растровыми, так и с векторными изображениями такого размера, оставляет желать лучшего.

Просмотр содержимого графического файла, масштабирование различных его фрагментов и навигация по смежным фрагментам – операции массовые, требующие активного взаимодействия с пользователем. Для выполнения этих операций обычно используется специальная программа просмотра (визуализатор) графических файлов. В статье рассматриваются алгоритмы и методы, позволяющие повысить эффективность визуализаторов используемых в составе АКС для просмотра электронных карт.

2. ПРЕДВАРИТЕЛЬНЫЙ АНАЛИЗ

Экспериментальный анализ и апробация алгоритмов оптимизации выполнялись в процессе проектирования и разработки визуализатора векторных ЦТК в формате HP-GL/2 [7] (именно в этом формате формируются электронные карты современными версиями АКС, разрабатываемыми на базе НИИ ПМК Нижегородского государственного университета).

Формат HP-GL/2 (Hewlett-Packard Graphics Language) является на сегодняшний день de facto промышленным стандартом для производителей серийных большеформатных

плоттеров (применяемых для воспроизведения цветных оттисков) и специализированных прецизионных фотоплоттеров (используемых при подготовке издательских оригиналов и заливочных шаблонов).

Образ электронной карты в формате HP-GL представляет собой линейную графическую программу (без циклов и ветвлений), ориентированную на управление плоттером и содержащую достаточно длинную цепочку команд вида:

CD [p1,p2,...];

где CD – двухсимвольный код графической команды, а p1,p2,... – обязательные или необязательные числовые параметры, записанные в символьном формате. В качестве параметров могут выступать целочисленные и/или вещественные координаты, радиусы окружностей, углы, характеристики пишущего узла и др. Таким образом, электронная карта в формате HP-GL/2 это – обычный текстовый файл, каждая строка которого представлена графической командой.

Наиболее многочисленную группу в файле представляют команды воспроизведения различных графических примитивов – отрезков прямых той или иной толщины, залитых и не залитых окружностей, дуг, круговых и эллиптических секторов, фрагментов кубических парабол. Особенно много таких примитивов порождается при обработке надписей. Аппаратура плоттеров, как правило, не содержит графические описания кириллических шрифтов, поэтому каждую русскую букву приходится формировать из достаточно большого количества графических примитивов. При этом надо учитывать, что дизайн российских карт, отсчитывающий свою историю от времен правления Петра I, предполагает использование только для топографических карт 26 различных символьных наборов.

Лист топографической карты со средней нагрузкой может содержать порядка 10^6 - 10^7 графических команд. Получив от пользователя заказ на отображение соответствующего фрагмента листа, программа визуализации должна последовательно обработать все команды в графическом файле, и воспроизвести их на экране с соответствующим масштабным коэффициентом. На полном листе карты таких фрагментов от 100 до 1000, и к некоторым из них приходится обращаться не по одному разу. Поэтому лобовой вариант интерпретации команд графического файла (с повторным синтаксическим анализом и масштабированием для каждого очередного фрагмента) представляется крайне неэффективным.

Сформулируем ряд проблем, решение которых позволило бы повысить эффективность системы отображения при обработке графических файлов большого размера.

Внутренний формат хранения данных.

Необходимо ли при многократных просмотрах фрагментов одной и той же карты каждый раз заниматься синтаксическим анализом текстового файла и преобразованием графических команд из символьного представления в соответствующие машинные форматы? Очевидно, что это не целесообразно: такую интерпретацию следует проделать один раз и при просмотре новых фрагментов нужно обрабатывать информацию, полученную при первичном анализе. Реализация такого подхода связана с созданием подходящего

формата внутреннего представления, исключающего повторение уже проделанной работы.

Следует отметить, что введение подобного внутреннего формата позволяет заблаговременно (на этапе открытия файла) вычислить различные дополнительные характеристики объектов (площадь, взаимосвязи с другими объектами и пр.), которые при работе с фрагментом электронной карты позволяют ускорить его обработку.

Отбор данных для последующей визуализации.

Есть ли необходимость воспроизводить всё множество графических команд электронной карты при просмотре её фрагмента? С одной стороны, возможно при каждом изменении коэффициента масштабирования растеризовать всю карту целиком, сохраняя её в виде промежуточного растрового изображения. Однако в этом случае возрастёт объём оперативной памяти компьютера, используемой для хранения промежуточного раstra. Кроме того, при любом изменении масштаба карты – будет выполняться генерация растрового представления всей карты заново, что повлечёт существенные временные потери.

Альтернативный подход основывается на идее динамического отбора графических команд, в зависимости от параметров запрашиваемого фрагмента (масштаб и положение на листе карты). Так, при каждом изменении масштаба можно обрабатывать только те объекты, которые пересекаются с указанным кадром (двумерное отсечение).

Допустимы и некоторые другие критерии отбора. Например, при малом коэффициенте увеличения вполне целесообразно отбрасывать те объекты, которые заведомо не будут различимы человеческим глазом (генерализация карты). Однако, непосредственное применение названных критериев не всегда достаточно эффективно.

Эффективный доступ к информации и индексирование.

Как ускорить доступ к нужной информации при отборе графических команд, отображающих объекты, принадлежащие заданной области? К решению этой проблемы ведут два пути.

Во-первых, в состав таблиц с внутренним представлением можно включить различные данные, упрощающие поиск. В роли таких данных обычно выступают некоторые характеристики графического объекта, например координаты его "точки привязки" (той точки, в которую пишущий узел переводится перед началом построения объекта) или минимальных окаймляющих прямоугольников (MBR – Minimal Bounding Rectangle).

Другой подход заключается в индексировании внутренних данных, т.е. в построении системы указателей (индексов), обеспечивающих прямой доступ к графическим данным, инцидентным окну просмотра.

Отметим, что при рассмотрении каждого из этих подходов, необходимо определиться с термином "объект". Нужно ли считать таковым объект в смысле классификатора соответствующей предметной области или абстрагироваться от такой содержательной трактовки и перейти на уровень простейших графических примитивов? В первом случае количество объектов существенно уменьшится, при этом появится тематическая зависимость. И хотя внедрение тематической зависимости является несомненным

преимуществом, - для её поддержки может потребоваться, чтобы программа подготовки графического файла перед формированием образа очередного картографического объекта заносила в HP-GL файл команду-комментарий (благо в версии HP-GL/2 такая команда с кодом СО появилась), по которой можно было автоматически определять точку привязки. Во втором случае точка привязки и другие характеристики определяются элементарно, но зато количество объектов возрастает на один-два порядка.

В целом, вопросы представления графических объектов, организации индексирования и внедрения дополнительных данных тесно взаимосвязаны между собой, воздействуя на производительность визуализатора самым существенным образом.

Ускорение отсечения ломанных

За счет чего можно ускорить процедуру отсечения графических команд, воспроизводящих объекты, не попадающие в кадр просмотра? Очевидно, что только точка привязки объекта не позволяет судить об его принадлежности к заданному окну просмотра. Картографический объект может быть линейным и достаточно протяженным или площадным с размерами, превышающими габариты окна просмотра, в которое попадают какие-то фрагменты объекта. Даже графический примитив, соответствующий длинному отрезку прямой, может начинаться за пределами окна просмотра, но, тем не менее, пересекать кадр. Существует довольно много алгоритмов отсечения, описанных в литературе, но в нашей задаче имеется достаточно важная специфика – приходится анализировать принадлежность окну просмотра не одного отрезка или его части, а множества мелких отрезков, образующих ребра ломанных или расположенных достаточно близко друг к другу (например, в случае аппроксимации контуров букв). В этом случае затраты на процедуру отсечения можно сократить примерно вдвое за счет того, что результаты анализа конца текущего отрезка можно учесть при анализе положения начала следующего отрезка.

Оптимизация просмотра смежных фрагментов.

Можно ли сэкономить время просмотра при переходе к соседнему фрагменту, граничащему с текущим по одной из четырех границ? Плавная навигация по соседним областям листа карты – один из достаточно распространенных режимов работы редактора-картографа. В простейшем случае можно разбить лист карты на фиксированное число прямоугольных фрагментов и вести просмотр любого из них. Однако это не очень удобно. Во-первых, часто требуется менять коэффициент масштабирования как в сторону увеличения, так и в сторону уменьшения. Во-вторых, скачкообразный переход к новому фрагменту не позволяет проанализировать структуру карты в районе границ смежных фрагментов. При плавной навигации возникает желание воспользоваться частью ранее полученного растрового изображения и заново воспроизвести только обновленную часть кадра. Таким образом можно примерно на 30% сократить время формирования следующего кадра и снизить неприятный эффект скачкообразного перехода к следующему изображению.

По результатам анализа перечисленных проблем были предложены различные методы и алгоритмы оптимизации, описанию которых посвящены следующие разделы статьи.

Отметим, что большинство рассматриваемых алгоритмов применимо не только к конкретному формату HP-GL/2, но и к векторным графическим форматам вообще.

3. ВНУТРЕННЯЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ДАННЫХ.

Как отмечалось выше, синтаксический анализ исходного файла имеет смысл выполнять однократно с целью создания внутреннего представления данных. Для решения этой задачи желательно использовать специальный компонент ПО, ответственный за предобработку графических данных. Одна из задач подобного компонента – распознавание символьного кода графической операции, преобразование значения параметров команды из символьного представления в целочисленный машинный формат. Вторая задача заключается в объединении смежных команд в группы, соответствующие понятию абстрактного графического объекта и формированию атрибутов этого объекта.

Понятие объекта может варьироваться от элементарного примитива, представленного единственной командой языка HP-GL до тематически зависимого картографического объекта со всеми его характеристиками. На наш взгляд, может быть предложена некоторая промежуточная модель, представленная группами следующих графических примитивов:

1. Группа установочных объектов, задающих текущие цвет, толщину и тип окончания линии, используемых при визуализации графических данных (для формата HP-GL/2 формируются командами SP).
2. Группа не залитых полигонов/ломанных (применительно к формату HP-GL/2 – речь идёт о командах PD, PU, PM, EP)
3. Группа "криволинейных" объектов типа дуга, окружность, сектор и др. Для представления объектов этой группы, не используются координаты точек контура. По этому признаку к этой группе можно также отнести объекты текстовых меток.
4. Подгруппа полигональных объектов, описывающих залитые области (для формата HP-GL/2 генерируются командами FP).
5. Подгруппа объектов координатной основы.

Группировка объектов в составе внутреннего формата осуществляется иерархически с использованием шаблона проектирования "Компоновщик" [1]. При этом, например, объекты координатной основы – являются дочерними по отношению к объектам из группы не залитых полигонов, а те в свою очередь – являются дочерними по отношению к установочным объектам.

В процессе построения описываемой структуры данных для каждого объекта (за исключением объектов нижнего уровня) заблаговременно вычисляются координаты точки привязки и MBR (окаймляющего прямоугольника – от англ. minimal bounding rectangle). При этом MBR объектов i -го уровня (узлов дерева глубины $i-1$) иерархии определяется как окаймляющий прямоугольник, включающий MBR всех дочерних объектов $i+1$ го уровня (узлов дерева глубины i).

Можно привести следующие аргументы в пользу предлагаемой модели представления данных (и их классификации):

- Использование установочных объектов на верхнем уровне иерархии позволяет оперировать с данными внутреннего формата как с картографическими слоями, облегчая генерализацию и минимизируя количество переключений палитры в процессе визуализации.
- Низкоуровневые объекты координатной основы отличаются от объектов других групп. Прежде всего, данные объекты всегда являются листовыми в древовидной структуре данных. Во-вторых, количество этих объектов очень велико. Так, например для типовых ЦТК в формате HP-GL/2 число объектов этой группы достигает 1-5 млн. единиц. Поэтому размер конкретного экземпляра структуры данных внутреннего формата определяется, в первую очередь, количеством объектов этого типа, а производительность визуализатора – определяется способом обработки этих объектов.
- Применение объектов второго типа может способствовать более эффективному использованию процедур графического API. Известно, например, что обращение к подпрограмме воспроизведения ломаной линии Windows GDI Polyline намного эффективнее, чем использование многократных вызовов функции LineTo. Поэтому агрегирование цепочек отрезков в составе одного объекта может заметно повысить эффективность визуализации (в особенности когда заведомо известно, что весь объект целиком лежит внутри текущего окна просмотра).
- Хотя криволинейные объекты и объекты-надписи могут быть представлены с помощью объектов-ломаных (криволинейные – могут быть только аппроксимированы), они требуют специальных алгоритмов индексирования и растеризации. Поэтому их целесообразно выделить в отдельную подгруппу и не сопоставлять с ними координатную основу.
- Применение шаблона “Компоновщик” упрощает иерархическую организацию данных и обход полученной древовидной структуры.

4. АЛГОРИТМЫ ИНДЕКСИРОВАНИЯ.

Предварительная обработка данных позволяет выполнять дальнейшее эффективное индексирование картографических документов при просмотре увеличенных фрагментов карты. При этом в качестве выходных данных алгоритма индексирования выступает специализированная структура данных, называемая индексом, содержащая номера тех графических объектов, которые удовлетворяют одному или нескольким критериям отбора.

Индекс строится динамически. Это означает, что для каждого нового окна просмотра индекс очищается и заполняется заново на основе выбранного критерия.

При этом внутреннее представление индекса также является иерархическим, что позволяет в частности, минимизировать число операций отсеечения объектов. В самом деле, если объект верхнего уровня не попадает в индекс (то есть лежит целиком вне окна просмотра) – отпадает необходимость в индексировании всего множества его дочерних объектов, т.к. все они также заведомо не попадают в индекс. Аналогично, если объект верхнего уровня, лежит целиком внутри окна отсеечения, нет необходимости применять критерий отсеечения для всех его дочерних объектов, т.к. очевидно, что каждый из них попадает в индекс (см. рис. 1).

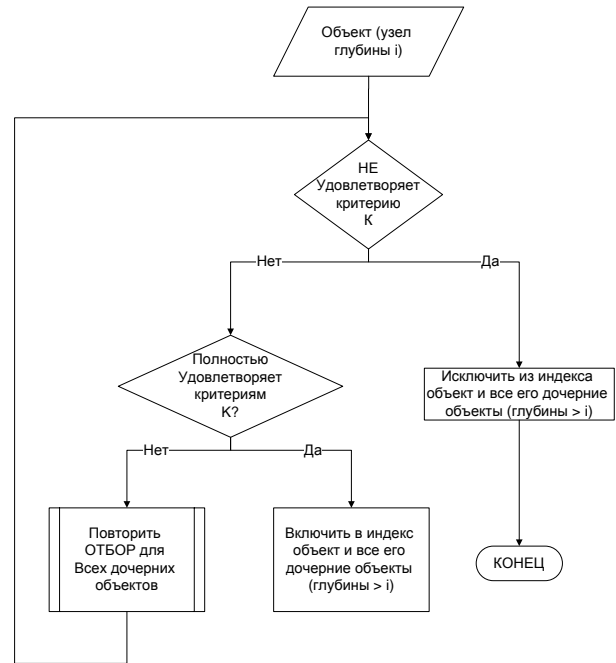


Рис. 1. Блок-схема иерархической индексации.

Для объектов, расположенных на первых двух уровнях иерархии основным критерием отбора является пересечение MBR объекта с текущим окном просмотра (“грубый критерий” отбора).

В случае если MBR объекта второго уровня лежит на границе окна просмотра – выполняется уточняющий шаг: двумерное отсеечение дочерних объектов третьего уровня (отрезков). Для определения принадлежности очередной вершины отрезка окну просмотра используется информация о предыдущей точке, полученная на предыдущем шаге согласно упомянутому выше алгоритму отсеечения ломаной.

Наряду с анализом взаимного расположения объектов и окна отсеечения, при построении индекса учитываются относительные размеры объектов. В случае если при данном значении коэффициента масштабирования объект является неразличимым на экране дисплея (его площадь или периметр меньше некоторой величины, зависящей в свою очередь от коэффициента масштабирования) – он также не заносится в индекс. В данном случае уже используется количественный критерий генерализации.

Таким образом, индексирование выполняется на основе трёх критериев отбора информации: грубого пространственного, точного пространственного и количественного генерализационного.

На практике, в целях оптимизации, возможны некоторые отклонения от описанной схемы. Например, в случае если пользователь выполняет два последовательных запроса на отображение фрагментов с увеличением n и $2*n$ соответственно (без сдвига окна просмотра), при переходе от одного фрагмента к другому индекс не очищается, т.к. индекс, построенный при первом запросе, уже содержит все необходимые идентификаторы объектов - и остаётся лишь отбросить те, из них, которые не принадлежат новому фрагменту. Другая особенность используемого подхода заключается в том, что для залитых областей на практике не выполняется уточняющее индексирование, т.к. двумерное отсечение залитой области само по себе является трудоёмкой задачей.

5. ДОПОЛНИТЕЛЬНЫЕ СРЕДСТВА ОПТИМИЗАЦИИ ВРЕМЕНИ ОТОБРАЖЕНИЯ.

5.1 Оптимизация локальной навигации.

Не все разработанные алгоритмы оптимизации просмотра укладываются в схему предобработки и последующего индексирования карты. Так, например, существенный прирост производительности был достигнут за счёт реализации алгоритма локальной навигации (рис. 2).

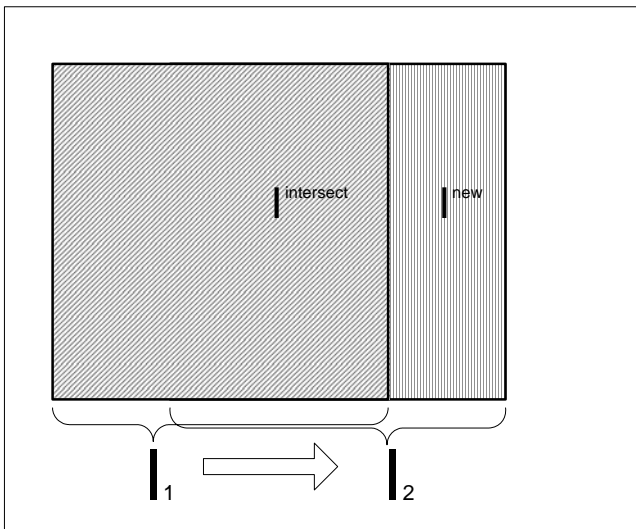


Рис. 2. Алгоритм локальной навигации.

Этот алгоритм применяется при работе в режимах панорамирования и фиксированного смещения без изменения масштаба. Оба названных режима подразумевают последовательное перемещение от одного фрагмента изображения (с окаймляющим прямоугольником I^1) к другому (I^2 соответственно) с тем, что часть предыдущего фрагмента видна на мониторе и после перемещения (т.е. фактически $I^{\text{intersect}} = I^1 \cap I^2 \neq \emptyset$). Причём если в режиме панорамирования площадь $I^{\text{intersect}}$ может быть очень мала (в случае, если пользователь интенсивно "перетаскивает" изображение), то в режиме фиксированного смещения площадь $I^{\text{intersect}}$ гарантированно составляет 50-75% от площади исходного I^1 .

В связи с этим возникает идея дальнейшего сужения множества отображаемых объектов; в самом деле, если часть требуемого растрового изображения ($I^{\text{intersect}}$) уже была сгенерирована при предыдущем запросе пользователя – нет

необходимости генерировать её заново; достаточно лишь растеризовать *новый* фрагмент ($I^{\text{new}} = I^2 \setminus I^{\text{intersect}}$, $I^{\text{new}} \subset I^2$, см. также рис. 2).

5.2 Параллельная модификация алгоритма.

В связи с широким распространением ПК на базе многоядерных процессоров особенный интерес представляют параллельные модификации алгоритмов визуализации.

В ходе анализа и апробации различных способов декомпозиции алгоритмов индексирования и растеризации, автором была реализована схема функциональной декомпозиции (конвейерного типа), в результате чего удалось снизить время отображения на 20-25%.

Идея предложенного способа декомпозиции заключается в том, что в то время, как i -й слой (установочный объект) индексируется в первом потоке, – второй поток может выполнять преобразование координат для $i-1$ -го слоя, а третий поток – растеризацию $i-2$ -го слоя. Графически конвейерная схема проиллюстрирована на рис. 3. В роли общих данных для трёх потоков выступает индекс (взаимодействие 1 и 2 потоков) и дерево внутреннего формата (взаимодействие 2 и 3 потоков).

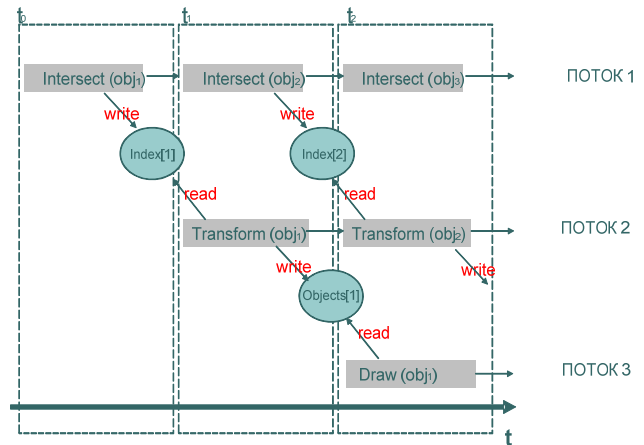


Рис. 3. Графическая иллюстрация алгоритма функциональной "конвейерной" декомпозиции (Intersect – подпрограмма индексирования, Transform – подпрограмма преобразования координат, Draw – подпрограмма растеризации).

Очевидным недостатком предложенной схемы является слабая масштабируемость алгоритма. В тоже время, в отличие от различных вариантов декомпозиции по данным, "конвейерный" алгоритм эффективно функционирует независимо от целевой аппаратной платформы и возможностей графического API.

В завершение упомянем, что на основе анализа узких мест программы были применены различные методы низкоуровневой оптимизации: использование внутреннего менеджера памяти; кэширование идентификаторов команд; избирательное применение целочисленной арифметики в процессе изменении масштаба карты. Каждый из этих методов оказался вполне адекватным поставленной задаче, обеспечивая прирост производительности на той или иной стадии обработки графической информации.

6. ЗАКЛЮЧЕНИЕ.

В ходе внедрения алгоритмов оптимизации, было выполнено несколько вычислительных экспериментов, позволивших проанализировать как вклад каждого алгоритма по отдельности, так и совокупный вклад всех рассмотренных алгоритмов оптимизации.

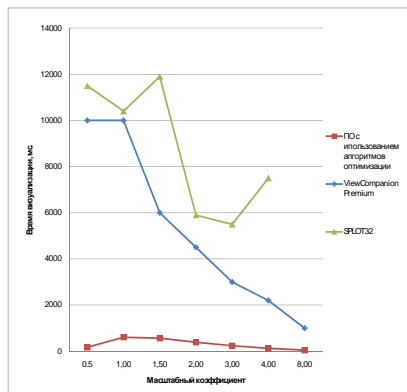


Рис. 4. Время отклика визуализатора в процессе интерактивной навигации (математическое ожидание времени отклика в с.).¹

В роли экспериментальных данных использовались типовые топографические цифровые и морские навигационные карты в формате HP-GL/2 (количество объектов координатной основы в пределах 1-2 млн. единиц на карту).

Показатели, продемонстрированные ПО, разработанным с использованием рассмотренных алгоритмов [5], оказались значительно выше показателей сторонних коммерческих визуализаторов HP-GL/2 [6], [8] (рис. 4).

При этом по результатам сравнительного анализа рассмотренных в статье алгоритмов были выявлены следующие закономерности:

- Индексирование с использованием грубого пространственного критерия (окаймляющих прямоугольников) позволяет существенно ускорить визуализацию при коэффициенте масштабирования более 2 за счёт быстрого отсека протяжённых объектов.
- В свою очередь, алгоритм генерализации и иерархической индексации обеспечивают прирост производительности при малых коэффициентах масштабирования (одно- – трёх- кратное увеличение

листа ЦТК), в частности за счёт исключения небольших пространственных объектов.

- При использовании всей совокупности алгоритмов индексирования и генерализации – время, затрачиваемое на визуализацию фрагмента, монотонно убывает с ростом коэффициента увеличения.
- Применение внутреннего формата позволяет существенно повысить производительность визуализации на этапе интерактивной навигации по листу ЦТК (время визуализации уменьшается до 10 раз). В тоже время внедрение внутреннего формата ведёт к увеличению времени первоначальной загрузки исходного документа и требует большего объёма доступной оперативной памяти (порядка 80-100Мб для типовых ЦТК в формате HP-GL/2).
- Функциональная декомпозиция визуализации позволяет дополнительно повысить эффективность алгоритма визуализации на 10% – 25% практически при всех масштабных коэффициентах. Однако при 8-кратном увеличении полученный выигрыш начинает уравниваться накладными расходами, связанными с организацией многопоточных вычислений.

В целом, применение изложенных подходов позволило достичь высокой степени эффективности при интерактивном просмотре ЦТК.

7. ЛИТЕРАТУРА

- [1] Гамма, Э. Приемы объектно-ориентированного проектирования: Паттерны проектирования [Текст] : пер. с англ./ Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. – СПб : Питер, 2004. – 366 с.
- [2] Ласло, М. Вычислительная геометрия и компьютерная графика на C++ [Текст] / М. Ласло. – М. : БИНОМ, 1997. – 304 с.
- [3] Фень, Ю. Программирование графики для Windows [Текст] / Ю. Фень. – СПб. : Питер, 2002. – 1072 с.
- [4] Gutting, R.H. An introduction to spatial database systems [Text] / R.H. Gutting. – VLDB Journal. – 1994. – Vol 3. – No 4. – P. 357-399.
- [5] Matveev, Z. Indices Organization as Visualization System Performance Improvements Means [Text] / Z. Matveev // 9-th International Conference on Pattern Recognition and Image Analysis: New Information Technologies PRIA-9-2008 : conference proceedings. – Nizhni Novgorod, 2008. – vol.2. – P. 15-17.
- [6] Novy A. SPlot V5.20 for Windows. User Guide [Electronic resource] / A. Novy. – Режим доступа: <http://www.swplot.com>, свободный – Загл. с экрана.
- [7] The HP-GL/2 and HP RTL reference guide: a handbook for program developers [Text] / Hewlett-Packard. – MA : Addison-Wesley, 1997. – 3rd edition. – 544 p.
- [8] View Companion. Premium Edition. Quick reference [Electronic resource] / Software Companions. – Режим доступа:

¹ Использовалась тестовая конфигурация клиентского ПК на базе процессора Intel Core Duo, 1024 Mb SDRAM ОЗУ, оборудованного графическим видеоакселератором Intel GMA 950. В качестве тестовых данных была выбрана навигационная карта побережья Северного моря, размер файла ЦТК – 28 Мб.

<http://www.softwarecompanions.com/viewpremium.html>,
свободный. – Загл. с экрана.

Об авторах

Кетков Юлий Лазаревич – д.т.н., профессор кафедры МО
ЭВМ факультета ВМК Нижегородского государственного
университета им. Н.И. Лобачевского (ННГУ), зав. отделом
НИИ Прикладной Математики и Кибернетики при ННГУ.

E-mail: ket@unn.ru

Захар Александрович Матвеев – сотрудник нижегородского
отделения Intel Corporation (разработчик программного
обеспечения), выпускник аспирантуры ННГУ.

E-mail: zakhar.a.matveev@intel.com

Моделирование каустик в реальном времени

Денис Боголепов, Дмитрий Сопин, Вадим Турлапов
 Нижегородский государственный университет им. Лобачевского
 Факультет вычислительной математики и кибернетики, Нижний Новгород, Россия
 {denisbogol, sopindm, vadim.turlapov}@gmail.com

АННОТАЦИЯ

Целью данной работы является адаптация метода фотонных карт для моделирования каустик в реальном времени. В работе предлагается быстрый алгоритм, который целиком выполняется на ГПУ и реализован на основе инструментов OpenGL и OpenCL. Для построения фотонной карты и визуализации сцены используются шейдеры OpenGL. Для увеличения производительности визуализации разработана основанная на вокселях ускоряющая структура, которая конструируется на ГПУ средствами OpenCL. Приводятся оценки производительности.

Ключевые слова: Интерактивная визуализация на ГПУ, трассировка лучей, метод фотонных карт, моделирование каустик, GPGPU, OpenGL, GLSL, OpenCL.

1. ВВЕДЕНИЕ

Эффекты *глобального освещения* играют важную роль в задачах синтеза реалистичных изображений. К базовым эффектам можно отнести *тени*, *многократные отражения* и *преломления*. Для их моделирования в большинстве случаев применяется алгоритм *трассировки лучей* [1]. Расширенные эффекты, связанные с *диффузным отражением* света и *каустиками*, требуют более сложных алгоритмов. К числу наиболее распространенных относится метод *фотонных карт*, который обеспечивает практическое и вычислительно эффективное решение для задач глобального освещения [2]. Алгоритмы глобального освещения и метод фотонных карт в частности традиционно полагаются на сложные программные реализации и статическую визуализацию. Тем не менее, появление производительной программируемой графической аппаратуры позволило сделать большой шаг в направлении интерактивного применения данных методов.

Одна из первых успешных реализаций метода фотонных карт на графическом процессоре описана в работе [3]. Авторы отказались от использования традиционного *kd*-дерева для ускорения доступа к фотонной карте. Вместо этого, в работе используется регулярная сетка, и предлагаются два способа распределения фотонов по вокселям. Кроме того, авторами был предложен метод поиска *k* ближайших фотонов сетки для заданной точки соударения. Данный метод оказался весьма ресурсоемким и занимал порядка 90% от времени визуализации. Безусловно, используемая на тот момент аппаратура класса NVIDIA GeForce FX 5900 Ultra не могла обеспечить интерактивной визуализации даже для самых простых сцен. Например, сцена с одним прозрачным объектом в окне 512×512 точек для 65×10^3 фотонов обрабатывалась более одной минуты. Тем не менее, данный подход характеризуется высокой трудоемкостью и вряд ли обеспечит реальное время для сцен средней сложности даже на современном оборудовании. Хотя производительность

уступала существующим реализациям для центрального процессора, работа была крайне интересной и послужила основой для дальнейших улучшений. К их числу относятся работы [4]–[5], в которых авторы исследовали различные ускоряющие структуры, включая *kd*-деревья и иерархию ограничивающих объемов. Однако применение современного оборудования не позволило достичь реального времени даже на простых тестовых сценах. Например, в работе [5] на простых сценах в окне 512×512 точек для 128×10^3 фотонов получена производительность порядка 4–6 к/с на ГПУ NVIDIA GeForce GTX 280.

Целью настоящей работы является реализация *упрощенного* метода фотонных карт для исполнения на графическом процессоре в *реальном* масштабе времени. В данной работе показано, каким образом можно дополнить трассировку лучей для корректного моделирования отражающих и прозрачных объектов, которые вызывают каустики – области с резко возрастающей интенсивностью светового поля. Основой программной реализации служат инструменты OpenGL и OpenCL, которые являются стандартами в области графики и гетерогенных вычислений и обеспечивают поддержку оборудования различных производителей.

2. МОДЕЛИРОВАНИЕ КАУСТИК

2.1 Простейший подход

Для моделирования каустик может использоваться простой двухпроходный алгоритм. На *первом* проходе в соответствии с расположением и ориентацией источников света генерируются фотоны, число которых определяется заранее. В простейшем случае можно рассматривать *точечные* (испускают фотоны равномерно по всем направлениям) или *прямоугольные* (испускают фотоны в произвольном направлении полупространства) источники света.

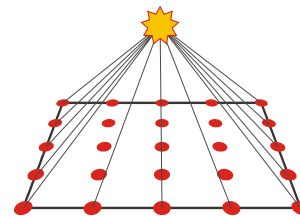


Рис. 1. Генерация фотонов для регулярных точек на прямоугольной площадке

В данной работе использовался точечный источник, однако для выделения значимых областей сцены выбирались только направления, проходящие через заданную прямоугольную площадку. Вместо использования генератора случайных чисел площадка покрывалась равномерной сеткой, для каждого узла которой генерировался фотон. При таком подходе размерность сетки определяет общее число фотонов

N , а фотонная карта может быть легко представлена в виде текстуры, каждый тексель которой соответствует точке регулярной сетки.

Для каждого фотона прослеживается траектория его движения и взаимодействия с объектами сцены. При соударении с *диффузным* объектом в фотонную карту записываются координаты найденной точки соударения. При соударении с объектами, обладающими *отражающими* или *прозрачными* свойствами, отслеживаются дальнейшие взаимодействия в направлениях отражения или преломления соответственно. Для фотонов, покинувших сцену без взаимодействия с объектами, записываются *недопустимые* координаты. Интенсивность всех фотонов считается одинаковой и равной параметру I_{photon} . Подход может быть обобщен для учета уменьшения интенсивности фотона при соударении с *неидеальными* отражающими или прозрачными объектами (необходимо сохранять не только координаты точки соударения, но и интенсивность на момент данного соударения) или *хроматических аберраций* (необходимо генерировать фотоны отдельно для каждой длины волны).

При *втором* проходе используется традиционный алгоритм трассировки лучей. При этом к вычисленной в точке соударения освещенности прибавляется интенсивность фотонов из некоторой окрестности, доступных посредством обращения к сформированной фотонной карте. В данной работе принимается, что на освещенность точки влияют *все* фотоны из ее окрестности вместо k ближайших, что в значительной степени упрощает процедуру сбора. Кроме того, принимается тривиальное соображение, что влияние фотона уменьшается с увеличением расстояния до точки соударения. В простейшем случае можно воспользоваться *линейной* весовой функцией, что приводит к следующей формуле:

$$I = I_{ray\ tracing} + I_{photon} \times \sum_{i=0}^N \max \left\{ 0, 1 - \frac{\|p - p_i\|}{\varepsilon} \right\} \quad (1)$$

Здесь p – координаты обрабатываемой точки соударения, p_i – координаты i -ого элемента фотонной карты. Параметр ε – радиус окрестности сбора фотонов, который подбирается эмпирически для каждой конкретной сцены.

Согласно работе [2] сбор всех фотонов из фиксированной окрестности вместо k ближайших может привести к худшей оценке освещенности в областях с малой плотностью фотонов или размытому результату в областях с высокой плотностью. Тем не менее, за счет применения излагаемых далее быстрых алгоритмов выигрыш производительности оказывается весьма существенным и оправдывает данный подход. Кроме того, в качестве альтернативной весовой функции можно использовать *интерполяцию Эрмита* вида $3t^2 - 2t^3$, t – параметр в диапазоне $[0, 1]$, отдавая больший приоритет близким фотонам и меньший – более отдаленным (в сравнении с линейным весом). Данный вид интерполяции реализован в GPU аппаратно и доступен через встроенную функцию `smoothstep` языка шейдеров GLSL [6]. С использованием данной функции формула (1) принимает вид:

$$I = I_{ray\ tracing} + I_{photon} \times \sum_{i=0}^N \max \left\{ 0, 1 - \text{smoothstep} \left(0, \varepsilon, \|p - p_i\| \right) \right\} \quad (2)$$

Применение модифицированной формулы (2) позволяет придать каустикам дополнительную резкость.

2.2 Сортировка фотонов

Описанный подход неэффективен, поскольку предполагает перебор *всех* элементов фотонной карты. Для повышения производительности можно использовать предварительную сортировку фотонной карты по координатам фотонов в лексикографическом порядке. Данная сортировка позволяет применять алгоритм бинарного поиска со сложностью $O(\log N)$ для определения индексов фотонов с *минимальной* и *максимальной* координатой в кубической ε -окрестности точки соударения. В результате вычисления на втором проходе сокращаются до перебора фотонов из *непрерывного* сегмента карты, заключенного между данными индексами:

$$I = I_{ray\ tracing} + I_{photon} \times \sum_{i=N_1}^{N_2} \max \left\{ 0, 1 - \frac{\|p - p_i\|}{\varepsilon} \right\} \quad (3)$$

Здесь N_1 и N_2 – индексы фотонов с минимальными и максимальными координатами соответственно. Сортировка может быть выполнена средствами ЦПУ или GPU. В первом случае целесообразно использовать *быструю* сортировку со сложностью $O(N \log N)$, во втором – *порядковую* [7] или *битоническую* [8] сортировку с трудоемкостью $O(N)$ и $O(N \log^2 N)$ соответственно.

Данный подход значительно повышает производительность визуализации, однако также не вполне эффективен: сбор фотонов осуществляется не из ε -окрестности точки соударения, а из ε -полосы вида $\|p_x - p_i\| \leq \varepsilon$. Далее будет рассмотрен алгоритм, позволяющий сократить перебор до некоторой окрестности точки соударения. Данный алгоритм развивает подход, предложенный в работе [3], обеспечивая более эффективное построение и использование ускоряющей структуры.

2.3 Воксельная ускоряющая структура

Разделим ограничивающий параллелепипед сцены на *воксели* со стороной ε (радиус окрестности для сбора фотонов). Тогда каждому фотону с координатами (x_p, y_p, z_p) можно сопоставить координаты вокселя, в котором он находится:

$$x_v = \left\lfloor \frac{x_p - x_{min}}{\varepsilon} \right\rfloor, y_v = \left\lfloor \frac{y_p - y_{min}}{\varepsilon} \right\rfloor, z_v = \left\lfloor \frac{z_p - z_{min}}{\varepsilon} \right\rfloor \quad (4)$$

Здесь через $(x_{min}, y_{min}, z_{min})$ обозначена *минимальная* точка ограничивающего параллелепипеда сцены.

После формирования фотонной карты описанным выше способом все фотоны сортируются в лексикографическом порядке по координатам соответствующих *вокселей*. При таком способе сортировки фотоны, принадлежащие одному вокселю, будут эквивалентны и займут *непрерывный* сегмент в отсортированной фотонной карте.

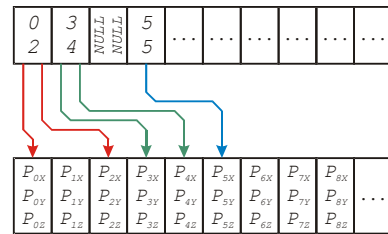


Рис. 2. Воксельная ускоряющая структура

Для отсортированной фотонной карты возможно создание воксельной ускоряющей структуры. Для этого необходимо

сформировать массив двумерных векторов, каждый элемент которого соответствует одному вокселю. В первую компоненту вектора записывается индекс первого фотона в соответствующем вокселе, во вторую – индекс последнего фотона. Для представления данного массива на ГПУ удобно использовать трехмерную текстуру.

Использование ускоряющей структуры на этапе визуализации основано на тривиальном соображении: все фотоны в ε -окрестности точки соударения принадлежат вокселям, координаты которых отличаются не более чем на 1 от координат вокселя для данной точки (в противном случае разность координат составляет не менее длины вокселя, которая равна ε). В итоге для вычисления освещенности в конкретной точке необходимо произвести суммирование по всем фотонам из $3^3 = 27$ вокселей. Окончательная формула будет выглядеть следующим образом:

$$I_{\text{photon}} \times \sum_{i=x_v-1}^{x_v+1} \sum_{j=y_v-1}^{y_v+1} \sum_{k=z_v-1}^{z_v+1} \sum_{l=N_1^{ijk}}^{N_2^{ijk}} \max \left\{ 0, 1 - \frac{\|p - p_l\|}{\varepsilon} \right\} \quad (5)$$

Здесь (x_v, y_v, z_v) – координаты вокселя, которому принадлежит рассматриваемая точка, N_1^{ijk} и N_2^{ijk} – индексы соответственно первого и последнего фотона в вокселе (x_v, y_v, z_v) . В большинстве случаев значимыми окажутся лишь несколько вокселей из окрестности, в то время как пустые воксели будут быстро отброшены.

2.4 Формирование массива вокселей

Для типичной сцены значение параметра ε лежит в диапазоне 0,5 – 1,5% от размера сцены (в качестве которого примем наибольшую сторону ограничивающего параллелепипеда). Такой выбор параметра ε порождает множество вокселей размером не более 256^3 элементов, что делает возможным создание и хранение данной ускоряющей структуры на ГПУ. Кроме того, в ряде случаев можно ограничиться *двумерным* массивом вокселей для моделирования каустик на плоской поверхности (например, на полу и стенах комнаты), что сокращает размер массива на два порядка. Для вычисления индекса первого и последнего фотона в каждом вокселе предлагается следующий параллельный алгоритм.

Для каждого элемента *отсортированной* фотонной карты выполняется проверка, является ли данный элемент *первым* в соответствующем вокселе (не лежит ли предыдущий элемент в *другом* вокселе). В случае положительного результата номер элемента записывается в соответствующий воксель в качестве индекса первого элемента. Аналогичная процедура выполняется для определения индекса последнего элемента вокселя. Поскольку в каждом вокселе существует единственный первый и последний элемент, не возникнет никаких коллизий при параллельном выполнении указанных операций. Данный алгоритм эффективно выполняется на ГПУ и имеет линейную сложность $O(N)$, где N – число генерируемых фотонов.

2.5 Схема результирующего алгоритма

В результате получаем следующий четырехпроходный алгоритм:

- Генерация фотонной карты с помощью *прямой* трассировки лучей. На ГПУ фотонная карта представлена двумерной текстурой размера $P_x \times P_y$, каждый тексель

которой соответствует ровно одному выпущенному фотону. Общее число фотонов $N = P_x \times P_y$. Для реализации данного этапа использовался шейдерный язык GLSL.

- “Воксельная” сортировка фотонной карты на ГПУ. Эффективными алгоритмами являются битоническая сортировка со сложностью $O(N \log^2 N)$ и поразрядная сортировка со сложностью $O(N)$. В данной работе реализована битоническая сортировка на базе OpenCL.
- Формирование массива вокселей для отсортированной фотонной карты на ГПУ. Предложенный алгоритм решает данную задачу за линейное время $O(N)$. Для реализации данного этапа использовался OpenCL.
- Визуализация сцены с помощью *обратной* трассировки лучей. Для обработки каустик выполняется сбор фотонов из окрестности точки соударения предложенным выше способом. Наряду с первым проходом для реализации выбран шейдерный язык GLSL. Оба этапа используют общие шейдеры с минимальными изменениями.

Предлагаемый алгоритм обеспечивает эффективную визуализацию каустик, выполняя подготовку необходимой ускоряющей структуры за *линейное* время $O(N)$ при использовании поразрядной сортировки.

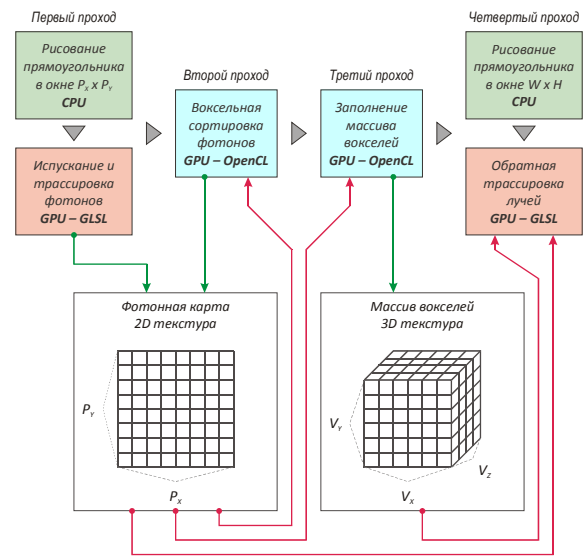


Рис. 3. Схема работы результирующего алгоритма

Необходимо сделать несколько замечаний относительно программной реализации. Текущая версия стандарта OpenCL не позволяет ядру выполнять чтение и запись в одну и ту же текстуру, что необходимо на этапе воксельной сортировки. Для обхода данного ограничения текстура с фотонной картой копируется в буфер (простой одномерный массив) перед этапом сортировки, содержимое которого вновь копируется в текстуру после окончания сортировки. На этапе формирования массива вокселей запись первого и последнего фотона осуществляется в различные компоненты текселя. Тем не менее, стандарт не позволяет обновлять *отдельные* компоненты текселя: записи подлежат сразу весь вектор цвета. Для обхода ограничения данный этап разбивается на два прохода: при первом проходе определяются индексы первых фотонов, а при втором – последних.

3. ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ

Оценка производительности рассмотренных алгоритмов проводилась на примере визуализации неявно заданной поверхности – “Metaballs” или “Blobs” [9]. Поверхность визуализировалась непосредственным образом без предварительной триангуляции. При расчете учитывался эффект полного внутреннего отражения при глубине трассировки 5.

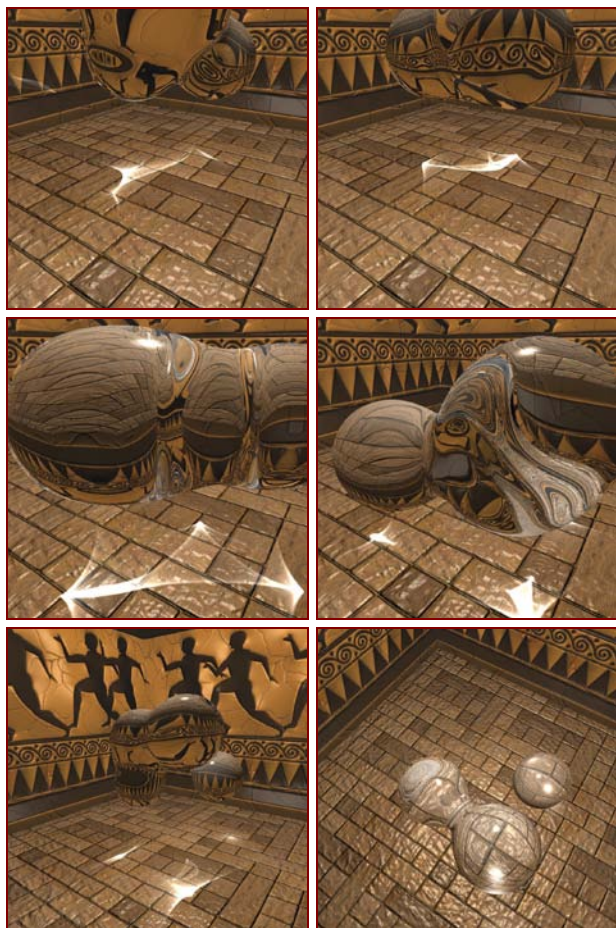


Рисунок 4. Пример визуализации “Metaballs” (число фотонов $N = 256 \times 256$, глубина трассировки 5)

Для проведения эксперимента использовались графические ускорители AMD Radeon 5870 (R800) и NVIDIA Quadro FX 5600 (G80). Карта AMD работала под управлением драйвера Catalyst 10.4 и ATI Stream SDK 2.1, карта NVIDIA – под управлением драйвера ForceWare 197.13 и CUDA Toolkit 3.0. Для тестовой сцены ограничивающий параллелепипед имеет размеры $10 \times 10 \times 10$, а значение параметра $\varepsilon = 0,04 = 0,4\%$ от размера сцены. Визуализация выполнялась в разрешении 1920×1080 (Full HD). В следующих таблицах представлены результаты замера производительности.

Результаты экспериментов позволяют сделать вывод о высокой эффективности предложенного метода. Воксельная ускоряющая структура более чем в четыре раза быстрее метода на основе лексикографической сортировки и обеспечивает качественную визуализацию каустик в реальном времени с разрешением Full HD.

Таблица 1. Лексикографическая сортировка фотонов

Число фотонов	Лексикографическая сортировка (мс)		Производительность визуализации (к/с)	
	R5870	Q5600	R5870	Q5600
$128^2 = 16K$	10,2	12,5	20	7,5
$256^2 = 65K$	14,5	24,2	8	3,2
$512^2 = 262K$	28,5	60,1	3	1,0

Таблица 2. Воксельная ускоряющая структура

Число фотонов	Воксельная сортировка и заполнение массива вокселей (мс)		Производительность визуализации (к/с)	
	R5870	Q5600	R5870	Q5600
$128^2 = 16K$	10,5	16,8	49,5	19,5
$256^2 = 65K$	17,5	30,3	35,8	14,5
$512^2 = 262K$	39,4	98,5	18,2	6,5
$1024^2 = 1M$	140	381	5,2	1,8

4. ЗАКЛЮЧЕНИЕ

Предложен упрощенный вариант метода фотонных карт для выполнения на ГПУ в реальном масштабе времени. Данный метод позволяет корректно моделировать каустики, возникающие при взаимодействии света с прозрачными или отражающими объектами. Эффективность визуализации достигается за счет воксельной ускоряющей структуры, которая формируется непосредственно на ГПУ средствами OpenCL. Формирование данной структуры включает в себя “воксельную” сортировку фотонов и определение для каждого вокселя индекса первого и последнего фотона. Данные задачи могут быть решены с помощью алгоритмов со сложностью $O(N)$. Предложенный подход обеспечивает высокую скорость работы при использовании фотонных карт вплоть до 1 миллиона элементов с разрешением Full HD.

5. ЛИТЕРАТУРА

- [1] Holly Rushmeier, David Banks, Peter Shirley. “A Basic Guide to Global Illumination”. SIGGRAPH 98 Course 5.
- [2] Henrik Wann Jensen. “Realistic Image Synthesis Using Photon Mapping”. AK Peters. Ltd., Massachusetts. 2001.
- [3] Timothy J. Purcell, Craig Donner, Mike Cammarano, Henrik Wann Jensen and Pat Hanrahan. “Photon mapping on programmable graphics hardware”. Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference, 2003.
- [4] Martin Fleisz. “Photon Mapping on the GPU”. Master’s thesis, School of Informatics, University of Edinburgh, 2009.
- [5] B. Fabianowski and J. Dingliana. “Interactive Global Photon Mapping”. Eurographics Symposium on Rendering, Volume 28 (2009), Number 4.
- [6] John Kessenich, Dave Baldwin, Randi Rost. “The OpenGL Shading Language” (Revision 8, 7 September 2006). <http://www.opengl.org/registry/doc/GLSLangSpec.Full.1.20.8.pdf>
- [7] “Radix sort” (From Wikipedia, the free encyclopedia). http://en.wikipedia.org/wiki/Radix_sort
- [8] “Bitonic sorter” (From Wikipedia, the free encyclopedia). http://en.wikipedia.org/wiki/Bitonic_sorter
- [9] Ryan Geiss. “Metaballs (Blobs)” (3/10/2000). <http://www.geisswerks.com/ryan/BLOBS/blobs.html>

Визуализация полупрозрачных объектов трехмерных сцен реального времени

Коростелев Е. И., Долговесов Б. С., Мазурок Б. С.
Институт Автоматики и Электрометрии СО РАН
Новосибирск, Россия
kore3d@gmail.com

Аннотация

В статье предлагается два метода эффективной визуализации полупрозрачных объектов, использующих один проход растеризации трехмерных объектов и один либо два дополнительных прохода постобработки. Применяемые концепции разработаны с учетом специфики графических процессоров последнего и предыдущих поколений, что позволяет использовать их в системах визуализации разного уровня.

Один из методов основан на трафаретных масках и реализуется для Direct3D9-совместимых графических процессоров, а другой требует аппаратной поддержки последних технологий Direct3D11, используемых для построения динамических списков слоев прозрачности. Кроме этого, на графическом процессоре выполняется сортировка слоев прозрачности. Ее детали рассматриваются в соответствующем разделе статьи.

Ключевые слова: *порядко-независимая прозрачность, визуализация реального времени, А-буфер, сортировка на графическом процессоре.*

1. ВВЕДЕНИЕ

Одной из известных проблем компьютерной графики реального времени является визуализация полупрозрачных объектов. Полупрозрачность используется при визуализации стеклянных поверхностей, природных явлений (применяются системы частиц), а также в системах автоматизированного проектирования (САПР). В САПР полупрозрачность позволяет наглядно представить внутреннюю структуру проектируемого изделия.

Стандартным методом визуализации полупрозрачных объектов с использованием различных аппаратных решений является применение формулы альфа-смешивания пикселей растеризуемых полигонов (alpha blending):

$$dest_rgb = dest_rgb \cdot (1 - src_alpha) + src_rgb \cdot src_alpha,$$

где *dest* – текущее значение в буфере цвета, а *src* – значение цвета смешиваемого с ним пикселя.

Результат применения представленной формулы зависит от порядка смешивания, и его нарушение приводит к появлению визуальных дефектов. Так, например, в левой части рисунка 1 нарушен порядок растеризации объектов от дальнего к ближнему, и поэтому альфа-смешивание дает неправильный результат – дальние объекты перекрывают ближние и визуально имеют неверное расположение. В итоге, на рисунке дракон визуально находится перед остальными объектами сцены, что не соответствует реальному

расположению. Чтобы этого избежать, на сегодняшний день, применяют сортировку объектов в пространстве наблюдателя. Но она не эффективна, поскольку не решает проблему целиком и ресурсоемка при большом количестве сортируемых объектов. Кроме этого, она требует дополнительного разбиения пересекающихся примитивов.



Рисунок 1: Пример результата стандартного альфа-смешивания и правильного альфа-смешивания

В системах визуализации сортировка элементов сцены, как правило, производится на центральном процессоре и требует постоянного обновления упорядоченных данных в памяти видеокарты. Устранение передачи этих данных по шине возможно путем переноса вычислений на графический процессор (GPU). В этом случае, освобожденные ресурсы центрального процессора могут быть использованы для решения более важных задач – обработка взаимодействия с пользователем, задачи искусственного интеллекта, подкачка данных и другие. Но в настоящее время сортировка на GPU объектов трехмерной сцены практически не применяется. Причиной является отсутствие эффективного решения для случая большого числа объектов.

Существуют другие подходы, позволяющие гарантировать правильный порядок смешивания. Их основой является сортировка по z-координате пикселей, участвующих в альфа-смешивании. Благодаря этому, результат визуализации не зависит от порядка растеризации полигонов и, соответственно, сортировка полупрозрачных объектов не требуется. Такие методы принято называть методами порядка-независимой прозрачности (order-independent transparency). Одним из их достоинств является корректная обработка случая пересекающихся примитивов.

Одним из методов порядка-независимой прозрачности является метод расслоения по глубине (depth peeling) [1]. Он

обладает низкой производительностью, поскольку для разделения на слои (рисунок 2) применяется многопроходная визуализация. Худшая его производительность достигается на трехмерных сценах из множества анимированных полупрозрачных объектов с различными настройками материалов. Так, использование множества параметров приводит к многократному переключению состояний графического конвейера, что существенно снижает производительность. Уменьшение ресурсоемкости в этом случае достигается путем сокращения числа используемых проходов и объема памяти, резервируемого под слои прозрачности. Оптимизации такого рода являются основой различных модификаций этого метода: dual depth peeling [2], reverse depth peeling [3] и другие [4]. Но, они, как и оригинальный метод, широко не применяются, поскольку их производительности не достаточно для современных систем визуализации.



Рисунок 2: Концепция расслоения по глубине.

В ходе развития возможностей аппаратных решений предлагались и другие методы порядко-независимой прозрачности. Часть из них использует чересстрочную растеризацию [5] для записи значения слоя прозрачности поверх изображения непрозрачных, другие – реализуют концепцию K-буфера [6,7] или A-буфера [8,9,10]. Под K-буфером подразумевается абстракция структуры памяти для хранения ровно K слоев прозрачности, а A-буфер, в свою очередь, подразумевает хранение произвольного числа слоев прозрачности. Эти методы имеют ряд серьезных ограничений: требуют современного аппаратного обеспечения и обладают низкой производительностью, обрабатывая при этом лишь небольшое число слоев.

В различных публикациях также рассматриваются методы, реализация которых в настоящий момент невозможна [11,12,13]. Это ограничение связано с отсутствием у современных графических процессоров функции атомарного исполнения произвольных блоков инструкций.

Далее, в статье предлагаются два новых метода, которые обладают неплохой производительностью и имеют меньше ограничений по сравнению с существующими. Кроме этого, они позволяют эффективно визуализировать полупрозрачные объекты, как на новом аппаратном обеспечении (на основе архитектуры AMD Terascale2 и NVIDIA Fermi [14]), так и на аппаратном обеспечении предыдущих поколений (Direct3D9-совместимые видеокарты).

2. ПОРЯДКО-НЕЗАВИСИМАЯ ПРОЗРАЧНОСТЬ

В данном разделе статьи описываются два новых метода порядко-независимой прозрачности, которые обеспечивают правильную визуализацию полупрозрачных объектов. При этом непрозрачные объекты визуализируются отдельно, и

значения цвета и глубины после этого применяются для отсеивания невидимых полупрозрачных пикселей и в формуле альфа-смешивания. Для этого используется чтение буфера глубины, либо отдельной текстуры, заполненной соответствующими значениями. Важной деталью обоих методов является сортировка слоев прозрачности на графическом процессоре. Ее особенности представлены в следующем разделе статьи.

2.1 Метод трафаретных масок

Основная идея метода заключается в использовании маскирования пикселей для разделения их на слои прозрачности. При этом применяется тест и буфер трафарета. В буфер предварительно записываются номера выделяемых слоев прозрачности, а на этапе визуализации выполняется проверка текущих значений на равенство нулю. Если тест пройден, то значение перезаписывается на 255, если нет – уменьшается на единицу. Таким образом, в буфер цвета попадают только пиксели, соответствующие маскам слоев прозрачности. Пиксели различных слоев при этом записываются по соседству и образуют группы. В итоге, разрешение получаемого изображения получается меньше. На рисунке 3 приводится пример его заполнения, где хорошо наблюдается наличие групп размером 4 на 4.



Рисунок 3: Пример заполнения буфера цвета (цвета инвертированы)

Дополнительно используется еще один буфер. В него производится запись значения z-координаты (в пространстве наблюдателя) соответствующего пикселя. Кроме этого, в реализации метода буферы представляются текстурами формата R8G8B8A8 и R16F. То есть 32 бита для 4-канального целочисленного значения цвета и 16 бит для 1 вещественного значения глубины, соответственно.

Сохранение разрешения финального изображения обеспечивается за счет использования буферов большего размера, чем размеры формируемого кадра (применяется supersampling). В этом случае, N-кратное увеличение буфера результата растеризации позволяет выделять до N слоев прозрачности без потери разрешения. Далее, отдельным проходом постобработки, выполняется его чтение и сортировка. Полученные слои прозрачности упорядочиваются по z-координате, и значения их цветов смешиваются по формуле альфа-смешивания.

Расчет слоев прозрачности по группе пикселей в общем случае приводит к появлению визуальных дефектов. Они возникают на границах полигонов, поскольку в этой области растеризатор формирует не все пиксели группы. Таким образом, происходит модификация не всех масок и выделяются не те слои прозрачности. В итоге, слои могут быть пропущены либо продублированы. Суть этой проблемы поясняется рисунком 4. Пусть левый треугольник полностью покрывает группу A и частично B, D, E, а правый – полностью E, F и частично B. В этом случае в группах B, D, E при растеризации будут модифицированы не все маски, что приведет к неверному альфа-смешиванию и визуальному дефекту.

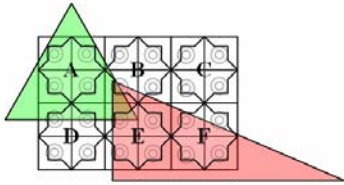


Рисунок 4: Пример растеризации треугольников

Устранение описанных дефектов производится путем фильтрации областей финального изображения, где потенциально могло произойти неправильное альфа-смешивание. Нахождение таких мест производится проверкой трех условий: разница суммы глубины по всем слоям прозрачности меньше некоторого константного значения, совпадение четности числа слоев прозрачности, расстояние в кубе RGB цветов меньше некоторой константы. При невыполнении любого из них соответствующий пиксель помечается некоторым значением ошибки (например, 0.25, 0.75 или 0.5, соответственно) Пример выделения таких областей приводится на рисунке 5 (линии на границах).

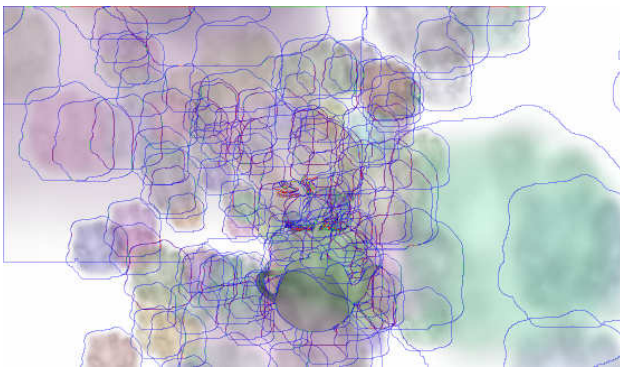


Рисунок 5: Пример маркирования потенциально дефектных пикселей (используется альфа-тест)

Используемый алгоритм фильтрации заключается в анализе изображения и обработке помеченных текстелей модифицированным фильтром Гаусса. Применяемая модификация заключается в выборе для усреднения из 12 ближайших соседей только текстелей, имеющих минимальное значение ошибки. Не соответствующие этому условию соседи могут использоваться с меньшими коэффициентами.

2.2 Метод динамических списков слоев прозрачности

Рассматриваемый далее метод является более универсальным по сравнению с методом трафаретных масок. Он не требует дополнительной фильтрации и использует меньше памяти под хранение слоев прозрачности. Экономия памяти достигается благодаря построению динамических списков поверх линейной памяти. Ее заполнение выполняется на этапе визуализации полупрозрачных объектов и реализует концепцию A-буфера [8]. Далее, отдельным проходом постобработки, элементы каждого списка сортируются и используются для расчета правильного альфа-смешивания значений цветов.

Общая схема работы алгоритма построения динамических списков описывается следующими шагами:

- Атомарно увеличить счетчик занятой памяти.
- Атомарно обновить текущую ячейку буфера индексов начал списков предыдущим значением счетчика.
- Записать в память цвет, глубину и адрес следующего элемента, который соответствует предыдущему значению в текущей ячейке буфера индексов начал списков.

Выполнение этих шагов гарантирует запись данных текущего слоя в незанятую ранее область памяти и приводит к построению связанных списков. В результате, начало формируемых списков всегда лежит по адресу, указанному в буфере индексов начал списков, а последний элемент списка хранит значение, которым был инициализирован буфер.

В качестве оптимизаций, используется один дополнительный буфер для хранения длины списка и память под требуемые буферы выделяется на видеокarte в виде объектов 2D-текстур. Для доступа на запись используется технология UAV (unordered access view), входящая в состав DirectX версии 11. Кроме этого, для синхронизации операций между потоками, применяются атомарные операции.

Использование метода на современном аппаратном обеспечении позволяет эффективно обрабатывать до 64 слоев прозрачности. В большинстве случаев этого более чем достаточно.

3. СОРТИРОВКА НА GPU

В этом разделе рассматриваются методы сортировки за один проход на графическом процессоре большого количества коротких последовательностей данных. Обрабатываемые последовательности имеют длину не более 64 и хранятся в памяти видеокарты. Другие длины не рассматриваются, поскольку для задачи сортировки слоев прозрачности этого достаточно и при большем N более предпочтительна сортировка в несколько проходов постобработки [15].

Решение задач на графическом процессоре имеет свою специфику. Так, например, потоки выполняются на симметричных процессорах группами, и потоки в группе выполняют одну инструкцию [14]. В связи с этим, операция ветвления является одной из дорогих. Поэтому предпочтительно использование слабовевающих алгоритмов. Кроме этого, особенности архитектуры не позволяют реализовывать рекурсивные вызовы. Все эти факторы влияют на выбор эффективного алгоритма.

Проводилось тестирование производительности различных алгоритмов сортировки: вставками (insertion sort), Шелла с $h = 2^p 3^q$ (Shell sort), битоническая (bitonic sort), нечетно-четная слиянием Бэтчера (odd-even merge sort) и сортирующая сеть [16,17]. Сортирующие сети, путем взаимодействия компараторов (блоков сравнения-обмена), позволяют сортировать последовательности некоторой ограниченной длины. В качестве такой сети в тестировании используется цепочка компараторов, получаемая путем развертки алгоритма нечетно-четной сортировки слиянием Бэтчера. Максимальная длина сортируемой последовательности при этом фиксируется некоторым N . Пример такой сети представлен на рисунке 6.

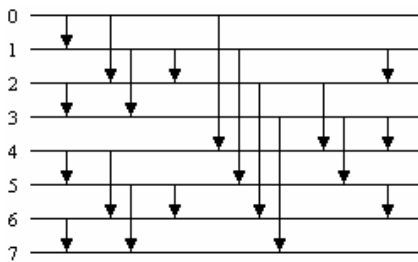


Рисунок 6: Сортирующая сеть для 8 значений

Как известно, алгоритмы Шелла с $h = 2^p 3^q$ и битонической сортировки дают эффективные сортирующие сети. Но они, по сравнению с используемой, имеют большее число компараторов. Так, например, битоническая имеет $S(2^k) = 2^{k-2} \cdot k \cdot (k+1)$ компараторов, что больше нечетно-четной сети Бэтчера – $S(2^k) = 2^{k-2} \cdot k \cdot (k-1) + 2^k - 1$.

В таблице 1 приводятся результаты измерения времени работы алгоритма на видеокарте NVIDIA GeForce 8600 MGT. Компиляция программ для видеокарты выполнялась в рамках шейдерной модели третьей версии. В качестве данных для сортировки использовались последовательности длины от 0 до N , заданные стандартным генератором псевдослучайных чисел. Общее число последовательностей при этом составляло 65536. По полученным результатам видно, что упорядочение сортирующей сетью наиболее эффективно.

Сортировка	N = 4	N = 8	N = 16	N = 32
Вставками	3 мс	13 мс	58 мс	296 мс
Шелла ($h = 2^p 3^q$)	6 мс	61 мс	480 мс	3887 мс
Битоническая	4 мс	34 мс	229 мс	1392 мс
Нечетно-четная слиянием	4 мс	30 мс	198 мс	1214 мс
Сортирующая сеть	1 мс	2 мс	4 мс	17 мс

Таблица 1: Производительность методов сортировки 65536 N-элементных последовательностей на GeForce 8600 MGT

В таблице 2 приводятся данные о производительности рассматриваемой сортирующей сети. Производительность указана в элементах в секунду и верна для последовательностей длины N . Данная таблица позволяет приблизительно оценить вклад сортировки в общее время

работы предлагаемых методов визуализации полупрозрачности.

Сортирующая сеть компараторов с N входами	Производительность на GeForce 8600 MGT
N = 4	1786 Мэлемент/сек
N = 8	608 Мэлемент/сек
N = 16	337 Мэлемент/сек
N = 32	133 Мэлемент/сек

Таблица 2: Производительность сортировки сетью компараторов N-элементных последовательностей

Для окончательной оценки производительности сортировок и выбора оптимальной для GPU приводится дополнительный график (рисунок 7). Отраженные на графике результаты получены на той же видеокарте и показывают зависимость времени (в мс) сортировки от числа сортируемых элементов в последовательности (определяется значением N). Последовательность при этом ограничена 32 элементами и число сортируемых последовательностей равно 65536. Верхнее ограничение на длину последовательности задается для фиксирования числа используемых при расчетах регистров вычислительных блоков видеокарты.

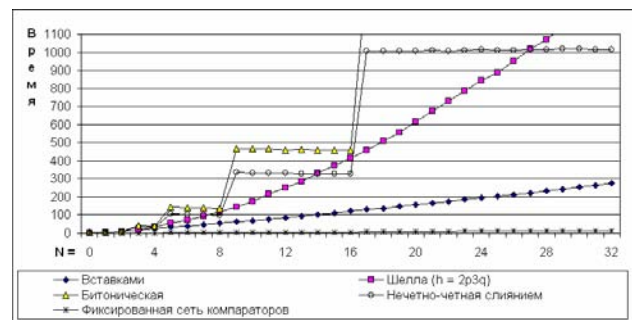


Рисунок 7: Сравнение методов сортировки N элементов из 32-элементных последовательностей

По результатам тестирования предпочтение отдается методу упорядочения сортирующей сетью, построенной алгоритмом нечетно-четной сортировки слиянием Бэтчера. Далее этот метод используется перед этапом альфа-смешивания, чтобы упорядочить слои прозрачности.

4. РЕЗУЛЬТАТЫ

Тестирование методов проводилось на сценах, состоящих из множества полупрозрачных объектов и имеющих большое число слоев прозрачности. В качестве тестовых аппаратных платформ использовались: компьютер с видеокартой ATI Radeon HD5670 и лэптоп с видеокартой NVIDIA GeForce 8600 MGT. Демонстрационные приложения запускались в оконном режиме и разрешении 1680x1050, а их производительность измерялась под операционной системой Microsoft Windows 7 Professional x64. Пример результата работы приложения приводится на рисунке 8.



Рисунок 8: Визуализация полупрозрачных объектов с использованием трафаретных масок

Для сравнения на рисунке 9 приводится результат визуализации полупрозрачных объектов путем применения стандартного альфа-смешивания. При этом он является некорректным – объекты визуально располагаются неверно.



Рисунок 9: Результат стандартного альфа-смешивания

4.1 Производительность метода трафаретных масок

Тестирование метода проводилось на сцене из 384 текстурированных треугольников с моделями прозрачного дракона из 871318 треугольников и непрозрачного чайника из 2256 треугольников. Число слоев прозрачности – от 0 до 15.

В таблице 3 приводится статистика производительности при разном числе обрабатываемых слоев прозрачности. Результат соответствует описанной сцене. Кроме этого, значение Мпикс/сек характеризует количество полупрозрачных пикселей, обрабатываемых за одну секунду. Графа таблицы с прочерком – метод стандартного альфа-смешивания.

Число слоев	Кадров в секунду	GeForce 8600 MGT
-	40	155 Мпикс/сек
8	11	41 Мпикс/сек
16	5	19 Мпикс/сек

Таблица 3: Производительность метода трафаретных масок

4.2 Производительность метода динамических списков слоев прозрачности

Тестирование метода динамических списков слоев прозрачности проводилось на сцене из 2048 треугольников с RGBA-текстурами с моделью прозрачного робота из 261431 треугольников и непрозрачной моделью чайника из 2256 треугольников. Число слоев прозрачности на пиксель при построении изображения находилось в диапазоне от 0 до 57. Статистика производительности метода приводится в нижеследующей таблице. При этом графа с прочерком соответствует методу стандартного альфа-смешивания.

Число слоев	Кадров в секунду	Radeon HD 5670
-	210	1661 Мпикс/сек
8	25	165 Мпикс/сек
16	21	161 Мпикс/сек
32	13	102 Мпикс/сек
64	7	55 Мпикс/сек

Таблица 4: Производительность метода динамических списков слоев прозрачности

5. ЗАКЛЮЧЕНИЕ

В статье рассмотрена проблема визуализации полупрозрачных объектов трехмерных сцен реального времени, используемых в современных системах виртуальной реальности и системах автоматизированного проектирования. Предложено два новых метода визуализации, которые могут быть легко встроены в существующий цикл визуализации и могут применяться на видеокартах разных поколений.

6. СПИСОК ЛИТЕРАТУРЫ

- [1] C. Everitt, “Interactive order-independent transparency”, Technical report, NVIDIA Corp., 2001.
- [2] L. Bavoil, K. Myers, “Order independent transparency with dual depth peeling”. Technical report, NVIDIA Corp., 2008.
- [3] N. Thibieroz, “Robust Order-Independent Transparency via Reverse Depth Peeling”, ShaderX6, pp. 211–226.
- [4] Meng-Cheng Huang Fang Liu et al., “Efficient depth peeling via bucket sort”, Proceedings of the Conference on High Performance Graphics 2009, pp.51-57.
- [5] D. Pangerl, “Deferred Rendering Transparency”, Article 2.7, ShaderX7.
- [6] K. Myers, L. “Bavoil, Deferred Rendering using a Stencil Routed K-Buffer”, ShaderX6, 2008, pp. 189–198.
- [7] Meng-Cheng Huang Fang Liu et al., “Multi-Fragment Effects on the GPU using Bucket Sort”, Article 8.1, GPU PRO, AK Peters, 2010.
- [8] L. Carpenter, “The A-buffer, an antialiased hidden surface method”, Proceeding of the 11th annual conf. On Computer graphics and interactive techniques, 1984, pp.103-108.
- [9] K. Myers, L. Bavoil, “Stencil routed A-Buffer”, ACM SIGGRAPH Technical Sketch, 2007.

- [10] C. Pepper, “Prefix sum pass to linearize A-buffer storage”, Patent, Microsoft Corp., 2006.
- [11] B.-Q. Liu, L.-Y. Wei, Y.-Q. Xu, “Multi-Layer Depth Peeling via Fragment Sort”, Tech report, Microsoft Research Asia, 2006.
- [12] L. Bavoil et al., “Multi-Fragment Effects on the GPU using the k-Buffer”, Proceedings of the 2007 symposium on Interactive 3D graphics and games, 2007, pp.97-104.
- [13] D. Pangerl, “ZT-Buffer Algorithm”, Article 2.8, ShaderX5: Advanced Rendering Techniques, Wolfgang Engel, Ed., Charles River Media, 2007, p.151–157.
- [14] NVIDIA, “NVIDIA’s Next Generation CUDA Compute Architecture: Fermi”, 2010.
- [15] Kipfer, R. Westermann, “Improved GPU Sorting”, Chapter 46, GPU Gems II.
- [16] D. Knuth, Volume 3: Sorting and Searching, The Art of Computer Programming, Addison-Wesley, 1997, pp.219–247.
- [17] R. Sedgewick, “Sorting Algorithms”, Algorithms in C++ (2nd edition), Addison-Wesley, 1992, pp. 93–192

Real-time rendering of semi-transparent objects

Abstract

The article proposes two techniques for efficient rendering of semi-transparent objects by single-pass rasterization of 3d objects and one or two additional stages of postprocessing. Applied concepts were developed with a glance to features of the latest and previous generations of GPU. This fact makes it possible to use these concepts in different virtual reality systems.

One of these methods is based on masking by stencil test and can be applied with Direct3D9-compatible videocards, while the other one requires a hardware support of the latest technologies used to create dynamic lists of transparent pixels. In addition, methods use the gpu for sorting multiple layers of semi-transparent pixels. This aspect is described in the relevant section of the article.

Keywords: *order-independent transparency, real-time rendering, A-buffer, gpu sorting.*

About the authors

Evgeny I. Korostelev is a Ph.D. student at Institute of Automation and Electrometry SB RAS. His contact email is kore3d@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

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

Алгоритм восстановления поверхности из облака точек на графическом процессоре

Дмитрий Козлов

Нижегородский Государственный
Университет им. Н.И. Лобачевского

dmitry.a.kozlov@gmail.com

Вадим Турлапов

Нижегородский Государственный
Университет им. Н.И. Лобачевского

vadim.turlapov@cs.vmk.unn.ru

Аннотация

В данной работе представлен алгоритм восстановления поверхности из облака точек трехмерного пространства, расположенных вблизи некоторой двусторонней замкнутой поверхности M . Описанный алгоритм принимает на входе множество точек $\{x_1, \dots, x_n\} \subset \mathbb{R}^3$ и порождает на выходе триангуляцию, аппроксимирующую поверхность M . Алгоритм не использует никакой дополнительной информации, кроме содержащейся в наборе входных точек, извлекая информацию о границах поверхности и ее топологии из особенностей входных данных. Алгоритм разработан для выполнения на устройствах с массовым параллелизмом и реализован при помощи технологии гетерогенных вычислений OpenCL. Приложениями вышеуказанного алгоритма могут выступать обработка данных с 3D сканера, восстановление объектов по набору сечений, обработка моделей с точечным представлением и т.д.

Ключевые слова

Восстановление поверхностей, анализ данных со сканера, графический процессор, OpenCL, CUDA, анализ главных компонент, метод марширующих кубов.

1. ВВЕДЕНИЕ

Проблема восстановления поверхности из облака точек встает во множестве областей компьютерной графики, одной из которых является анализ данных, полученных с 3D сканера. Такие данные обычно представляются двумерным массивом значений расстояния от сенсора сканера до соответствующей точки объекта (карта глубины). Сканирование может выполняться с нескольких точек обзора и порождать множество таких наборов данных. Так же облака точек могут быть получены при помощи различного медицинского оборудования и программного обеспечения. Первым этапом анализа данных такого типа обычно является восстановление поверхности и ее триангуляция с целью получить базовую аппроксимацию. Триангуляция может использоваться в дальнейшем для параметризации такой поверхности функциями более высоких порядков. В данной области было разработано множество алгоритмов, часть из которых использует априорные данные о входных точках [6], другая же опирается только на информацию, содержащуюся во входном наборе [5,2]. Отличительной чертой всех алгоритмов восстановления поверхностей является их исключительная ресурсоемкость, обусловленная необходимостью обработки большого числа однотипных данных. В основе работы лежит алгоритм, приведенный в статье [5], который был существенным образом переработан

для эффективного исполнения на архитектурах с массовым параллелизмом, разработаны структуры хранения данных, рассчитанные на такие устройства.

2. ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ

В области восстановления поверхностей было предложено множество алгоритмов, которые можно классифицировать по различным признакам. Как уже говорилось алгоритмы восстановления можно разделить по использованию априорных данных на те, которые используют дополнительные знания о входных точках и те, которые опираются только на информацию, заключенную во входном наборе. Работа [2], например, использует гладкость многообразия, в то время как алгоритм, описанный в [6] не предполагает каких-либо «хороших» свойств у поверхности M . Алгоритмы восстановления так же могут быть классифицированы по типу функции, аппроксимирующей поверхность. В качестве такой функции может использоваться неявное или параметрическое представление. В работах [6,10] строится аппроксимация поверхности суммой Гауссовых функций с различными параметрами, а в [15,16] используется построение параметрической модели над прямоугольной областью определения. Методы «подгонки» параметров функции тоже сильно разнятся. В работе [14] используются вариации метода наименьших квадратов с различными вариантами метрик. Работа [13] опирается на метод деформации сфер, а [2] построена на базе диаграмм Вороного и так называемых α -форм. До недавнего времени предпринималось сравнительно мало попыток распараллеливания алгоритмов восстановления поверхностей, из которых стоит отметить [8], где используется октодереве в качестве структуры данных для хранения обрабатываемых точек. В работе [6] исследуется масштабируемость алгоритма восстановления на основе поверхностей Пуассона при использовании адаптивной сетки в качестве ускоряющей структуры.

3. АЛГОРИТМ

Приведем краткий обзор используемого алгоритма. Алгоритм состоит из двух основных стадий. На первой происходит аппроксимация неявной функции $F: D \rightarrow \mathbb{R}$, где $D \subset \mathbb{R}^3$ некоторый слой, заключающий в себе поверхность M . Данная функция представляет собой оценку расстояния от любой точки слоя D до поверхности M . Ее значение может быть положительным или отрицательным, в зависимости от того с какой стороны поверхности находится точка. На второй стадии алгоритма производится триангуляция поверхности

нулевого уровня функции F при помощи алгоритма марширующих кубов [9].

3.1 Восстановление поля нормалей

Для построения функции расстояния мы, следуя [5] ассоциируем с каждой точкой набора $\{x_1, \dots, x_n\}$ ориентированную касательную плоскость $Tr(x_i)$. Каждая такая плоскость описывается некоторой точкой o_i и нормалью n_i . Точка o_i рассчитывается как центр масс k ближайших к x_i точек:

$$o_i = \frac{\sum_{x \in KNbh(x_i)} x}{K}$$

Нормаль n_i определяется, используя те же ближайшие k точек при помощи метода главных компонент. Для этого нужно построить ковариационную матрицу для данного набора из k точек:

$$CV = \sum_{x \in KNbh(x_i)} (x - o_i) \otimes (x - o_i)$$

Где \otimes обозначает тензорное произведение векторов. Ковариационная матрица является симметричной и положительно-полуопределенной. Далее необходимо найти собственный вектор этой матрицы, соответствующий минимальному по модулю собственному числу. Данный вектор характеризует направление, в котором разброс точек минимален. Это направление мы и принимаем за приближенное направление нормали.

3.2 Ориентация нормалей

Если две точки x_i и x_j расположены достаточно близко, то их нормали должны быть практически параллельными ($n_i n_j \approx 1$). Тем не менее, алгоритм их нахождения, описанный выше, не гарантирует нам такой ориентации, они могут быть направлены в противоположные стороны. Таким образом, в алгоритме возникает подзадача корректной ориентации нормалей. Стоит заметить, что для некоторых приложений алгоритма, таких как задача обработки сканированных данных, этот шаг алгоритма может быть опущен, так как данные с 3D сканера уже содержат, как правило, направление нормали к поверхности. Для решения данной задачи строится граф $G = (V, E)$, где множеству вершин соответствуют точки данных, а ребро $(v_i, v_j) \in E$ если x_i является одной из k ближайших к x_j точек. Далее каждому ребру (v_i, v_j) присваивается вес $1 - |n_i n_j|$, после чего задача сводится к нахождению минимального остовного дерева графа G и распространению вдоль его ребер правильной ориентации, что соответствует распространению по направлению минимальной кривизны. За начальную ориентацию нормали выбирается положительное направление оси z для точки с максимальной z координатой, после чего производится распространение направления от данной вершины по ребрам минимального остовного дерева. В нашей реализации используется параллельный алгоритм Боровки для построения минимального остовного дерева, описанный в [3].

3.3 Вычисление функции расстояния

Функция расстояния является оценкой расстояния от точки слоя D до поверхности M . Ее вычисление осуществляется по следующей формуле:

$$F(p) = (p - o_i)n_i$$

Где o_i – ближайший к p «центр» касательной плоскости, а n_i – соответствующая ему нормаль.

3.4 Триангуляция

Триангуляция осуществляется при помощи известного алгоритма марширующих кубов [9], предназначенного для триангуляции поверхностей равного уровня скалярного поля. Данный алгоритм основан на вычислении значений упомянутого скалярного поля в узлах регулярной сетки и последующей генерации треугольников для каждой из ячеек сетки на основе битовой маски. Мы используем адаптивную модификацию этого алгоритма для построения триангуляции множества $F(p) = 0$, которое является аппроксимацией поверхности M .

4. ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ

4.1 Терминология

В нашей параллельной реализации широко используются примитивы параллельного программирования, описанные в [3,4]. Таблица 1 содержит краткое описание таких примитивов.

Название операции	Краткое описание	Сложность (CRCW PRAM)
Reduce	Редукция массива с использованием ассоциативной операции	$O(\log_2 N)$
Scan	Нахождение префиксной суммы	$O(\log_2 N)$
Compact	Удаление повторяющихся элементов	$O(\log_2 N)$
Sort	Сортировка массива	$O((\log_2 N)^2)$

Таблица 1. Параллельные примитивы

4.2 Ускоряющая структура

Для эффективного исполнения алгоритма на GPU необходимо применение ускоряющей структуры для входного набора данных. Используемые нами алгоритмы предъявляют следующие требования к ускоряющей структуре:

- Эффективный доступ к информации о ближайших соседях точки
- Эффективный алгоритм построения структуры на GPU (в случае динамического облака)
- Эффективное использование видео памяти
- Простота реализации

Основываясь на этих требованиях, мы выбрали в качестве ускоряющей структуры регулярную сетку. Был разработан эффективный алгоритм параллельного построения данной структуры и удобный формат хранения данных, позволяющий избежать избыточного использования видео памяти. Листинг 1 содержит описание алгоритма формирования регулярной сетки в терминах приведенных ранее параллельных примитивов.

Листинг 1. Параллельное построение регулярной сетки

```
// Шаг 1: Построение bounding box'a
1: BBox = new bbox
2: Reduce(Q, BBox, union operator)
// Шаг 2: Вычисление идентификаторов точек
3: Pid = new array [N]
4: for each i = 0 to N-1 in parallel
5:   pid[i] = VoxelId(Q[i]) << 32 + i
// Шаг 3: Сортировка идентификаторов точек
6: SortedPid = new array [N]
7: Sort(Pid, SortedPid, N)
// Шаг 4: Переупорядочивание точек
8: SQ = new array [N]
9: for each i = 0 to N-1 in parallel
10:  SQ[i] = Q[SortedPid[i] & 0xFFFFFFFF]
// Шаг 5: Выделение вокселей сетки
11: Marks = new array [N + 1]
12: for each i = 1 to N-1 in parallel
13:  if ( (SortedPid[i] >> 32) != (SortedPid[i-1]>>32)
14:    Marks[i] = 1
15:  else Marks[i] = 0
16: Scan(Marks, Marks, +)
17: NumVoxels = Marks[N]
18: Voxels = new array [NumVoxels]
19: Compact(Voxels, Marks, SQ)
```

На первом шаге алгоритма строится bounding box облака точек, путем применения примитива Reduce к исходному массиву точек. Второй шаг предназначен для вычисления уникальных 64-битных идентификаторов точек, которые позволят переупорядочить массив точек таким образом, что точки, принадлежащие одному и тому же вокселю, будут располагаться в видеопамяти последовательно. На третьем шаге производится сортировка массива идентификаторов. Первичным ключом сортировки является идентификатор вокселя, вычисляемый как индекс в развертке трехмерного массива, вторичным – индекс точки в начальном массиве. Далее массив точек переупорядочивается в соответствии с новым порядком идентификаторов. Таким образом, точки, попадающие в один воксель сетки, лежат в видеопамяти последовательно, что позволяет сохранять лишь индекс начальной точки и количество точек, используя константный объем памяти для каждого вокселя. На пятом шаге происходит выделение групп точек, принадлежащих одному и тому же вокселю. Для этого строится массив маркеров в котором i -й элемент содержит единицу если точки с индексами i и $i-1$ принадлежат различным вокселям и ноль в противном случае. Путем применения примитива Scan к массиву маркеров мы находим для каждой точки индекс первой точки текущего вокселя. Примитив Compact для каждого вокселя сохраняет индекс первой точки в массиве и

количество точек, содержащихся в данном вокселе. Стоит отметить, что в памяти сохраняются только те воксели, которые содержат точки данных, что позволяет значительно снизить использование видеопамяти.

4.3 Восстановление поля нормалей

Для построения оценки расстояния необходимо нахождение касательной плоскости для каждой из точек входных данных. Далее приводится описание параллельного алгоритма построения оценки расстояния с использованием нашей ускоряющей структуры.

1. Получить из глобальной памяти обрабатываемую точку
2. Определить целочисленные координаты вокселя (x, y, z)
3. Выполнить выборку индексов точек вокселя (x, y, z) и его соседей
4. Скопировать точки с полученными индексами в локальную память
5. Отсортировать точки по возрастанию расстояния
6. Выбрать из локальной памяти k ближайших точек
7. Вычислить центроид c и поместить его в выходной буфер
8. Вычислить матрицу ковариации для выбранных точек
9. Найти минимальное собственное число и поместить соответствующий ему собственный вектор в выходной буфер

4.4 Вычисление оценки расстояния

Для вычисления оценки расстояния мы помещаем центроиды в ускоряющую структуру, что позволяет использовать следующую параллельную схему:

1. Получить из глобальной памяти обрабатываемую точку
2. Определить целочисленные координаты вокселя (x, y, z)
3. Найти ближайший центроид и соответствующую ему нормаль, используя ускоряющую структуру
4. Вычислить оценку расстояния

4.5 Алгоритм маршрутирующих кубов

Нахождение поверхности нулевого уровня оценки расстояния осуществляется при помощи алгоритма маршрутирующих кубов, который состоит в расчете значения функции в узлах регулярной сетки и последующей генерации треугольников для каждой ячейки этой сетки. Благодаря разработанной нами структуре, алгоритм существенным образом ускоряется, за счет того, что им обрабатываются только те воксели, которые действительно могут содержать поверхность нулевого уровня оценки расстояния. В листинге 2 приведено описание алгоритма.

Листинг 2. Алгоритм марширующих кубов

```

// Шаг 1: Вычисление шаблонов ячеек
1: Case = new array [NumVoxels]
2: for each i = 0 to NumVoxels-1 in parallel
3:     Case[i] = ComputeCase(F, Voxels[i]);
// Шаг 2: Вычисление адресов вершинного буфера
4: NumTris = new array [NumVoxels + 1]
5: for each i = 0 to NumVoxels-1 in parallel
6:     NumTris[i] = LookUp(Case[i]);
7: Scan(NumTris, NumTris, +)
8: TotalTris = NumTris[NumVoxels]
// Шаг 3: Генерация треугольников
8: Tris = new array [TotalTris]
9: for each i = 0 to NumVoxels-1 in parallel
10:    GenerateTris(Tris, NumTris[i]);

```

На первом шаге алгоритма для каждого вокселя вычисляется маска, которая определяет конфигурацию треугольников в данном вокселе. Для того, чтобы создать и заполнить вершинный буфер необходимо определить точное количество треугольников, генерируемое для всех вокселей. На втором шаге создается массив, каждый элемент которого содержит количество треугольников, содержащееся в соответствующем вокселе. Применяя примитив Scan к данному массиву, мы получаем общее количество треугольников и индексы для записи треугольников для каждого вокселя. Такая техника позволяет получить выходные данные различной длины от каждого потока в один буфер параллельно.

5. ОГРАНИЧЕНИЯ АЛГОРИТМА

Алгоритм применяется для восстановления замкнутых двусторонних поверхностей, но согласно [5], может быть модифицирован для восстановления поверхностей с границами. Предположим, что набор входных точек имеет плотность ρ (это означает, что любая сфера с центром, лежащим на M радиуса ρ содержит, по крайней мере, одну точку входного набора) и максимальный уровень шума δ (т.е. положение точки входных данных может отклоняться от поверхности не более, чем на δ). Очевидно, что если $d(p) > \rho + \delta$, где $d(p)$ - проекция точки на ближайшую касательную плоскость, то такая точка не может лежать на поверхности M . Таким образом, мы приходим к более общему правилу вычисления $F(p)$.

$$F(p) = \begin{cases} (p - o_i)n_i & \text{если } o_i - ((p - o_i)n_i)n_i \leq \rho + \delta \\ \text{не определено в противном случае} \end{cases}$$

6. ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

В таблице 2 приведены результаты измерения производительности алгоритма для сетки размером 256x256x256. Для тестирования использовалась рабочая станция с процессором Intel Core i7 975 Extreme 3,33ГГц и видеоадаптером ATI Radeon HD5870 с 1Гб видеопамяти.

Модель	Кол-во точек	Кол-во обработанных ячеек сетки	Время, мс			
			Построение сетки	Построение F	Триангуляция	Общее
Sphere	65536	52502	12	6	3	21
Sphere	131072	90208	26	11	11	48
Bunny	353272	178714	40	140	20	200

Таблица 2. Производительность

7. ЗАКЛЮЧЕНИЕ

В рамках данной работы был разработан параллельный алгоритм восстановления поверхности из облака точек для архитектур с массовым параллелизмом. Были разработаны и реализованы эффективные структуры хранения данных, а так же предложена архитектура системы для реализации в рамках технологии гетерогенных вычислений OpenCL. В дальнейшем работу предполагается использовать в качестве основы для построения алгоритмов аппроксимации восстановленной поверхности функциями более высоких порядков. Полученные показатели времени работы свидетельствуют о том, что алгоритм может быть использован для восстановления поверхности в реальном времени.

8. ССЫЛКИ

- [1] Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., And Silva, C. T. 2001. Point set surfaces. In Proceedings of IEEE Visualization'01, 21–28.
- [2] Amenta, N., Bern, M., and Kamvysselis, M. 1998. A new Voronoi-based surface reconstruction algorithm. In Proceedings of SIG-GRAPH'98, 415–421.
- [3] Blesl, G. Vector Models for Data-Parallel Computing. 1990. The MIT Press
- [4] Harris, M., Sengupta, S., and Owens, J. D. 2007. Parallel prefix sum (scan) with CUDA. In GPU Gems 3, H. Nguyen, Ed. Addison Wesley, Upper Saddle River.
- [5] Hoppe, H., Deroose, T., Duchamp, T., McDonald, J., and Stuetzle, W. 1992. Surface reconstruction from unorganized points. In Proceedings of SIGGRAPH'92, 71–78.
- [6] Kazhdan, M., Bolitho, M., and Hoppe, H. 2006. Poisson surface reconstruction. In Proceedings of SGP'06, 61–70
- [7] Kruskal, J., On the shortest spanning subtree of a graph and the traveling salesman problem. // Proc. AMS. 1956. Vol 7, No. 1. C. 48–50
- [8] Kun, K., Minmin, G., Xin, H., Baining, G., Highly parallel surface reconstruction. Microsoft research Asia.
- [9] Lorensen, W. E., And Cline, H. E. 1987. Marching cubes: A highresolution 3d surface construction algorithm. In Proceedings of SIG-GRAPH'87, 163–169.
- [10] Muraki, S. Volumetric shape description of range data using "blobby model". *Computer Graphics (SIGGRAPH '91 Proceedings)*, 25(4):227–235, July 1991.

- [11] Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., And Seidel, H.-P. 2004. Multi-level partition of unity implicits. *ACM Trans. Graph.* 22, 3, 463–470.
- [12] Smith, K., Eigenvalues of a symmetric 3x3 matrix. *Commun. ACM* 4(4): 168 (1961)
- [13] Taubin, G. Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations, with applications to edge and range image segmentation. Technical Report LEMS-66, Division of Engineering, Brown University, 1990
- [14] Turk, G., and O'Brien, J. F. 2002. Modelling with implicit surfaces that interpolate. *ACM Trans. Graph.* 21, 4, 855–873.
- [15] Vemuri, B. *Representation and Recognition of Objects From Dense Range Maps*. PhD thesis, Department of Electrical and Computer Engineering, University of Texas at Austin, 1987.
- [16] Vemuri, B., Mitiche, A., and Aggarwal, J. Curvaturebased representation of objects from range data. *Image and Vision Computing*, 4(2):107–114, 1986.
- [17] Yusov, E., Turlapov, V. 2008. JPEG2000-based compressed multiresolution model for real-time large scale terrain visualization. В кн.: Conference Proceedings of the 18th international Conference on Computer Graphics and Vision “GraphiCon’2007”, 2007. С. 164—171

9. РЕЗУЛЬТАТЫ РАБОТЫ АЛГОРИТМА

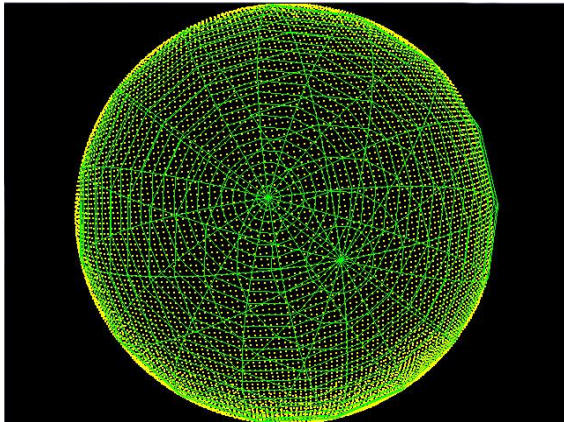


Рисунок 1. Облако точек на сфере

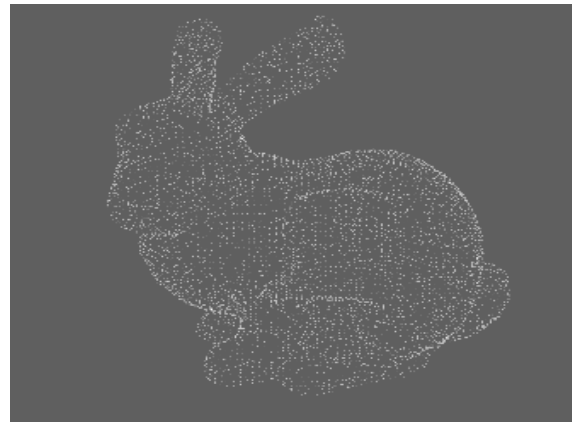


Рисунок 3. Облако точек (Стэнфордский кролик)

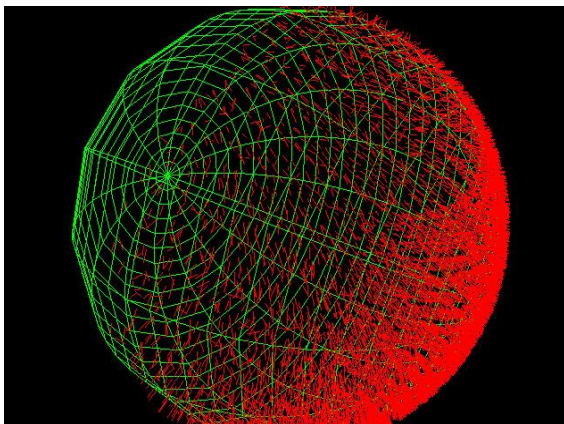


Рисунок 2. Восстановленное поле нормалей



Рисунок 4. Трингуляция с освещением

Моделирование монитора кругового обзора.

И.В. Валиев, А.Г. Волобой

Институт прикладной математики им. М.В.Келдыша РАН, Москва

Аннотация

В статье рассматриваются система моделирования монитора кругового обзора. Монитор кругового обзора – это отображение на дисплее (вид сверху) положения транспортного средства вместе с окружающими объектами во время сложных маневров, таких как парковка автомобиля, лоцманская проводка судна в порту или руление авиалайнера в аэропорту. Система моделирования позволяет разработать такой монитор, проверить и настроить характеристики используемых камер, а также их положение в пространстве.

Ключевые слова: визуализация, навигация, расширенная реальность, трассировка лучей.

1. ВВЕДЕНИЕ

Одним из важных применений компьютерной графики являются системы, помогающие управлять транспортными средствами или другими сложными техническими механизмами. В первую очередь к таким системам относятся космические тренажеры [1, 2], а также имитационные комплексы [3], которые позволяют проводить предварительную тренировку и отработку элементов управления сложными механизмами. Отличительной особенностью тренажерных комплексов является то, что они не ставятся на реальные установки и не работают в реальных условиях, т.к. это часто слишком дорого и связано с опасностью для человека.

С развитием быстродействия процессоров и портативных видеокамер появились программные системы, используемые непосредственно в навигации транспортных средств [4, 5]. Эти системы позволяют автоматически определить положение транспортного средства или аппарата в пространстве, отслеживать окружающую обстановку. На основе этих данных принимается решение о дальнейшем движении (навигации). Однако большие вычислительные затраты, связанные с реализацией алгоритмов, заложенных в этих системах, иногда не позволяют добиться необходимой скорости в реальном времени, что отражается на их надежности.

Удешевление аппаратных компонент навигационных систем, в частности портативных камер, привело к тому, что сейчас такие системы могут быть установлены не только на сложных космических аппаратах или судах, а также на обычных, массовых автомобилях. Водитель автомобиля имеет ограниченное поле зрения, что доставляет неудобство в процессе парковки автомобиля. Для облегчения парковки производители автомобилей применяют различные способы, такие как парковочные радары, камеры заднего вида, линзы Френеля. Одной из современных систем являются мониторы кругового обзора [6, 7]. Система состоит из 4-х широкоугольных видеокамер [8], расположенных спереди, сзади и по бокам автомобиля. Они расположены наклонно, так чтобы снимать сектора земли вокруг автомобиля. Высота

крепления камеры над землей, ее положение и ориентация относительно автомобиля известны. Это позволяет преобразовать эти снимки к изображениям, как бы снятым ортогографической камерой ориентированной вертикально вниз и направленной на автомобиль.

2. МОНИТОР КРУГОВОГО ОБЗОРА

Задачей монитора кругового обзора является преобразование видео потока, полученного с камер, в изображение автомобиля вместе с окружением с высоты «птичьего полета». На рис. 1 представлено изображение автомобиля с секторами видимости камер.

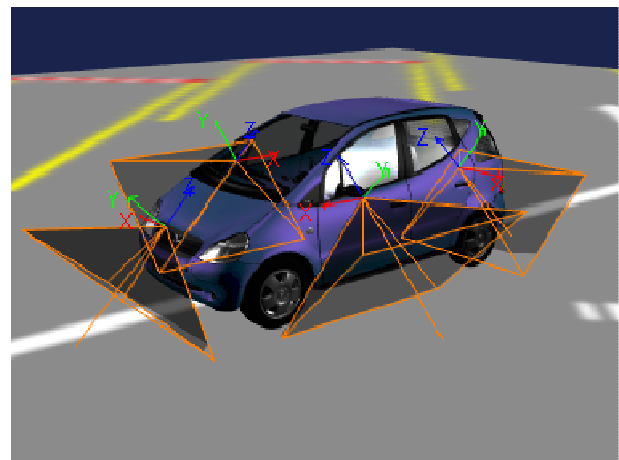


Рис. 1. Расположение камер относительно автомобиля.

Если автомобиль расположен на плоскости с шахматной текстурой, то итоговое изображение автомобиля на мониторе получится таким, как показано на рис. 2.

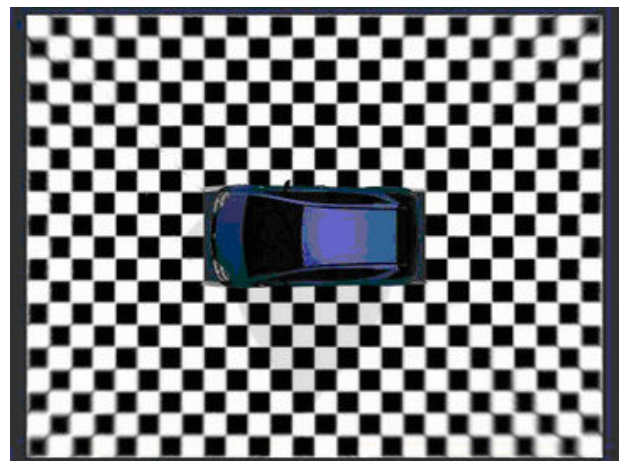


Рис. 2. Общий ортогографический вид сверху.

В процессе формирования итогового изображения решается задача преобразования исходного изображения, полученного с камеры, (рис.3а) в изображение этого участка в ортографическом виде сверху (рис. 3б).

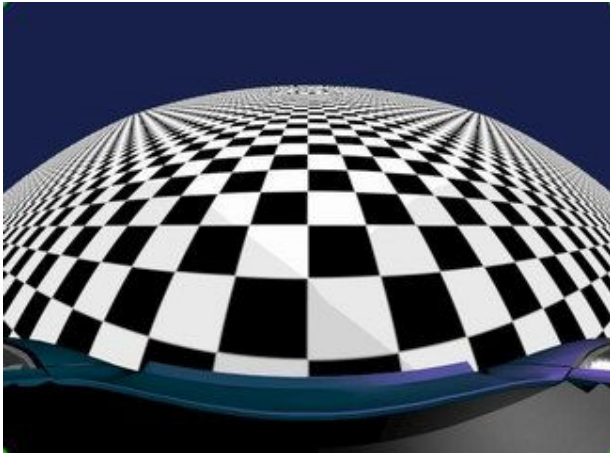


Рис. 3а. Снимок передней камеры.

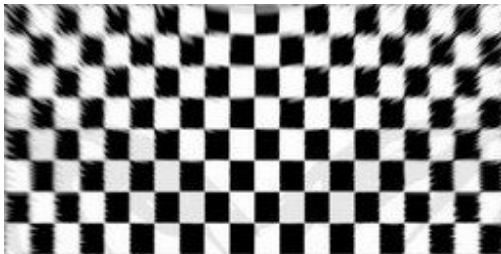


Рис. 3б. Снимок передней камеры, преобразованный в ортографический вид сверху.

Одним из начальных этапов разработки монитора кругового обзора является компьютерное моделирование системы с целью подбора оптимальных параметров камер, их расположения, и исследования влияния различных оптических эффектов на качество финального изображения.

3. ОСОБЕННОСТИ ЗАДАНИЯ СЦЕНЫ ДЛЯ МОДЕЛИРОВАНИЯ МОНИТОРА КРУГОВОГО ОБЗОРА

Нами была предложена следующая структура сцены для моделирования монитора кругового обзора.

Сцена состоит из плоскости, представляющей собой землю, и модели автомобиля. Также сцена может содержать любые другие объекты, представляющие препятствия, опасные для движения автомобиля (другой автомобиль, столб и т.д.). Отличительной особенностью сцены является наличие пяти камер (четыре камеры соответствуют камерам, закрепленным на автомобиле, а пятая – камера, направленная сверху на автомобиль), а также специального объекта – набора плоских секторов, копланарных с плоскостью земли. Каждый сектор соответствует автомобильной камере, снимающей участок земли (рис. 4). Плоскость земли является объектом съемки. Относительно нее может перемещаться автомобиль с прикрепленными камерами и специальным объектом – набором секторов.

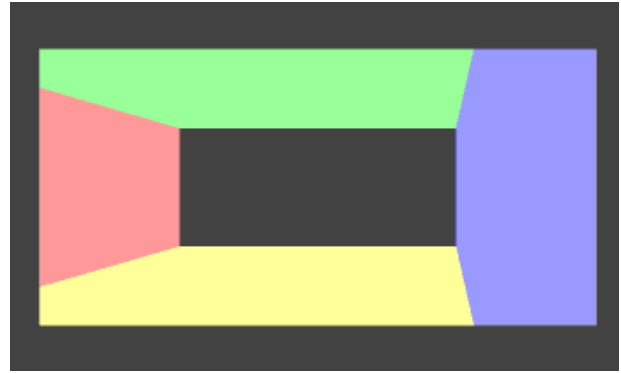


Рис. 4. Сектора кругового обзора для представления результирующих изображений

Дополнительные требования должны быть учтены при реализации моделей камер, закрепленных на автомобиле. Традиционные системы визуализации 3-хмерных сцен работают с точечными камерами, отображающими сцену в соответствии с законами перспективной проекции, и заведомо имеющими угол обзора меньше 180 градусов. Однако для построения монитора кругового обзора предполагается применять широкоугольные камеры, с углом обзора больше 180 градусов, с так называемыми объективами «рыбий глаз». Точечная камера характеризуется одной точкой, через которую осуществляется проекция, координаты которой и определяют положение камеры. Эти координаты не зависят от угла проекции. В широкоугольных объективах это не так. Положение точки проекции на оптической оси камеры зависит от угла проекции (рис. 5). Еще одно требование – должны поддерживаться камеры с произвольным положением оптической оси относительно растра изображения.

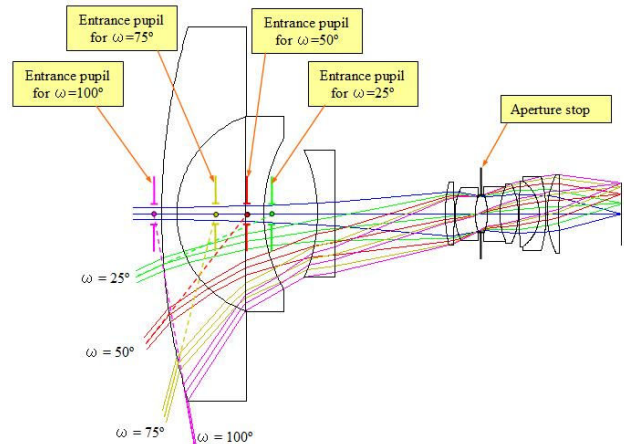


Рис. 5. Положение точек проекции на оптической оси для типичной широкоугольной камеры.

4. ПОЛУЧЕНИЕ ИТОГОВОГО ИЗОБРАЖЕНИЯ

Для получения итогового изображения мы сначала воспроизводим кадры, полученные камерами, прикрепленными к автомобилю (процесс съемки). Затем преобразуем их в текстуру соответствующего сектора путем обратной проекции изображения на плоскость земли. В момент съемки набор секторов прячется, а в момент показа

результата – показывается с восстановленной текстурой. Результат показывается при помощи ортографической камеры расположенной над автомобилем.

Исходные изображения (процесс съемки) получается при помощи обратной трассировки лучей из камеры. Для каждого пикселя исходного изображения вычисляется угол проекции в соответствии с функцией проецирования и луч из камеры направляется в вычисленном направлении. Если есть пересечение с объектами сцены, то вычисляется освещенность и видимая яркость соответствующей точки поверхности, в ином случае вычисляется яркость фона. Используется именно яркостное изображение без преобразования в цвет, представимый на экране монитора.

Дальше для каждого сектора плоскости кругового обзора вычисляется текстурное изображение из исходного изображения камеры, направленной в сторону данного сектора, по заданным координатам вершин текстурного прямоугольника, и известным параметрам соответствующей камеры кругового обзора. Алгоритм вычисления текстуры из изображения камеры простой: для каждого пикселя текстуры вычисляется его положение на плоскости сектора. Далее пускается луч из камеры в эту точку и определяется соответствующая точка в прямоугольнике изображения камеры. Точка находится между центрами ближайших четырех пикселей. Значения текстурного пикселя находится интерполяцией значений этих четырех пикселей с учетом расстояния до них.

Аналогичный алгоритм используется для нахождения изображения виртуальной камеры – сначала находится пересечение луча вертикальной камеры, соответствующего ее пикселю, с плоскостью сектора, а затем в ту же точку пускается луч из камеры, соответствующей реальному изображению.

Используется яркостная текстура, все прочие атрибуты поверхности сектора обнуляются, чтобы избежать дополнительного освещения источниками света. При визуализации результирующего кругового обзора плоскость земли и возможные дополнительные объекты за исключением самого автомобиля прячутся. В итоге показывается автомобиль и сектора кругового обзора с посчитанными текстурами.

Для учета параметров реальной камеры, например, ее дисторсию, используется функция проецирования. Функция задается через таблицу зависимости расстояния между пикселем и центром проекции на изображении от угла проекции. Расстояние пикселя от центра проекции может быть выражено в любых единицах. Для применения функции к конкретному окну изображения камеры оно масштабируется с учетом размеров изображения и угла зрения камеры для этого изображения. Функция предполагается монотонно возрастающей, что позволяет использовать её как для прямого, так и обратного преобразований. Функция проецирования для реальной камеры были измерены.

5. ОЦЕНКА ВЛИЯНИЯ НЕТОЧНОСТИ ЗАДАНИЯ ПОЛОЖЕНИЯ КАМЕРЫ И ЕЕ ХАРАКТЕРИСТИК

При моделировании монитора кругового обзора положение и ориентация камер известны абсолютно точно. Поэтому итоговое изображение секторов кругового обзора совпадает с

изображением плоскости земли, снятым той же камерой. Единственным отличием являются дефекты изображения, возникающих при преобразовании исходных изображений в текстуры. Дефекты возрастают с возрастанием расстояния от камер до земли. Наиболее качественно представлены участки земли около автомобиля, наименее качественно – удаленные участки, что хорошо заметно на рис. 3б. При этом швы на стыке секторов совершенно не заметны.

В реальном мониторе кругового обзора положение и ориентация камер известны с некоторой погрешностью, и изображение секторов может отличаться от изображения земли из-за этого. Для оценки влияния погрешности задания параметров камеры, было предложено использовать смещенную камеру, чьи снимки используются для восстановления текстуры, но с параметрами базовой несмещенной камеры.

Как видно из текстур на рис. 6, восстановленной для базовой и смещенной камер (смещенная камера повернута на 1^0 по всем трем осям), такие малые погрешности задания камеры слабо влияют на результат. Это говорит о надежности такой системы в реальной жизни.

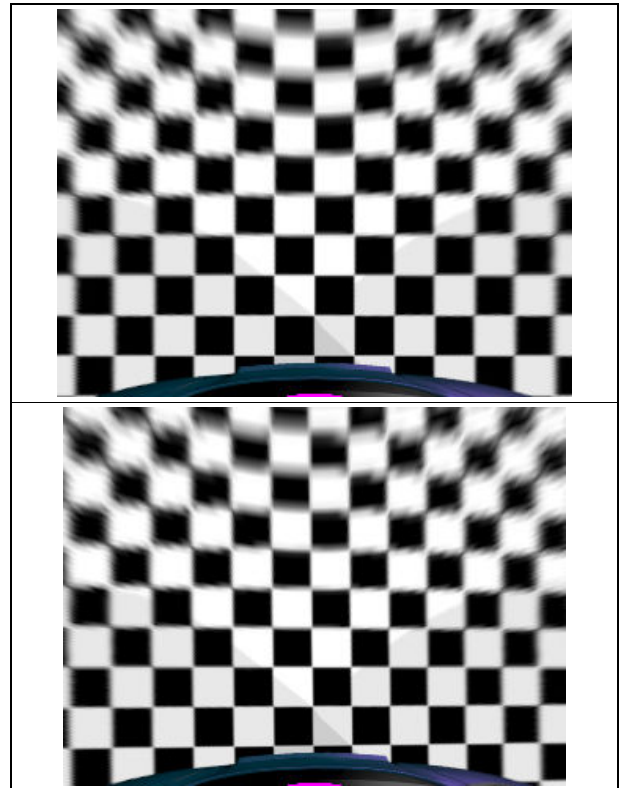


Рис. 6. Вверху текстура, восстановленная из изображения базовой камеры, внизу – из изображения камеры, повернутой на 1^0 по всем трем углам относительно базовой камеры.

Изображения, представленные выше, были синтезированы для камеры с идеальной частотно-контрастной характеристикой (ЧКХ). Для оценки влияния ЧКХ камеры было предложено производить фильтрацию синтезированных изображений. Фильтрация изображения производится путем двумерного преобразования Фурье. К полученному образу пространственных частот применяется заданная функция передачи модуляции. Затем производится обратное

преобразование Фурье. На рис. 7 представлены текстуры кругового обзора, восстановленные для камеры с идеальными ЧКХ и для камеры с реальными, заранее заданными характеристиками.

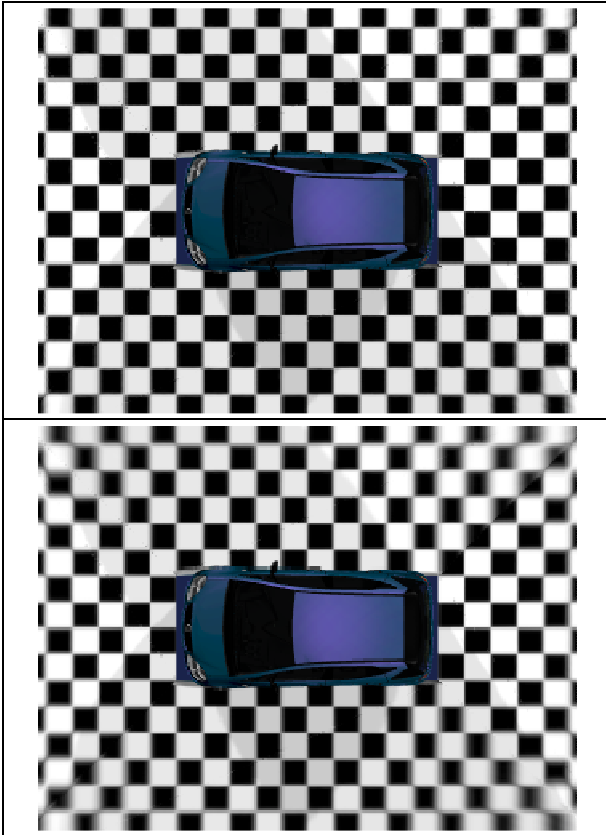


Рис. 7. Вверху показан круговой обзор, полученный при помощи камер с идеальной ЧКХ, внизу – с заданной ЧКХ

Таким образом, при помощи моделирования монитора кругового обзора возможно оценить применимость той и или иной реальной камеры, влияние расположения камер на итоговое изображение.

6. РЕЗУЛЬТАТЫ

Система моделирования монитора кругового обзора была разработана и встроена в программный комплекс расчета освещенности Inspire2 и CAD-систему CATIA (Dassault systems). Время генерации итогового изображения зависит от размеров текстур и характеристик камер. В реальной работе это время колеблется от нескольких секунд до нескольких минут.

С помощью этой системы возможно разработать и промоделировать оптические компоненты монитора кругового обзора, который значительно упрощает проведение сложных маневров, таких как парковка автомобиля, лоцманская проводка судов в порту, рулежка самолетов в аэропортах.

Работа поддержана грантом Президента РФ «Ведущие научные школы» НШ-8129.2010.9, а также фирмой Integra Inc. (Япония).

Электронный вариант статьи с цветными иллюстрациями размещен на сайте http://www.keldysh.ru/pages/cgraph/publications/cgd_publ.htm.

7. ЛИТЕРАТУРА

[1] Б.С. Долговесов, Б.С. Мазурок, М.У. Шевцов, Б.Б. Морозов, А.В. Рухлинский, А.В. Пекарский, В.А. Наумов, В.М. Фомичев. 3D графика реального времени: от тренажеров до виртуальных студий. Труды конференции Графикон'2005. Новосибирск, 2005. стр. 44-47.

[2] А.И. Масалкин. Опыт использования систем компьютерной графики в тренажерах ПКА. Сборник докладов Международной научной конференции, посвященной 80-летию со дня рождения академика В. А. Мельникова, Москва, 2009. стр. 96-98.

[3] М.В. Михайлюк, М.А. Торгашев, И.А. Хураськин. Общая структура программного обеспечения тренажерного комплекса. Сборник докладов Международной научной конференции, посвященной 80-летию со дня рождения академика В. А. Мельникова, Москва, 2009. стр. 68-71.

[4] В.А. Бобков, Ю.С. Борисов, С.В. Мельман, Ю.И. Роншин. Моделирующий комплекс для исследования методов управления движением автономного подводного аппарата. Труды конференции Графикон'2006. Новосибирск, 2006. стр. 279-283.

[5] Ю.С. Борисов. Решение задачи навигации подводного аппарата по последовательности изображений. Труды конференции Графикон'2006. Новосибирск. стр. 284-288.

[6] NISSAN. Around view monitor.

<http://www.nissan-global.com/EN/TECHNOLOGY/INTRODUCTION/DETAILS/AVM/>

[7] Nissan releases details about "Around View Monitor"

<http://www.autoblog.com/2007/10/12/nissan-releases-details-about-around-view-monitor/>

[8] Sony CMOS camera for Nissan around view monitor. http://www.fareastgizmos.com/transport/sony_cmos_camera_for_nissan_around_view_monitor.php

Abstract

Around view monitor simulation system is presented in the article. Around view monitor is a display of top view image of a vehicle together with around environment. This display simplifies such advanced maneuvers as car parking, ship and boat piloting in harborage, aircraft taxiing in airport. The simulation system allows to elaborate monitor, to check and verify its optical components.

Keywords: *visualization, navigation, augmented reality, ray tracing.*

Authors:

Ildar V. Valiev, researcher, Keldysh Institute for Applied Mathematics RAS.

Alexey G. Voloboy, PhD, senior researcher, Keldysh Institute for Applied Mathematics RAS.

E-mail: voloboy@gin.keldysh.ru

Методы быстрого формирования изображения в параллельной системе постобработки результатов научных вычислений ScientificView

А.Л. Потехин

Институт теоретической и математической физики

РФЯЦ-ВНИИЭФ, Саров, Россия

Potekhin @ vniief.ru

Аннотация

В работе описываются некоторые особенности реализации параллельной системы постобработки ScientificView, созданной в Институте теоретической и математической физики РФЯЦ-ВНИИЭФ. Система предназначена для графического и числового анализа результатов моделирования физических процессов сеточными (на разностных сетках регулярного и нерегулярного типов) и бессеточными методами.

Кратко описаны возможности по обработке данных и схема построения системы. Более полно описаны реализованные в системе алгоритмы, направленные на ускорение процесса формирования изображения.

Ключевые слова: научная визуализация, параллельные системы постобработки, методы формирования изображения.

1. ВВЕДЕНИЕ

Мощные многопроцессорные ЭВМ позволяют более детально описывать физические процессы за счет применения более точных моделей и более подробного описания геометрии задачи пространственной сеткой. Для достижения требуемого качества моделирования сложных физических процессов часто используются сетки, состоящие из сотен миллионов счетных ячеек и обрабатываемые на тысячах процессоров. Оценка качества расчетов на таких сетках требует развитых средств анализа и постобработки, в частности, систем визуализации, способных быстро обрабатывать большой объем данных. Добиться этого с помощью однопроцессорных программ затруднительно по причине ограниченности ресурсов ПЭВМ, поэтому во всем мире создаются системы визуализации, способные работать в параллельном режиме. Среди доступных в открытых кодах зарубежных систем в первую очередь нужно отметить ParaView[1] и VisIt[2], построенные на библиотеке VTK[3]. Отечественные специалисты также решали задачу графической обработки большого объема данных, например, при создании модуля визуализации FlowVision[4] и системы визуализации ScientificVR[5].

Для постобработки результатов, полученных при численном моделировании физических процессов по программам математического отделения ИТМФ РФЯЦ-ВНИИЭФ, разрабатывается система параллельной постобработки ScientificView[6]. Система предназначена для отображения, фильтрации и числового анализа данных, определенных на разностных сетках регулярного и нерегулярного типов,

двумерных и трехмерных, а также данных, полученных при моделировании бессеточными методами, с помощью молекул, частиц, кластеров. Ядро системы позволяет быстро создавать новые методы анализа, в настоящий момент на его основе реализовано более двадцати алгоритмов графической обработки данных. Часть из них - «классические» для систем визуализации: построение сечений, векторных полей, линий профиля, изоповерхностей и т.д. (рисунок 1).

Некоторые алгоритмы направлены на обработку особенностей представления данных специализированными решателями. Они предназначены для обработки смешанных ячеек, содержащих несколько веществ; адаптивно-встраиваемых сеток, применяемых для более точного моделирования и т.д. К таким алгоритмам относятся, например, фильтры «Анализ топологии» и «Выделение границы раздела веществ» (рисунок 2). Кроме средств графического анализа, ScientificView предоставляет возможности по числовой обработке данных, в частности, табличный просмотр и редактирование массивов, списков и структур данных (рисунок 3).

2. СХЕМА ПОСТРОЕНИЯ СИСТЕМЫ

ScientificView построен по схеме клиент-сервер (рисунок 4).

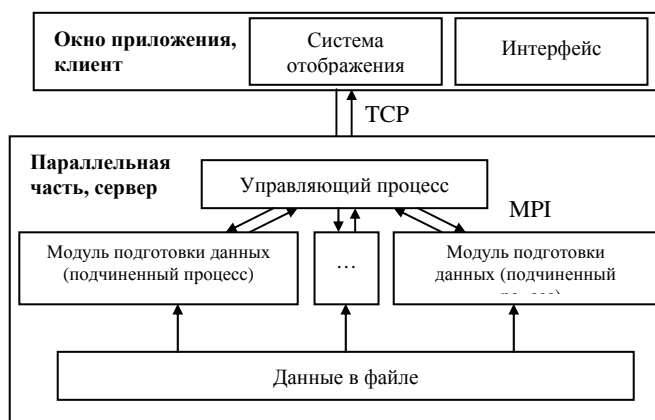


Рис 4: Схема построения ScientificView.

Серверная часть работает в параллельном режиме под Windows или Linux на серверах с распределенной памятью и содержит в себе методы чтения, предобработки и фильтрации данных. Распараллеливание процедур чтения и фильтрации позволило снизить время обработки данных до комфортных величин – в случае задач с числом ячеек около 500 млн. и использовании 64 процессоров в серверной части время работы данных процедур составляет 40-80 секунд.

Клиентская часть ScientificView – Windows приложение, содержащее интерфейс с пользователем и систему формирования изображения.

Такое распределение задач между клиентской и серверной частями ScientificView позволило быстро провести первичную реализацию системы, но привело к дисбалансу нагрузки – клиентская часть должна одним процессором обеспечить приемлемую скорость визуализации данных, полученных на большом числе процессоров сервера.

3. НЕКОТОРЫЕ КЛАССИЧЕСКИЕ РЕШЕНИЯ ЗАДАЧИ БЫСТРОГО ФОРМИРОВАНИЯ ИЗОБРАЖЕНИЯ

Для ускорения процесса формирования изображения системами визуализации в первую очередь используются классические решения, основанные на оптимизации обработки данных графическими библиотеками и видеокартами. Среди простейших решений – стрипование многоугольников, использование списков отображения, массивов вершин. К более сложным можно отнести различные типы шейдеров и языки программирования видеокарт, такие как CUDA.

В совокупности эти средства способны ускорить процесс формирования изображения в десятки раз и на этапе обработки геометрии и при растеризации. Однако их использование налагает ряд дополнительных ограничений. Для некоторых решений необходима видеокарта определенного производителя или видеокарта как таковая. Использование других требует привести отображаемые данные к специальному виду, часто неудобному для дополнительной, неграфической обработки. В таких случаях программа будет расходовать двойную оперативную память – для хранения данных в оптимальном для рендеринга виде и для хранения данных в виде удобном для дополнительной обработки.

Система постобработки ScientificView разрабатывалась как кроссплатформенное средство быстрой обработки данных большого объема, поэтому описанные выше ограничения не позволяют в полной мере воспользоваться классическими решениями. В системе, наряду со стрипованием многоугольников и использованием списков отображения используются собственные алгоритмы, реализованные для быстрого формирования изображения.

4. РЕШЕНИЯ, РЕАЛИЗОВАННЫЕ В SCIENTIFICVIEW

Все реализованные в ScientificView алгоритмы, направленные на ускорение процесса формирования изображения, можно условно разделить на два типа:

- алгоритмы, оценивающие текущее положение, размер и степень видимости объектов и их частей. Не видимые объекты не отображаются;
- алгоритмы, определяющие в зависимости от ситуации рекомендуемую степень детализации объектов. При необходимости формируют упрощенные модели рассматриваемых объектов.

Первый алгоритм, реализованный в ScientificView для быстрого формирования изображения, используется при

отображении замкнутых трехмерных тел: трехмерных разностных сеток и результатов их фильтрации некоторыми алгоритмами, также формирующими замкнутые трехмерные объекты. У таких объектов видимыми будут грани, для которых угол между внешней нормалью и линией взгляда (линией, ортогональной плоскости экрана для текущего ракурса просмотра) меньше 90 градусов – рисунок 5.

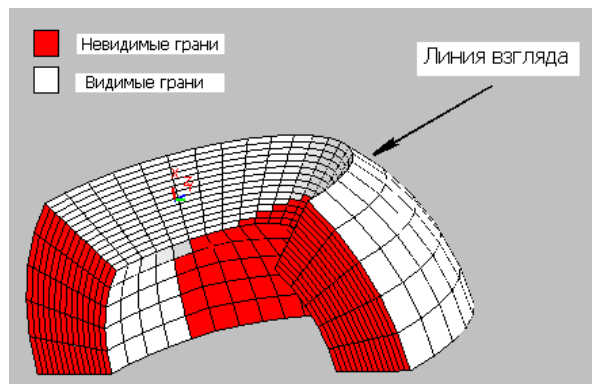


Рис 5: Определение видимости граней на основе нормалей и линии взгляда

Практическое использование ScientificView показало, что данный алгоритм в среднем уменьшает число отображаемых граней на 50%, за счет чего примерно в 2 раза возрастает скорость отображения.

Число отображаемых многоугольников также можно уменьшить при объединении граней ячеек по тому или иному критерию. Критерии могут быть различными, но главное – они должны максимально сохранить важные особенности рассматриваемого объекта и ускорить работу за счет более грубого представления второстепенных фрагментов. В ScientificView объединение граней осуществляется с учетом значений текущей величины, и, на практике, до 10 раз уменьшает число отображаемых многоугольников без существенной потери информативности изображения – рисунок 6.

Объединение проводится только для граней ячеек регулярных математических областей на основе их регулярной топологии. Обработка таким алгоритмом результатов фильтрации со сложной топологической структурой невозможна.

Кроме того, эксплуатация системы показала, что самый большой дискомфорт вызывают задержки, возникающие при попытках пользователя повернуть, подвинуть или масштабировать отображаемую сцену. В данном случае важно максимально быстро получить конечный результат, возможно путем частичной потери информативности изображения. Многие системы визуализации (например, ParaView[1]) используют для отображения во время поворотов и смещений грубую, упрощенную модель объектов. В ScientificView также реализован алгоритм построения упрощенных моделей. Он основан на аппроксимации вершин многоугольников узлами прямоугольной сетки, построенной в параллелепипеде, ограничивающем рассматриваемый объект. Алгоритм работает быстро, применим к двумерным или трехмерным регулярным и нерегулярным сеточным данным, и позволяет управлять степенью детализации – рисунок 7.

Использование алгоритма позволяет добиться скорости вывода 10-20 кадров в секунду даже при перемещениях сложных объектов. Однако после остановки перемещения система должна провести формирование полноценной картинки, что может привести к задержкам в 5 и более секунд. Для уменьшения задержек был разработан еще один алгоритм, основанный на режиме «обратной связи» (feedback) OpenGL. Данный режим предназначен для получения пиксельных размеров образов на экране. Если размер образа мал (например, менее 50 пикселей по вертикали или по горизонтали) можно применять для отображения объекта результат работы алгоритма построения упрощенных моделей. В зависимости от размера образа используется разная степень детализации упрощенного объекта – меньше образ – грубее детализация. Если объект имеет образ нулевого размера, то он находится вне зоны видимости и его не нужно отображать вообще. К сожалению, время работы в режиме обратной связи сопоставимо со временем работы в режиме формирования изображения, поэтому анализ пиксельных размеров сложных объектов не дает выигрыша в скорости по сравнению с полным отображением этих объектов. Поэтому в ScientificView анализируется не сам объект, а минимальный включающий его ограничивающий параллелепипед с осями, параллельными осям координат. Параллелепипед обрабатывается быстро и позволяет получить грубую оценку видимости самого объекта.

На рисунке 8, слева, представлен пример обработки параллелепипедов для объекта, состоящего из 150 деталей. В данном ракурсе просмотр только для трех деталей огрубление недопустимо. Они изображены на рисунке с габаритной коробкой фиолетового цвета. Остальные можно отображать с разной степенью детализации: для объектов с зелеными габаритными коробками применена высокая степень огрубления, с синими – средняя степень, с желтыми – низкая степень. На рисунке 8, справа, представлено изображение, полученное с использованием алгоритма на основе «обратной связи». При этом заметить потерю качества глаз не способен, а скорость формирования изображения увеличилась в 5 раз.

5. ЗАКЛЮЧЕНИЕ

Эксплуатация ScientificView показала, что совместное применение описанных алгоритмов позволяет обеспечить при вращениях, перемещениях, масштабировании сцены отображение не менее 10 кадров в секунду даже для сложных объектов. Отображение сложных сцен без потери информативности требует не более 2-3 секунд. Пример работы системы вывода при отображении сцены с рисунка 8, представлен на гистограмме 9. На гистограмме отражено время формирования кадра (в секундах) во время вращений («быстрый» режим) и при формировании полноценной картинки («статический» режим) при использовании разного набора алгоритмов.

Однако с дальнейшим увеличением числа многоугольников в отображаемых объектах, либо с появлением принципиально новых моделей представления данных (например, воксельной и стереографики, мультиэкранов) возможны некомфортные для пользователя задержки. В этом случае возможно применение параллельных схем формирования изображения, работа над которыми уже ведется.

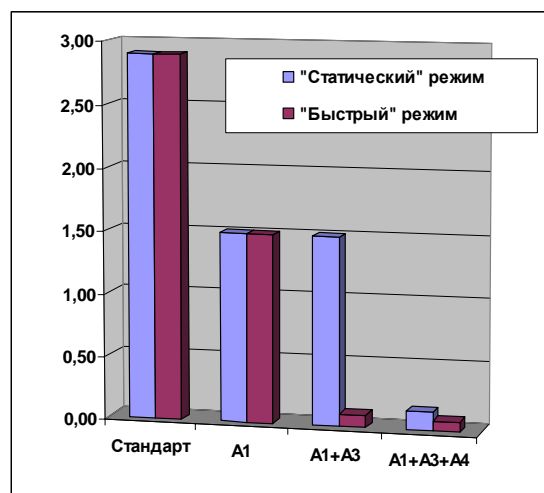


Рис 9: Зависимость времени отображения от набора использованных алгоритмов

6. БИБЛИОГРАФИЯ

- [1] Henderson A., Ahren J., etc. "The ParaView guide", Kitware, Inc. 2004
- [2] "VisIt User's Manual, version 1.5" Lawrence Livermore National Laboratory, October 2005
<https://wci.llnl.gov/codes/visit/1.5/VisItUsersManual1.5.pdf>
- [3] VTK Visualization Toolkit software, <http://www.vtk.org/>
- [4] Программный комплекс FlowVision, компания TECNIS
<http://www.flowvision.ru/>
- [5] Бабаев Д.Б. "Функциональные возможности визуализатора Scientific-VR", *Супервычисления и математическое моделирование. Международный семинар, Саров, 2003, стр. 17-19*
- [6] А. Потехин, В. Тарасов, С. Фирсов, И. Логинов, В. Никитин, М. Кузнецов, Н. Попова, А. Деманова и Ю. Козачек. *ScientificView - система параллельной постобработки результатов, полученных при численном моделировании физических процессов. Труды международной конференции "ГрафиКон' 2008". Москва, 2008.*

Об авторе

Потехин Антон Львович, начальник сектора лаборатории общего сервиса Института теоретической и математической физики, Российского федерального ядерного центра, Всероссийского научно-исследовательского института экспериментальной физики. Почта: potekhin@vniief.ru

Fast rendering methods in ScientificView - parallel system for postprocessing scientific modeling

Abstract

This paper is told about some features of realization of parallel postprocessing system, developed in Institute theoretical and mathematical physics RFYAC-VNIIEF. The system is intended

for visual and numerical analysis of result modeling of the physical processes on regular, non-regular grids and non-grids methods.

General features and building scheme of system are shortly described in paper. The algorithms realised in system directed on acceleration of rendering process are more full described.

Keywords: *scientific visualization, parallel post-processing systems, rendering methods.*

About the author

Anton L. Potekhin, leader of visualisation team in mathematical department ITMF, RFYAC-VNIIEF. E-mail: potekhin@vniief.ru

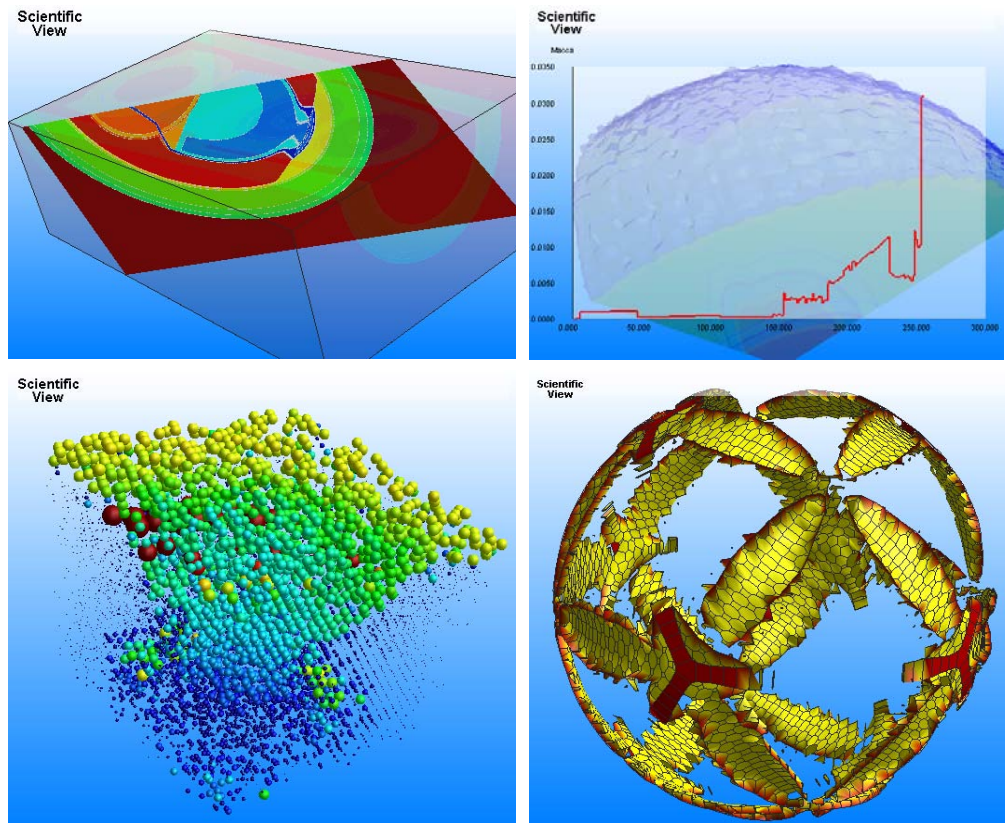


Рисунок 1: Результат работы фильтров «Сечение плоскостью» и «Линия профиля», сверху фильтров «Векторное поле» и «Изоповерхность», внизу

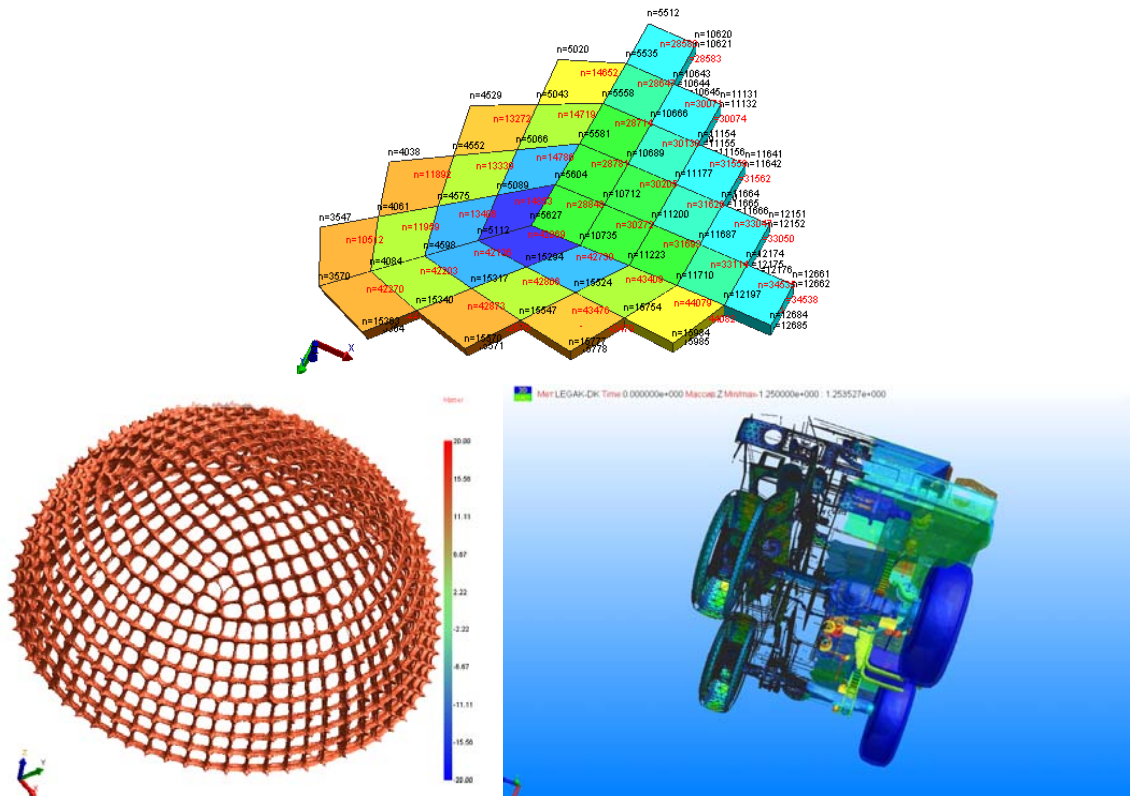


Рисунок 2: Результаты работы фильтров «Анализ топологии», сверху, «Выделение границы раздела веществ», внизу (слева - выделение рамки держателя пористого материала, справа – обработка обсчета перспективного броневика)

Секция 6. Вопросы визуализации

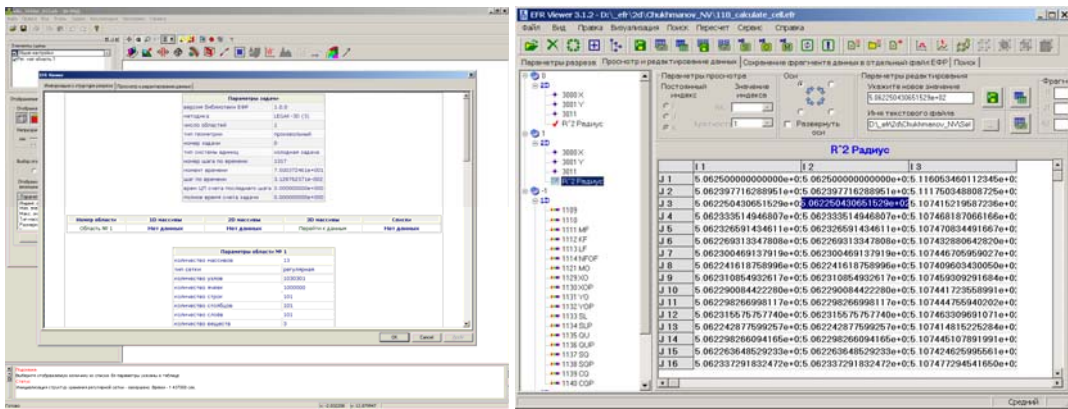


Рисунок 3: Некоторые средства числового анализа – выдача интегральных характеристик, табличный просмотр и редактирование

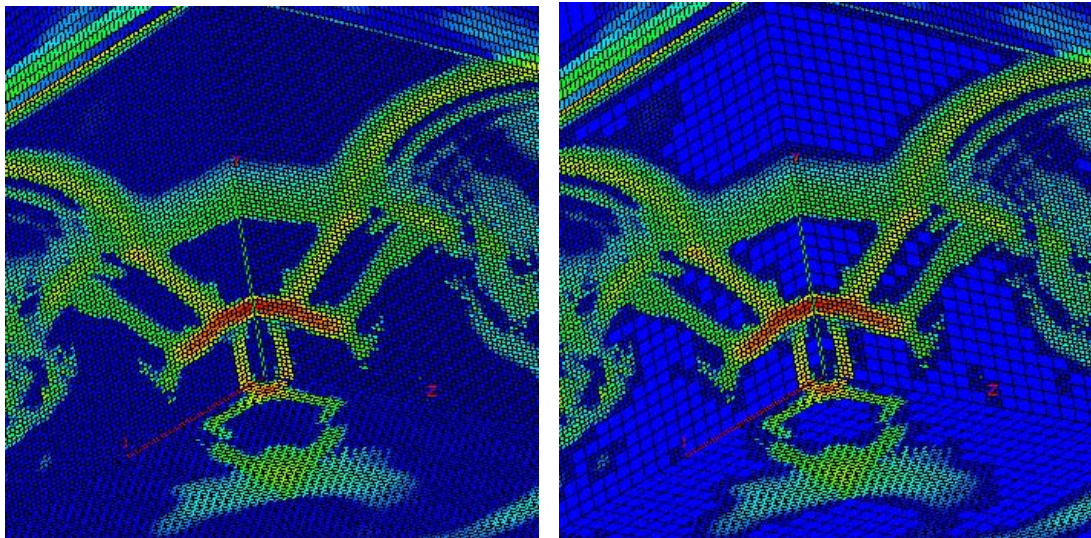
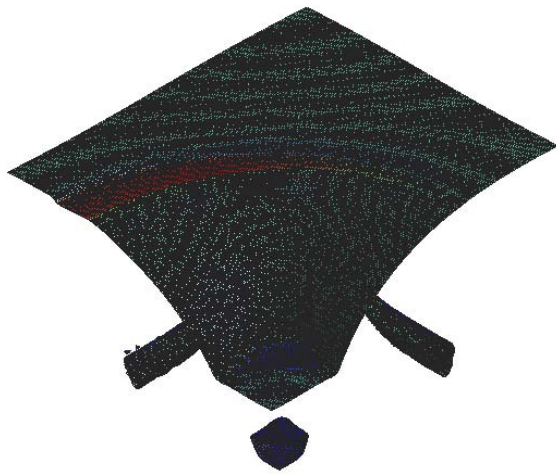
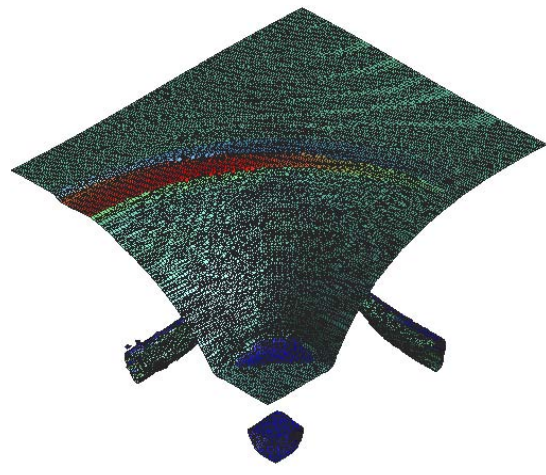


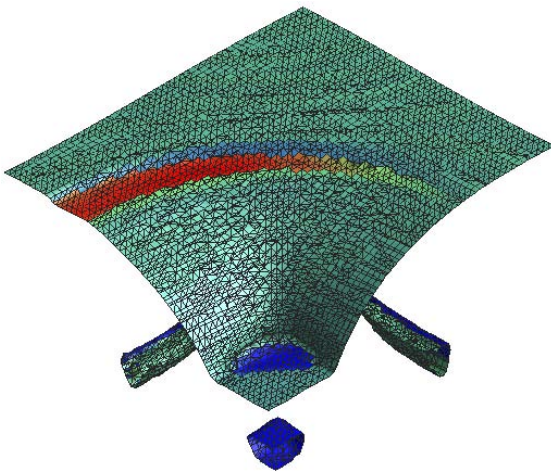
Рисунок 6: Исходные данные (слева) и данные, полученные при объединении граней ячеек регулярной сетки (справа). Задача о падении ударной волны на плоский угол, отображается величина «счетная вязкость».



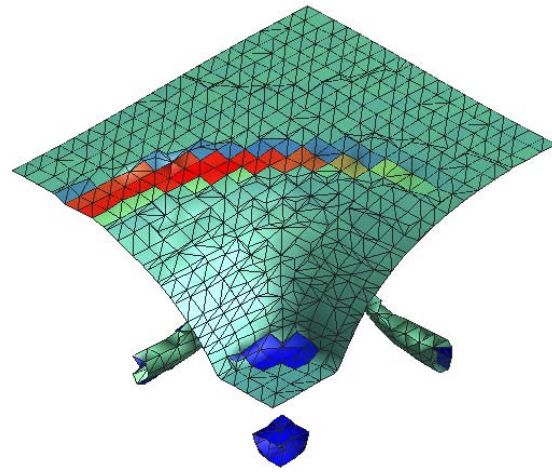
Исходная модель – 108 тыс. граней



Упрощенная модель высокого качества – 34 тыс. граней



Упрощенная модель среднего качества – 9 тыс. граней



Упрощенная модель низкого качества – 1 тыс. граней

Рисунок 7: Применение алгоритма построения упрощенных моделей к результатам работы фильтра «Изоповерхность». Анализируется задача о падении ударной волны на плоский угол, изоповерхность компоненты скорости U_y .

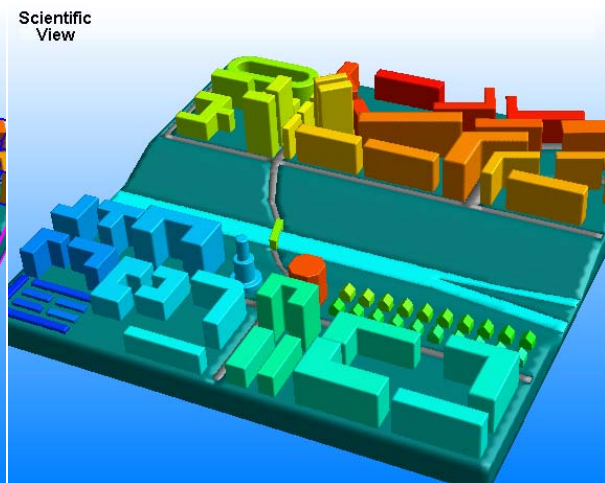
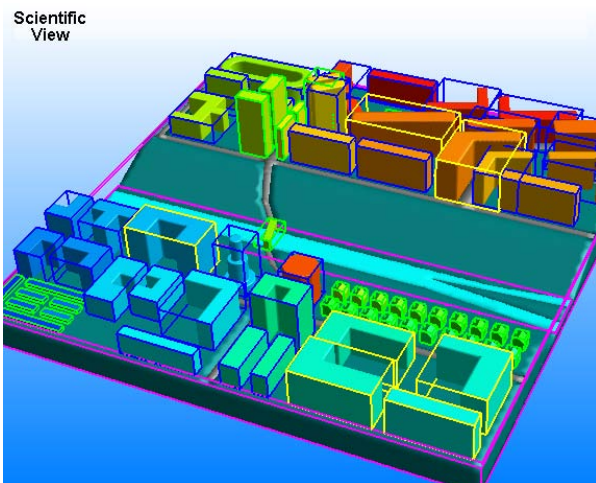


Рисунок 8: Работа алгоритма, основанного на методе «обратной связи» OpenGL. Отображается тестовая CAD геометрия «Город»

ПОИСК ОПТИМАЛЬНОГО ПУТИ НА МНОЖЕСТВЕ ПРЕГРАД В КОМПЬЮТЕРНЫХ ИГРАХ

Василий Терещенко, Денис Янчик, Дмитрий Пустовойтов

Факультет кибернетики

Киевский национальный университет имени Тараса Шевченко, Киев, Украина

v_ter@ukr.net, razor.den@gmail.com, dmytro.pustovoytov@gmail.com

Аннотация

В работе представлен алгоритм поиска оптимального пути на плоскости с учетом преград в виде простых не пересекающихся многоугольников. Алгоритм состоит из множества эффективных алгоритмов, которые в совокупности образуют единый оптимальный алгоритм. Общая временная сложность алгоритма $O(n \log n)$, объем используемой памяти $O(n)$, где n – суммарное количество всех вершин многоугольников, из которых формулируется область поиска.

Ключевые слова: *Оптимальный путь, Преграды, Монотонный многоугольник, Цель, Компьютерные игры.*

1. ВВЕДЕНИЕ

Постановка проблемы. В современных компьютерных играх-стратегиях разработчики часто решают такую проблему: провести некоторые отряды юнитов между двумя пунктами по пересеченной местности кратчайшим путем. Построение эффективного алгоритма решения этой проблемы позволило бы оптимизировать использование вычислительных ресурсов компьютера и создавать более сложные игры с препятствиями любой формы и размеров. По сути, такая задача сводится к нахождению кратчайшего пути на графах. На сегодняшний день, одним из самых эффективных алгоритмов поиска пути по графу, является A^* - алгоритм [1] и его оптимизации [1, 2]. Большинство предложенных оптимизаций этого алгоритма связаны с поиском решения проблемы быстрой и оптимальной обработки большого количества данных и преобразования их к виду, приемлемому для A^* - алгоритма. В особенности, такая проблема возникает, если игровое пространство непрерывно. При этом позиции объектов и препятствий сохранены в виде непрерывных значений и должны быть настолько точно представлены, как и разрешение экрана. Примером решения этой проблемы могут быть подходы, использующиеся для мобильных роботов. Одним из таких подходов есть сведение непрерывного пространства к нескольким дискретным вариантам [2]. Способы дискретизации пространства включают:

- нанесение ячеистой сетки на поверхность пространства поиска;
- выделение критических точек, расположенных, в основном, вблизи вершин препятствий обхода;
- разбиение на выпуклые полигоны пространства, не занимаемого полигональными препятствиями; промежуточными точками могут быть центры полигонов или точки на границах полигонов [3];

- разбиение на квадраты, где каждый квадрат, не являющийся однородным, разделяется на четыре меньших квадрата, центры которых используются для поиска пути;

- пространство между смежными препятствиями рассматривается как цилиндр, форма которого изменяется вдоль его оси. Вычисляется ось, проходящая через пространство между двумя смежными препятствиями (включая стены), и эти оси используются для поиска пути.

- представления препятствия в виде потенциального поля [4], сила которого обратно пропорциональна расстоянию до него. Так же существует однородная сила притяжения к цели. Через близкие постоянные интервалы времени вычисляется сумма притягивающих и отталкивающих векторов и объект передвигается в этом направлении. Проблема этого подхода состоит в попадании объекта в локальный минимум.

Цель работы - построение общего эффективного алгоритма поиска кратчайшего пути, на множестве преград в виде простых многоугольников, оптимизирующего современные компьютерные игры по количеству юнитов в играх жанра “стратегии”, а также по сложности и разнообразию препятствий в играх жанра “экшен”.

В работе предложен новый подход решения рассматриваемой задачи, который позволяет реализовывать игры на качественно новом уровне.

2. ОСНОВНАЯ ЧАСТЬ

В основе представленного алгоритма лежит идея интеграции нескольких эффективных методов решения задач вычислительной геометрии и геометрического моделирования для достижения максимально быстрого поиска кратчайшего пути на плоскости. При этом используется минимальный объем памяти для хранения структур данных.

Постановка задачи. Пусть задано две точки – A (стартовая) и B (конечная). Область поиска (внешний мир) представлена в виде простого многоугольника, а преграды внутри рассматриваемого мира описаны в виде самонепересекающихся и не пересекающих внешнюю границу мира простых многоугольников, рис. 1. Необходимо найти кратчайший путь между точками A и B .

В случае самопересечения или пересечения между собой следует решить задачу о разделении таких многоугольников на множество простых самонепересекающихся и непересекающихся между собой многоугольников, что не является целью данной статьи.

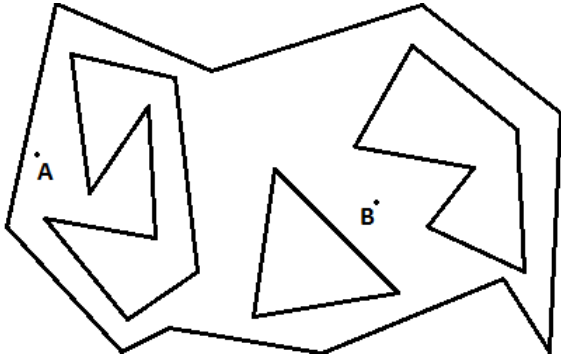


Рисунок 1: Область поиска – простой многоугольник.

Пусть многоугольники представляются и хранятся в виде циклически упорядоченной последовательности точек (вершин). Объем памяти, необходимый на хранение такой структуры $O(n)$.

Чтобы решить задачу, разобьем ее на пять подзадач: регуляризация графа, выделение монотонных многоугольников, триангуляция монотонных многоугольников, локализация точки, A^* - алгоритм. Для решения этих подзадач будем использовать известные эффективные алгоритмы. Учитывая то, что эти алгоритмы детально описаны, рассмотрим лишь вкратце каждый из них, на примере, рис.1.

2.1 Регуляризация графа

Регулярность графа рассматривается относительно некоторой прямой l . Не ограничивая общности, пусть это будет ось OX . Сориентируем граф (рис.1.) по возрастанию в направлении оси OX . Тогда вершину v будем называть *регулярной*, если множества входящих и исходящих из этой вершины ребер одновременно не пусты. Вершина v называется *не регулярной*, если хотя бы одно из множеств, входящих или исходящих из нее ребер, пустое.

Граф называется *регулярным* относительно некой прямой (например, оси OY или OX), если все его вершины, кроме первой и последней, регулярны. Если же хотя бы одна из вершин (кроме первой и последней) не регулярна, граф называется *не регулярным*. *Регуляризация* графа – процесс преобразования не регулярного графа в регулярный.

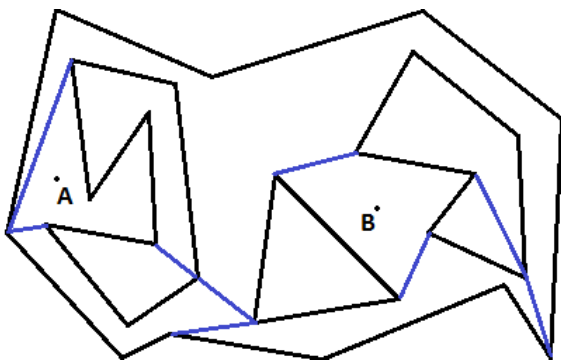


Рисунок 2: Область поиска после применения алгоритма регуляризации.

Граф на рис. 1 не регулярен относительно оси OX и его необходимо регуляризовать. Для этого используем алгоритм плоского заметания [5], в результате чего получим область поиска в виде регулярного графа, рис. 2.

2.2 Выделение монотонных многоугольников методом цепей

Следующий шаг алгоритма - разбиение полученной области поиска (рис. 2) на монотонные многоугольники. Для этого наиболее подходящим является эффективный алгоритм монотонных цепей (определения монотонной цепи и монотонного многоугольника можно найти в [5]), предложенный Ли и Препаратой [6] и усовершенствованный в работе [7], который применяется для решения задачи локализации точки на плоскости. Согласно этому алгоритму, за три линейных прохода по графу (рис. 2) мы получим сбалансированный взвешенный граф, который расщепляется на полное множество монотонных, относительно оси OX , цепей, рис. 3.

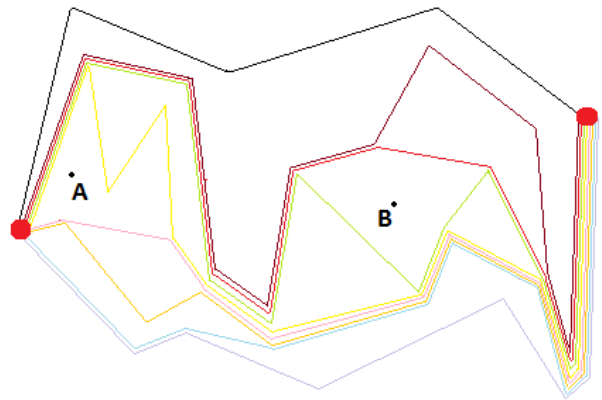


Рисунок 3: Выделенные цепи.

На основе этого множества строится структура данных – взвешенное бинарное дерево монотонных цепей. Эта структура данных позволяет нам за линейное время произвести триангуляцию графа и за $O(\log n)$ локализовать точку.

2.3 Триангуляция монотонных многоугольников

В этой задаче необходимо триангулировать монотонные многоугольники, образованные множеством последовательных монотонных цепей графа на рис.3. Полигон называется *монотонным*, если он состоит из двух (верхней и нижней), монотонных относительно одной и той же прямой l , цепей [5].

При выделении многоугольников методом цепей все полученные многоугольники являются монотонными так, как все выделенные цепи – монотонные относительно оси OX (вертикальное заметание слева направо обеспечивает монотонность), а все многоугольники образованы двумя последовательными монотонными цепями (монотонность по условию). Более детально это описано в [8], [9].

Алгоритм триангуляции монотонных многоугольников.

Переименуем вершины нашего монотонного многоугольника в v_1, v_2, \dots, v_n в порядке возрастания координаты абсцисс (именно в таком порядке наш алгоритм будет обрабатывать вершины). Алгоритм хранит стек s вершин, которые были проверены, но полностью не обработаны. Алгоритм формирует последовательность монотонных полигонов $p = p_1, p_2, \dots, p_n = \emptyset$. Полигон p_i , как результат обработки вершины v_i получается путем отсеечения нуля или нескольких треугольников от предыдущего полигона. Алгоритм завершает работу с пустым полигоном, а множество треугольников, полученное в процессе обработки, представляет собой триангуляцию исходного полигона p . Пусть s_1, s_2, \dots, s_n вершины со стека в порядке возрастания снизу вверх, тогда на каждом шаге возможно три варианта их расположения:

1. s_1, s_2, \dots, s_t упорядочены до возрастанию координаты x и содержат каждую из вершин полигона p_{i-1} , расположенную справа от s_t и слева от s_1 .
2. s_1, s_2, \dots, s_t являются последовательными вершинами либо в верхней, либо в нижней цепочках полигона p_{i-1} .
3. s_1, s_2, \dots, s_n являются вогнутыми вершинами полигона p_{i-1} (внутренний угол каждой из них более 180°).

Следовательно, последующая вершина v_i в обработке может быть в трех соотношениях с вершинами s_1, s_2, \dots, s_t , рис 4:

- a. v_i соседняя с s_t , но не с s_1 ;
- b. v_i соседняя с s_1 , но не с s_t ;
- c. v_i соседняя и с s_1 , и с s_t ;

Ядро полигона – подмножество его точек, из которых виден весь полигон (ядро выпуклого полигона – он сам). Полигон называется *веерообразным*, если ядро содержит одну или более вершин – *корней* полигона. Соответственно действия в каждом из трех случаях разные.

В случае *a*: пока $\angle v_i s_t s_{t-1} < 180^\circ$, отсекаем полигон $v_i s_t s_{t-1}$ и уменьшаем t . Когда угол стал больше 180° либо t стало равным 1, заносим v_i в стек, рис 4 *a*.

В случае *b*: отсекаем полигон $v_i s_1 s_{t-1} \dots s_t$ – он является веерообразным с узлом в точке v_i , триангулируем его. Заносим в стек s_t , после – v_i , рис. 4 *b*.

В случае *c*: v_i является v_n и полигон $v_i s_1 s_{t-1} \dots s_t$ является веерообразным с углом в точке v_n . Выполняем триангуляцию и завершаем алгоритм, рис 4 *c*.

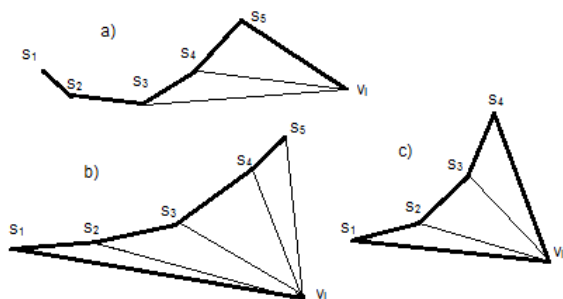


Рисунок 4: Три случая обработки следующей вершины v_i .

В результате получим триангулированный граф, рис. 5. Порядок выполнения шагов триангуляции сверху вниз, слева направо. Более детальное описание алгоритма представлено в работах [5, 9, 10].

2.4 Локализация точки методом цепей

Для локализации точек A и B будем использовать метод цепей [6] и структуру данных (бинарное дерево цепей) построенную в разделе 2.2. Сделать это достаточно легко, учитывая то, что у нас имеется планарный граф. Запускаем метод цепей для уже триангулированной области поиска. Особенности хранения данных в памяти позволяют бинарным поиском найти, между какими 2-мя цепями находится наша точка. После этого не составит труда вычислить, в каком треугольнике находится точка. После этого можно приступить к применению A^* - алгоритма.

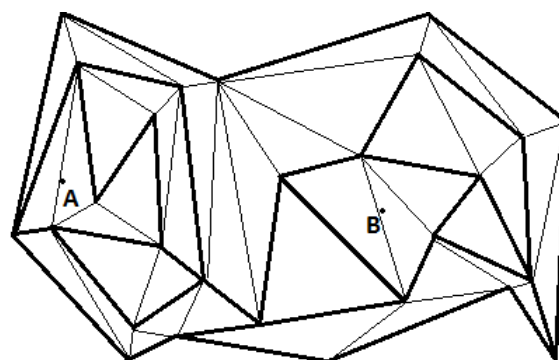


Рисунок 5: Триангуляция области поиска.

2.5 A^* алгоритм

A^* - алгоритм [1, 2 11] пошагово просматривает все пути, ведущие от начальной вершины в конечную, пока не найдет минимальный. Как и все алгоритмы поиска, он просматривает сначала те маршруты, которые «кажутся» ведущими к цели. От жадного алгоритма (который тоже является алгоритмом поиска по первому лучшему совпадению) его отличает то, что при выборе вершины он учитывает, помимо прочего, весь пройденный до неё путь (составляющая $g(x)$) – это стоимость пути от начальной вершины, а не от предыдущей, как в жадном алгоритме [1].

В начале работы рассматриваются узлы, смежные с начальной точкой; выбирается тот из них, который имеет минимальное значение $f(x)$, после чего этот узел раскрывается. На каждом этапе алгоритм оперирует с множеством путей из начальной точки до всех ещё не раскрытых (листовых) вершин графа («множеством частных решений»), которые размещаются в очереди с приоритетом. Приоритет пути определяется по значению $f(x) = g(x) + h(x)$. Алгоритм продолжает свою работу до тех пор, пока значение $f(x)$ целевой вершины не окажется меньшим, чем любое значение в очереди (либо пока всё дерево не будет просмотрено). Из множества решений выбирается решение с наименьшей стоимостью. Чем меньше эвристика $f(x)$, тем больше приоритет (поэтому для реализации очереди можно использовать сортирующие деревья).

Алгоритм A^* обходит при этом минимальное количество вершин, благодаря тому, что он работает с «оптимистичной» оценкой через вершину. Оптимистичной в том смысле, что, если он пойдёт через эту вершину, то у алгоритма «есть шанс», что реальная стоимость результата будет равна этой оценке, но никак не меньше.

Когда A^* завершает поиск, то он, согласно определению, нашёл путь, истинная стоимость которого меньше, чем оценка стоимости любого пути через любой открытый узел. Но поскольку эти оценки являются оптимистичными, соответствующие узлы можно без сомнений отбросить. Иначе говоря, A^* никогда не упустит возможности минимизировать длину пути, и потому является допустимым.

Предположим теперь, что некий алгоритм B выдал в качестве результата путь, длина которого больше оценки стоимости пути через некоторую вершину. На основании эвристической информации, для алгоритма B нельзя исключить возможность, что этот путь имел и меньшую реальную длину, чем полученный результат. Соответственно, пока алгоритм B просмотрел меньше вершин, чем A^* , он не будет допустимым. Итак, A^* проходит наименьшее количество вершин графа среди допустимых алгоритмов, использующих такую же точную (или менее точную) эвристику. Более подробную информацию, с различными модификациями и оптимизациями A^* - алгоритма можно найти в [11].

3. АНАЛИЗ СЛОЖНОСТИ АЛГОРИТМОВ

Общая сложность алгоритма решения задачи поиска оптимального пути на плоскости с учетом преград в виде простых не пересекающихся многоугольников определяется суммой сложностей алгоритмов пяти подзадач.

1. Время, потраченное на регуляризацию n – вершинного планарного графа составляет $O(n \log n)$ с использованием $O(n)$ памяти [5].

2. Общее время алгоритма выделения монотонных многоугольников в заданном графе равно $O(n \log n)$. При этом, $O(n \log n)$ времени уходит на предобработку (сортировка ребер графа за и против часовой стрелки), после инициализация, 1-ый и 2-ой проходы по $O(n)$ каждый – мы бываем в каждой вершине всего 1 раз. $O(n)$ – сложность выделения многоугольников из последовательности цепей.

3. На триангуляцию монотонных многоугольников уходит $O(n)$ времени. Задаваемый на входе полигон содержит n вершин. Заметим, что при каждой итерации двух внутренних циклов while из стека исключается одна вершина. Однако каждая вершина записывается в стек только однажды (при первой ее обработке) и, следовательно, может быть выбрана из стека тоже только однажды. Поскольку алгоритм выполняет $O(n)$ операций со стеком одинаковой длительности и затрачивает одинаковое время между двумя такими последовательными операциями, то время работы программы пропорционально $O(n)$. Нижняя граница определяется тем, что должна быть обработана каждая из n вершин [12]. Кроме того, в работе [10] показано, что триангуляцию можно оптимизировать до сложности $O(n \log(\log n))$. А Чазелле показал [4], что простой многоугольник может быть триангулирован за линейное время.

4. Локализацию точки в n -вершинном планарном подразбиении можно реализовать методом цепей [6] за время $O(\log^2 n)$ с использованием $O(n)$ памяти при затратах $O(n \log n)$ времени на предобработку. Учитывая оптимизацию нашего алгоритма под метод цепей, сложность поиска составляет порядка $O(\log(2 \log n))$.

5. Временная сложность алгоритма A^* зависит от эвристики. В худшем случае, число вершин, исследуемых алгоритмом, растёт экспоненциально по сравнению с длиной оптимального пути, но сложность становится полиномиальной, когда эвристика удовлетворяет следующему условию:

$$|h(x) - h^*(x)| \leq O(\log h^*(x))$$

Где h^* – оптимальная эвристика, то есть точная оценка расстояния из вершин x к цели. Другими словами, ошибка $h(x)$ не должна расти быстрее, чем логарифм от оптимальной эвристики, [1].

4. ПРАКТИЧЕСКАЯ ЧАСТЬ

На основе выше изложенной теории нами была разработана практическая реализация данного алгоритма. На Рис. 6 приведен график зависимости скорости работы программы от суммарного количества вершин всех преград.

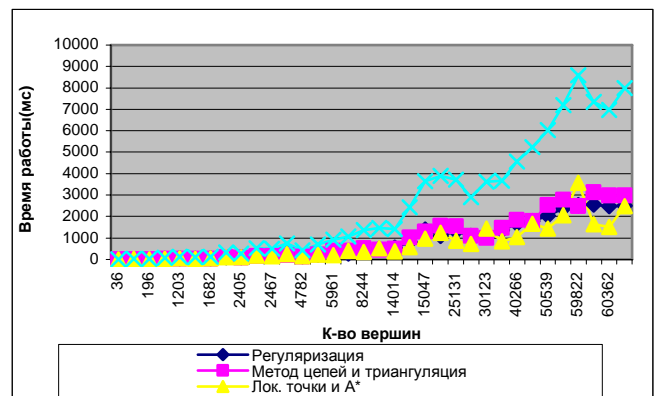


Рисунок 6: Результаты работы алгоритма.

С графика видно, что время работы всех алгоритмов в отдельности имеют один порядок сложности, причем он не является квадратическим. Таким образом, можно утверждать, что приведенные теоретические оценки производительности имеют место и на практике.

5. ЗАКЛЮЧЕНИЕ

В статье рассматривается новый алгоритм поиска оптимального пути на множестве преград. Представленный алгоритм реализуется с помощью решения пяти подзадач за оптимальное время $O(n \log n)$ (сложность всех 5-ых подзадач составляет не более $O(n \log n)$), используя минимальный объем памяти $O(n)$, для хранения структур данных. Предложенный обобщенный алгоритм можно применять при создании компьютерных игр жанра «стратегия», позволяя моделировать динамическое изменение области поиска.

6. СПИСОК ЛИТЕРАТУРЫ

- [1] Lester, Patrick. "A* Pathfinding for Beginners". 21 Oct. 2006 <http://www.policyalmanac.org/games/aStarTutorial.htm>.
- [2] Bryan Stout. Smart move: Intelligent path-finding. Online at gamasutra.com/view/feature/3317/smart_move_intelligent.php, 1999. 8.
- [3] Y.-J. Chiang and R. Tamassia. Optimal shortest path and minimum-link path queries between two convex polygons inside a simple polygonal obstacle. *Internat. J. Comput. Geom. Appl.*, 7, 1997, pp. 85-121.
- [4] Chazelle B. Triangulating a Simple Polygon in Linear Time. *Discrete Comput. Geom.* 6 (1991), 485-524.
- [5] Препарата Ф., Шеймос М. Вычислительная геометрия: Введение / Пер. с англ. М.: Мир, 1989.
- [6] D. T. Lee and F.P. Preparata. Location of a point in a planar subdivision and its applications. *SIAM Journal on computing* 6(3), 1977, pp. 594-606.
- [7] E. Edelsbrunner, L. J. Guibas and J. Stolfi. Optimal point location in a monotone subdivision. *SIAM J. Computing.* 15(2), 1986, pp. 317 – 340.
- [8] Скворцов А.В., Костюк Ю.Л. Применение триангуляции для решения задач вычислительной геометрии // Геоинформатика: Теория и практика. Вып. 1. Томск: Изд-во Томск. ун-та, 1998. С. 127–138.
- [9] Gilbert P.N. New results on planar triangulations. Tech. Rep. ACT-15, Coord. Sci. Lab., University of Illinois at Urbana, July 1979.
- [10] Tarjan R.E., Van Wyk C. J. An $D(\eta \log \log \eta)$ -time algorithm for triangulating a simple polygon, *SIAM J. Comput.* 17 (1988), pp. 143-178.
- [11] Björnsson Y., Enzenberger M., Holte R. C., Schaeffer J.. Fringe Search: Beating A* at Pathfinding on Game Maps. In Proc. of the 2005 IEEE Symposium on Computational Intelligence and Games. Colchester, Essex, UK, 2005, pp. 125-132.
- [12] М.Ласло. Вычислительная геометрия и компьютерная графика на C++ : Пер. с англ. В. Львова. - М.: Бином, 1977.

About the author

Vasyl Tereshchenko is an associate professor at National Taras Shevchenko University of Kyiv, Faculty of Cybernetics. His contact email is v_ter@ukr.net.

Denis Yanchik is a student at National Taras Shevchenko University of Kyiv, Faculty of Cybernetics. His contact email is razor.den@gmail.com

Dmitro Pustovoytov is a student at National Taras Shevchenko University of Kyiv, Faculty of Cybernetics. His contact email is dmytro.pustovoytov@gmail.com

Разработка инструментария для показа стерео-презентаций из файлов в KML-формате

Александр Бобков, Илья Казанский, Станислав Клименко
 Факультет общей и прикладной физики,
 Московский физико-технический институт, Москва, Россия
 alexbobbkov@list.ru, kazansky@icpt.su, stanislav.klimenko@gmail.com

Аннотация

Статья посвящена разработке инструментария для показа интерактивных стерео-презентаций, содержащих геопривязанные данные. Стереопрезентация — это презентация с возможностью визуализации данных в стереоскопических системах виртуальной реальности. Для хранения данных используется формат KML. Для подготовки данных используется геоинформационное приложение Google Earth. Оно обеспечивает просмотр спутниковых снимков и работу с геопривязанной информацией. Имеется возможность добавления пользовательских данных: меток, объектов, изображений, виртуальных путешествий. Эти данные экспортируются в файлы в KML-формате.

В рамках данной работы разработано приложение, которое обеспечивает работу со стерео-презентациями на базе данных в KML-формате и показывать эти презентации на установке виртуальной реальности. Для работы с KML-файлами используется библиотека libkml, для стерео-визуализации — фреймворк AVANGO.

Keywords: виртуальное окружение, стереопрезентации, Google Earth, KML.

1. ВВЕДЕНИЕ

В настоящее время увеличивается популярность стереоскопических систем: снимаются 3d-фильмы, на рынке становятся доступны бюджетные системы активного стерео. Стереоскопические технологии проникают в обычные виды деятельности. Например, возникает понятие стереопрезентации. Стереопрезентация — это презентация, которая обеспечивает стереоскопический просмотр и взаимодействие с данными. Стереопрезентации опираются на технологию виртуального окружения. Виртуальное окружение — современная технология человеко-машинного взаимодействия, позволяющая “погружение” человека в трехмерную среду предметной области и более эффективное усвоение информации.

В данной работе рассматриваются стереопрезентации, содержащие географическую информацию. Под этим понимаются следующие вещи: географические карты, спутниковые снимки, трехмерный ландшафт, трехмерные модели, текстовая, звуковая и видеоинформация, привязанная к географическим координатам, история изменений этой информации во времени.

Чтобы определить способы и методы работы с геопривязанной информацией, имеет смысл рассмотреть понятие “неогеография”. Этот термин был введен в 2006 году Эндрю Тернером и обозначает совокупность новых

методов и средств для работы с геопространственной информацией. Неогеографию отличают следующие принципы: использование растровых данных, как основного представления данных; использование географических координат, вместо картографических; использование открытых форматов данных.

Один из таких открытых форматов — формат KML (Key-hole Markup Language), который является также родным для приложения Google Earth. В настоящее время это открытый стандарт, который поддерживается Открытым геопространственным консорциумом (Open Geospatial Consortium). KML позволяет хранить ссылки на внешние данные (звуки, фото, видео, гиперссылки) и задавать географическую привязку для этих данных. Также KML позволяет задавать векторные данные (точки, маршруты, контуры), добавлять временную метку к данным и указывать способ обзора данных (точку обзора, направление обзора, облет по заданному маршруту).



Figure 1: Устройство стереопрезентации

Использование методов неогеографии и формата KML видится разумным для работы со стереопрезентациями, содержащими геопривязанные данные. Можно выделить следующие концепции представления данных:

- Трехмерная модель Земли. Модель должна содержать спутниковые снимки поверхности и трехмерный ландшафт;
- Географические и другие карты, наложенные на трехмерную модель Земли;
- Текстовая, звуковая, фото и видеоинформация, прикрепленная к модели Земли в заданных точках;
- Векторная информация, наложенная на модель Земли;

- Просмотр информации в исторической перспективе
- Интерактивное перемещение по трехмерной модели Земли и взаимодействие с геопривязанной информацией;
- Воспроизведение записанных виртуальных “путешествий”.

Цель данной работы — разработка программного инструментария для показа стерео-презентаций, который обладает вышеприведенными возможностями. Все возможности можно разбить на 2 группы: интерактивная трехмерная модель Земли и работа с геопривязанной информацией из KML-файлов. Устройство стерео-презентации изображено на рис. 1. В статье рассматривается только работа с KML-файлами. Разработка интерактивной модели Земли выходит за рамки этой статьи.

Разработанный инструментарий может использоваться для широкого круга задач, связанных с географией. Это могут быть стерео-презентации в школах, вузах, музеях, выставках и конференциях. Стерео-презентации могут использоваться для бизнес-планирования и принятия решений. На базе инструментария могут быть разработаны ситуационные центры, авто и авиа-тренажеры нового поколения.

Новизна подхода заключается в использовании формата KML и фреймворка AVANGO. Формат KML обеспечивает возможность интеграции инструментария с другими приложениями, прозрачный способ хранения и обмен данными. Фреймворк AVANGO обеспечивает возможность использования различных стереосистем, устройств ввода и обеспечивает быструю модификацию инструментария для конкретных задач.

2. ПОХОЖИЕ СИСТЕМЫ

2.1 Present3D

Present3D — приложение для показа стерео-презентаций. Оно разрабатывается Робертом Осфилдом, который также является разработчиком графического инструментария OpenSceneGraph. Present3D использует OpenSceneGraph в качестве графической подсистемы.

Презентация в Present3D состоит из набора слайдов. Present3D позволяет вставлять в слайды 3d-модели и анимировать их. Для задания слайдов используется формат XML.

Наш инструментарий отличается тем, что он нацелен на более узкую область презентаций — презентаций, которые содержат геопривязанную информацию. Инструментарий позволяет размещать такую информацию на виртуальном глобусе — интерактивной модели планеты Земля — в заданных геокоординатах. И, в отличие от Present3D, он использует формат KML для описания данных.

2.2 Google Earth

Google Earth — геоинформационное приложение, которое содержит трехмерную модель Земли со спутниковыми снимками и трехмерным ландшафтом высокой детализации. Google Earth также содержит различную геопривязанную информацию: названия

стран, населенных пунктов, рек, географические достопримечательности.

Google Earth позволяет загружать из KML-файлов пользовательские данные и просматривать их. Google Earth позволяет также создавать свои собственные геопривязанные данные: ставить метки, накладывать изображения, отмечать маршруты и области, добавлять 3d-модели зданий и других объектов. Google Earth позволяет записывать “путешествия” и записывать звуковое сопровождение. И в этом смысле его можно рассматривать как средство создания и показа географических презентаций.

Недостатком Google Earth является его закрытость и проприетарность. Google Earth не позволяет писать свои расширения. В том числе, невозможна стереоскопическая визуализация.

Наш инструментарий отличается тем, что он использует открытые библиотеки и язык Python для высокоуровневого программирования. Это позволяет обеспечить стереоскопическую визуализацию, а в перспективе и другие возможности, которые нет у Google Earth, например, синтез русской речи для воспроизведения описаний геообъектов.

Но Google Earth остается удобным средством для ввода данных. Его можно использовать для первичной подготовки данных, затем экспортировать их в формате KML и использовать с нашим инструментарием.

3. ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ

3.1 KML

Рассмотрим формат KML[1]. Этот формат был разработан и начал использоваться в приложении Google Earth. Сейчас это открытый стандарт[2], и его поддерживает различное программное обеспечение. KML-файлы часто распространяются в виде ZIP-архива с расширением .kmz. В этом случае архив может также содержать дополнительные файлы: изображения, звуковые файлы, 3d-модели.

Формат KML основан на XML, и поэтому файлы в KML-формате представляют собой иерархическое дерево элементов. На рис. 2 изображена схема элементов KML. Все элементы наследуются от базового элемента Object. Object имеет свойство id — идентификатор элемента. Это позволяет всегда найти в KML-файле нужный элемент по его идентификатору.

Главный структурный элемент KML-файла — Feature. Производные элементы — Document и Folder служат для группировки других элементов Feature. Другие производные элементы — листья дерева элементов — называются визуальными, т.к. они содержат данные, которые видит пользователь. К ним относятся элемент Placemark и элементы-наложения GroundOverlay, ScreenOverlay и PhotoOverlay, содержащие разные виды изображений. Все остальные элементы KML являются вспомогательными и задают свойства структурных элементов.

Элемент Placemark содержит геометрию — один из производных элементов от Geometry. Геометрия может быть следующих типов: точка (Point), путь (LineString), многоугольник (Polygon), 3d-модель (Model). Элемент Model содержит координаты местоположения модели и ссылку на внешний файл с моделью в формате COLLA-

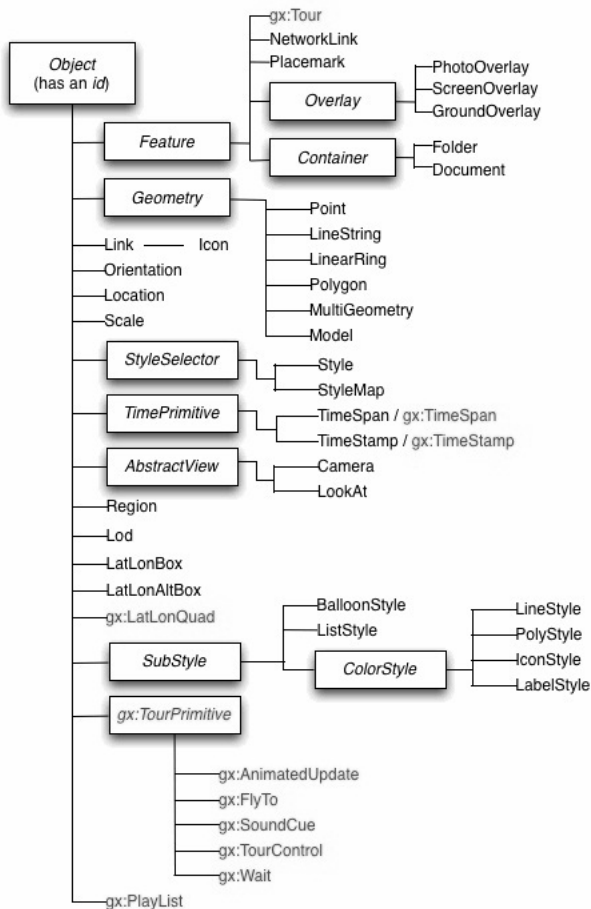


Figure 2: Схема KML

DA.

Элементов-наложения существует 3 типа: GroundOverlay, ScreenOveraly и PhotoOverlay. Все они содержат ссылки на внешние изображения. Отличаются они способом показа этого изображения.

GroundOverlay — изображение, которое накладывается на трехмерный ландшафт поверх других изображений. Изображение вписывается в четырехугольную рамку. Географические координаты углов рамки задаются через свойство LatLonBox.

ScreenOverlay — изображение, которое накладывается на экран. Можно задавать размеры изображения и положение на экране.

PhotoOverlay — позволяет помещать фотографии в любую точку Земли и задавать параметры просмотра. Это могут быть как обычные плоские фотографии, так и панорамные.

Во всех вышеприведенных элементах используются для задания координат точек используются географическая система координат: долгота от -180° до $+180^\circ$, широта от -90° до $+90^\circ$ и высота в метрах. При этом многие элементы имеют дополнительное свойство altitudeMode, которое может принимать 3 значения: clampToGround — игнорировать высоту, поместить точку прямо на трехмерный ландшафт, relativeToGround — считать высоту как высоту над ландшафтом, absolute — считать

высоту, как высоту над уровнем моря.

Внешний вид геометрии может задаваться с помощью стилей. Как видно из рис. 2, существуют стили для линий, для полигонов, для меток и других элементов.

В KML-файле можно задавать положение точки наблюдения и направление наблюдения с помощью двух элементов: LookAt и Camera (рис. 3). LookAt позволяет задать положение относительно рассматриваемого объекта. Camera позволяет задать абсолютное положение обзора. Элемент LookAt можно использовать, например, вместе с элементом Placemark для задания оптимального обзора для наблюдения данного объекта.

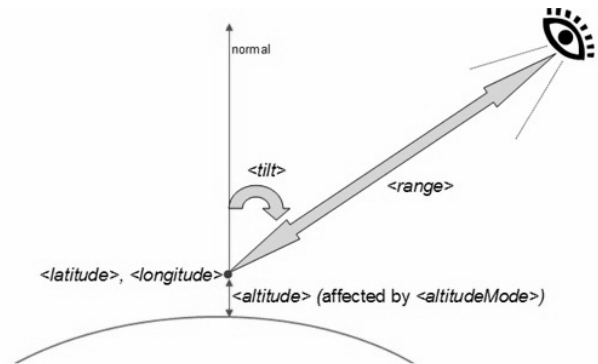


Figure 3: Задание положения камеры

В KML-файле можно задавать “путешествия” (Tour). Путешествие — полет камеры, который сопровождается звуками и изменениями объектов из файла. Путешествие состоит из последовательности примитивов. Примитивы бывают нескольких видов: полет камеры к заданной точке (FlyTo), пауза (Wait), запуск звука (SoundCue), запуск изменения (AnimatedUpdate).

Элемент FlyTo содержит конечное положение камеры, заданное с помощью LookAt или Camera, и продолжительность движения из текущего положения в конечное.

Элемент AnimatedUpdate позволяет изменять свойства элементов из KML-файла. Каждый элемент может иметь идентификатор id. AnimatedUpdate ссылается на идентификатор и указывает свойство, которое нужно изменить. Например, это можно использовать для управления видимостью элементов.

3.2 libkml

С KML-файлами можно работать как с обычными XML-файлами. Но для более удобного программного доступа к KML-файлам компанией Google разработана библиотека libkml[3]. Библиотека написана на языке C++, и имеет привязки к языкам Java и Python.

libkml состоит из нескольких составных частей. KML DOM — программный интерфейс, который обеспечивает доступ к иерархической структуре KML-файла. Он позволяет проходить по дереву элементов, осуществлять прямой доступ к свойствам элементов, в том числе вносить изменения в KML-файл. KML Engine — программный интерфейс, расширяющий функционал KML DOM. Он позволяет читать KML и KMZ-файлы и разных источников, разрешать ссылки на другие KML-файлы, делать поиск стиля по идентификатору

(несколько геометрий могут иметь общий стиль, поэтому стили обычно хранятся отдельно, а геометрии содержат ссылки на нужные стили), работать с размерами элементов.

3.3 AVANGO

Для создания стереоскопической визуализации применяется кроссплатформенный фреймворк AVANGO[4]. AVANGO поддерживает различные виды дисплеев и устройств ввода. AVANGO обеспечивает быстрое создание интерактивных графических приложений. На нижнем уровне AVANGO опирается на графическую библиотеку OpenGL. Для написания программ используется язык Python[5]. AVANGO состоит из нескольких подсистем:

- Подсистема поддержки различных видов дисплеев и многоэкранных конфигураций;
- Подсистема поддержки различных устройств ввода (джойстиков, рулей, манипуляторов);
- Подсистема поддержки звука;
- Подсистема поддержки распределенных приложений. Она позволяет синхронизировать данные между разными компьютерами;
- Настройка над библиотекой OpenGL[6]. Она добавляет к классам OpenGL возможность быть узлами графа потоков данных;

Главная особенность AVANGO — это граф потоков данных[7]. Это структура данных, которая позволяет обеспечить гибкий обмен данных между разными объектами. Состояние каждого объекта — узла в графе потоков данных — описывается набором полей. Поле может хранить некоторое значение и быть связанным односторонней связью с другими полями других объектов. В случае изменения значения поля-источника это значение пересылается по связи в поле-приемник. У объекта, которому принадлежит это поле-приемник, вызывается специальный метод, где он может прореагировать на изменение[8].

Такой граф потоков данных можно формировать по ходу выполнения приложения. Можно добавлять и удалять узлы и связи. Граф потоков данных позволяет организовать взаимодействие между разными подсистемами приложения. Данные от устройств ввода по связям поступают в объекты-вычислители. Объекты-вычислители передают данных по связям к объектам OpenGL[9], которые уже отрисовываются на экране. Данные могут пересылаться по сети для синхронизации между экземплярами приложения, запущенными на разных компьютерах. Связи могут использоваться для анимации трехмерных объектов.

4. РАЗРАБОТКА ИНСТРУМЕНТАРИЯ

4.1 Архитектура инструментария

Весь функционал инструментария можно разбить на 3 большие группы:

- Визуализация данных из KML-файла;

- Взаимодействие с данными:

- Запуск воспроизведения видео;
- Запуск воспроизведения звуков;
- Показ изображений;
- Показ описаний объектов;
- Скрытие/показ самих объектов;
- Просмотр исторической информации;

- Управление движения камерой;

- Ручное управление с помощью мыши или джойстика;
- Автоматическое управление — воспроизведение “путешествий” из KML-файла.

Чтобы данные могли быть визуализированы, они должны быть организованы в структуру данных графа сцены. Граф сцены управляется библиотекой OpenGL. Эта структура данных удобна для визуализации, но не удобна для взаимодействия с данными. Данные в KML-файла также не удобны для прямой работы с ними. Поэтому удобно ввести промежуточное хранилище данных. Данные из KML-файла помещаются в хранилище данных, а затем из них формируется граф сцены для визуализации.

На основе данных из хранилища можно организовать систему меню. С помощью меню можно осуществлять взаимодействие с данными, как перечислено в вышеприведенном списке. В том числе из меню можно управлять графом сцены, а значит и визуализацией.

KML-файл может также содержать записанные “путешествия”. Модуль путешествий может управлять движение камеры по трехмерной сцене. Как было указано в описании формата KML, путешествия могут также содержать изменения данных. Например, можно автоматически скрывать или показывать отдельные объекты.

Схему работы инструментария можно увидеть на рис. 4. Далее все модули будут рассмотрены подробнее.

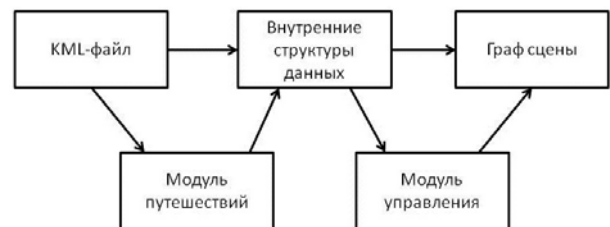


Figure 4: Схема работы инструментария

4.2 Подготовка данных

Рассмотрим сначала хранилище данных. Основные сущности, которые необходимо хранить: метки (точки), контуры, сложную геометрию, 3d-модели, изображения, стили.

В хранилище каждая сущность состоит из двух частей: это “сырые” данные взятые из KML-файла (они характеризуются географическими координатами)

и преобразованные данные, готовые в визуализации, т.е. узлы графа сцены (они характеризуются декартовыми координатами). Каждая сущность имеет также свой идентификатор, который берется из KML-файла.

В хранилище сущности хранятся в виде дерева. Граф сцены — это направленный ациклический граф, но для простоты его тоже можно считать деревом. Эти деревья структурно не совпадают, т.к. организованы по разным принципам. Структура дерева хранилища определяется KML-файлом. Данные в нем сгруппированы по логическому принципу: данные, близкие друг другу по смыслу или по типу объектов, находятся в хранилище рядом. В графе сцены же данные организованы по принципу эффективности визуализации. Данные, близкие пространственно, находятся в графе сцены рядом. Это необходимо для быстрого отбрасывания данных, которые выходят за объем видимости. Например, граф сцены может представлять собой структуру данных quadtree, когда пространство рекурсивно делится на 4 квадрата.

Инициализация инструментария состоит из нескольких этапов: разбор KML-файла и помещение данных в хранилище, построение узлов графа сцены для каждой сущности в хранилище и объединение всех узлов в граф сцены. Рассмотрим эти этапы подробнее.

Для разбора KML-файла применяется библиотека libkml. Происходит проход по иерархической структуре KML-файла, во время которого для обработки разных типов элементов применяются специальные функции-обработчики. Например, функция HandleContainers осуществляет обработку элементов Folder и Container, производит спуск по дереву элементов и вызывает другие функции-обработчики: HandlePlacemark, HandleGroundOverlay, HandleScreenOverlay, HandleTour. Функции обработчики помещают данные в хранилище.

Далее происходит создание узлов графа сцены для каждой сущности. При этом все географические координаты точек (долгота, широта, высота) преобразуются в декартовы координаты в системе координат, связанной с центром Земли.

Для меток создаются сферы, расположенные по координатам метки. Рядом со сферой помещается текстовая подпись с названием метки. Чтобы сфера и подпись были видны с любого расстояния, над сферой в графе сцены помещается объект AutoTransform. Он автоматически вычисляет необходимую матрицу масштабирования.

Пути и многоугольники напрямую преобразуются в стандартные геометрические примитивы OpenGL. Пути строятся двумя способами: с использованием геометрических примитивов LINE_STRIP и QUAD_STRIP. Первый вариант удобен, если путь должен выглядеть тонкой линией. Второй — если нужно показать толщину. OpenGL поддерживает только выпуклые многоугольники, поэтому сложные многоугольники из KML-файла предварительно разбиваются тесселятором на простые примитивы.

Накладываемое на землю изображение преобразуется в четырехугольник с текстурой.

Сложнее обстоит дело с экранными наложениями. Экранные наложения должны быть зафиксированы относительно камеры, а не трехмерной сцены. Для этого к корню графа сцены прикрепляется

специальный узел Camera. Камера содержит единичную матрицу вида и ортографическую матрицу проекции. Камера настраивается так, чтобы отрисовывать свой подграф после всех остальных объектов. Перед отрисовкой очищается буфер глубины. К этой камере прикрепляются подузлы, содержащие экранные наложения. Настройки камеры обеспечивают, что наложения будут всегда видны на экране. Матрица проекции не зависит от того, для левого или правого глаза отрисовываются экранные наложения. Поэтому для обоих глаз экранные наложения совпадают на экране. Это улучшает восприятие изображений.

После создания всех узлов начинается построение графа сцены. При этом для каждого узла сверху добавляет дополнительный узел Switch, который позволяет управлять видимостью объекта.

4.3 Взаимодействие с данными

Далее на основе хранилища можно создать меню для взаимодействия с данными. 3d-меню создается средствами библиотеки AVANGO. Для каждой сущности в хранилище можно добавить свою кнопку: для запуска видео, звука, для показа изображения, для показа/скрытия меток, контуров, объектов.

Сущностей может быть очень много, поэтому для каждой делать свою кнопку не целесообразно. Для каждой отдельной стерео-презентации разработчик может сам указать, какие кнопки ему нужны. При этом он может использовать стандартные классы кнопок для осуществления стандартных операций: показ/скрытие объектов, запуск видео, звуков.

Например, рассмотрим показ/скрытие объектов. Мы помним, что для каждого объекта в графе сцены создан специальный узел-переключатель Switch, который умеет управлять видимостью объекта. Существует специальный класс SwitchButton, который содержит ссылки на переключатель и кнопку меню, а также поля ButtonPressed и VisibleChild, которые связаны с соответствующими полями переключателя и кнопки меню. При нажатии на кнопку происходит изменение видимости объекта и изменение надписи на кнопке.

Помимо основного меню, существует слайдер, который соответствует шкале времени. При перемещении ползунка слайдера изменяется текущее время приложения. В зависимости от наличия у объектов временных меток показывают только те объекты, которые соответствуют текущему времени.

Также для взаимодействия с метками разработан специальный класс MarkPointBehavior. Этот класс позволяет выводить при нажатии на метку контекстное меню, описание метки и проигрывать звуковой файл с описанием метки (если такой существует). Создается единственный экземпляр этого класса и прикрепляется ко всем меткам с помощью стандартного механизма библиотеки AVANGO. При нажатии правой кнопкой мыши на метку вызывается функция show_context_menu() этого класса, и в качестве аргумента передается выбранная метка.

4.4 Управление камерой

Для полноценной работы со стерео-презентацией пользователь должен иметь возможность перемещаться по трехмерной сцене, либо самостоятельно управляя

камерой, либо автоматически, с помощью “путешествий” из KML-файла.

Для ручного режима разработан специальный класс-манипулятор, который умеет работать как с мышью, так и с джойстиком SpaceNavigator[10]. Главная особенность этого манипулятора в отличие от стандартных манипуляторов — учет кривизны Земного шара и учет трехмерного ландшафта. Манипулятор позволяет отдаляться от поверхности Земли и приближаться. При этом скорость зависит от высоты. Манипулятор не позволяет проваливаться сквозь землю. Манипулятор позволяет перемещаться вдоль поверхности Земли.

Рассмотрим теперь автоматический режим. Как мы помним, каждое путешествие состоит из набора элементов: FlyTo, Wait, SoundCue, Animated Update. Воспроизведение путешествия заключается в последовательном выполнении всех элементов. Для этого создан специальный класс CameraPath с единственным экземпляром. Для запуска путешествия в этот экземпляр загружается список примитивов путешествия. Каждый кадр происходит вызов функции evaluate и проход по списку примитивов. Если текущий примитив — полет камеры к заданной точке, то аппроксимируется текущее положение камеры между начальным положением на момент запуска этого примитива и конечным положением. Вычисленное значение помещается в выходное поле MatrixOut. Если текущий примитив — запуск звукового файла, то этот файл загружается и воспроизводится. Если текущий примитив — изменение, то производится поиск в хранилище объекта, который нужно изменить и вносятся необходимые изменения. Выходное значение MatrixOut соединяется связью с матрицей вида камеры.

Для каждого загруженного путешествия заводится своя кнопка меню. При нажатии на кнопку запускается соответствующее путешествие.

5. РЕЗУЛЬТАТЫ

На основе инструментария была создана система визуализации для проекта “Виртуальный Оболенск”. Проект содержит модели 18ти зданий в формате COLLADA с фотографическими текстурами (рис. 5). Ссылки на модели и их геокоординаты хранятся в соответствующем KML-файле. Также проект содержит данные об эпидемиях болезней в Оболенске с 2004 года. Каждому факту появления болезни соответствует метка в KML-файле. Метка обладает геокоординатами, которые указывают на дом и квартиру, где болезнь была обнаружена, и датой обнаружения болезни (рис. 6). Инструментарий позволяет просматривать эти метки в хронологическом порядке.

6. ЗАКЛЮЧЕНИЕ

Разработан инструментарий для показа стереопрезентаций, содержащих геопривязанные данные. Инструментарий позволяет использовать данные из файлов в формате KML и совмещать их с интерактивной моделью Земли. Инструментарий позволяет взаимодействовать с данными: запускать привязанные звуковые и видео файлы, показывать текстовую и графическую информацию. Инструментарий позволяет воспроизводить виртуальные “путешествия”, записанные в приложении Google Earth.

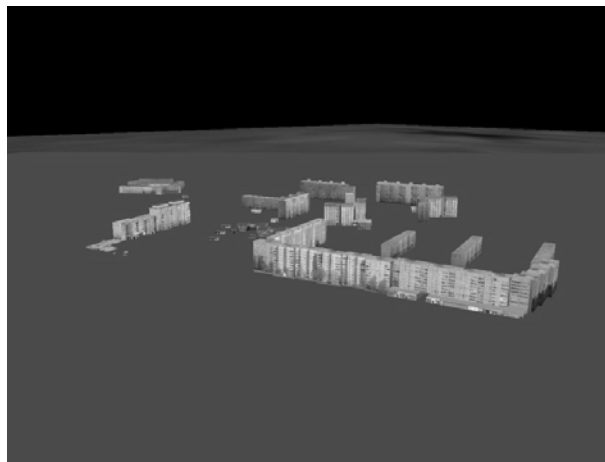


Figure 5: Виртуальный Оболенск

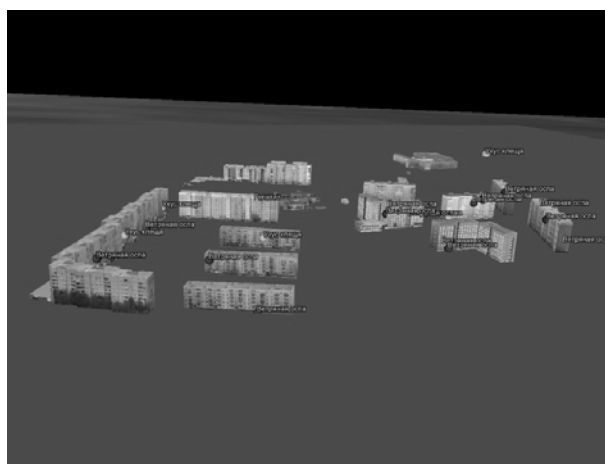


Figure 6: Информация об эпидемиях

Инструментарий опирается на открытые библиотеки и использует высокоуровневый язык Python. Это позволяет легко интегрировать его в сторонние приложения и создавать самые разнообразные географические презентации.

7. БЛАГОДАРНОСТИ

Авторы выражают благодарность Российскому фонду фундаментальных исследований (РФФИ) за частичное финансирование по грантам 08-07-00362-а и 09-07-00401-а.

Авторы выражают благодарность Еремченко Е.Н. за поддержку и предоставленные геоданные.

8. СПИСОК ЛИТЕРАТУРЫ

- [1] “Kml reference,” <http://code.google.com/apis/kml/documentation/kmlreference.html>.
- [2] “Kml specification,” <http://www.opengeospatial.org/standards/kml/>.
- [3] “libkml,” <http://code.google.com/p/libkml/>.

- [4] Henrik Tramberend, “Avocado: A distributed virtual reality framework,” in *VR '99: Proceedings of the IEEE Virtual Reality*, Washington, DC, USA, 1999, p. 14, IEEE Computer Society.
- [5] Roland Kuck et al., “Improving the avango vr/ar framework — lessons learned,” *Virtuelle und Erweiterte Realität: 5. Workshop der GI-Fachgruppe VR/AR*, pp. 209–220, 2008.
- [6] Robert Osfield, “Openscenegraph: Past, present and future,” in *SIGGRAPH 2006 OpenSceneGraph Birds of Feather session*, 2006.
- [7] Л.Д. Никитина С.В. Клименко, И.Н. Никитин, *AVANGO. Система разработки виртуальных окружений*, Москва-Протвино, 2006.
- [8] Henrik Tramberend, *Avocado: A Distributed Virtual Environment Framework*, Ph.D. thesis, Technische Fakultät at Universität at Bielefeld, 2003.
- [9] Paul Martz, *OpenSceneGraph Quick Start Guide*, Skew Matrix, 2007.
- [10] “The economic payback of 3d mice for cad design engineers,” Tech. Rep., Technology Assessment Group, 2008.

Об авторах

Александр Бобков — аспирант Кафедры системной интеграции и менеджмента Факультета общей и прикладной физики Московского физико-технического института. Email: alexbobkov@list.ru

Илья Казанский — ассистент Кафедры системной интеграции и менеджмента Факультета общей и прикладной физики Московского физико-технического института. Email: kazansky@icpt.su

Станислав Клименко — профессор, доктор физ.-мат. наук, директор Института физико-технической информатики Email: stanislav.klimenko@gmail.com

Abstract

The article is about development of toolkit for displaying stereo presentations, containing georeferenced data. Stereo presentation is a presentation with the possibility of stereoscopic visualization. KML is used for data storage. Google Earth is used for data preparation. Google Earth allows to view satellite imagery and to work with georeferenced information. Toolkit allows to add custom data: tags, objects, images, virtual tours. This data is exported in KML-format.

The application is developed that allows work with stereo presentations based on data in KML-format and show these presentation on the installation of virtual reality. Libkml library is used for reading KML-files, AVANGO framework - for highlevel development.

КОМПЛЕКС ВИРТУАЛЬНОЙ ИМИТАЦИИ ЗУБОЧЕЛЮСТНОЙ СИСТЕМЫ

Смирнов А.Г.,¹ Клименко С.В.,² Ростков Д.А.¹

¹Институт компьютерного моделирования биологических объектов, Санкт-Петербург

²Институт физико-технической информатики, Протвино

alex.smirnov.spb@mail.ru, stanislav.klimenko@gmail.com, eport77@mail.ru

Аннотация

Создание систем виртуальной имитации биомеханики анатомических органов является многообещающим направлением в развитии диагностических и прогностических медицинских комплексов. Применимость таких систем зависит от способности обеспечивать быструю, точную и относительно дешевую с точки зрения вычислительных затрат генерацию компьютерной модели физического объекта (системы) и её визуализацию в форме интерактивной трехмерной графики. Для решения подобных задач уместно использовать методы реконструкции сложных высоко-детализированных трехмерных сцен за счет обработки потоков данных, получаемых от сенсоров. Такие методы имеют общее название Image Based Rendering (IBR) [1,2]. В представленной работе освещаются вопросы, связанные с разработкой диагностической виртуальной имитации зубочелюстной системы.

Ключевые слова: имитация, визуализация, артикулятор, окклюзия.

1 Введение

Разработка виртуальной системы имитационного моделирования зубочелюстной системы пациента напрямую связана с решением вопросов повышения качества функционального анализа окклюзии и артикуляции.

В диагностики стоматологической патологии ключевым моментом является исследование функции височно-нижнечелюстного сустава (ВНЧС), зубочелюстной системы и связанных с этим движений нижней челюсти, окклюзионного поля.

Перемещения нижней челюсти имеют сложную организацию и траекторию. Для изучения перемещений нижней челюсти, контактов зубов в разные фазы жевательных движений применяют имитационные приборы – артикуляторы. Наибольшее распространение имеют механические шарнирные устройства (Рис.1) различных типов и производителей.



Рис.1 - Общий вид механической диагностической системы (регулируемого артикулятора) с гипсовыми моделями.

На Рис.1 видно, что объектом изучения в механической системе являются гипсовые модели челюстей. Гипсовые модели устанавливаются в прибор в соответствии с допустимыми для каждого конкретного типа артикулятора возможностями воспроизведения индивидуальных гнатологических параметров. В свою очередь такие гнатологические показатели задаются некими абстракциями: угол сагиттального суставного пути и другие – для имитации движения нижней челюсти в сагиттальной плоскости (Рис.2);

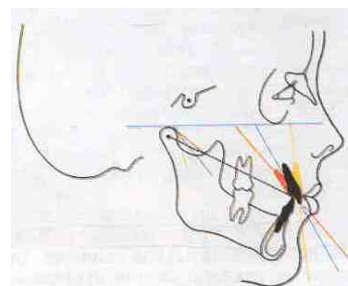


Рис. 2 – Изображение некоторых абстракций: горизонталей и углов, описывающих функцию движения нижней челюсти в сагиттальной плоскости.

угол Беннетта (Рис.3) – для имитации движения нижней челюсти в горизонтальной плоскости.

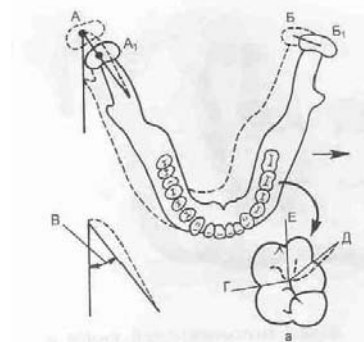


Рис. 3 – Движение нижней челюсти влево в горизонтальной плоскости. Пояснение: В – угол Беннетта.

В смоделированной системе непосредственно изучается не сам интересующий объект, а вспомогательная искусственная модель, находящаяся в некотором объективном соответствии с познаваемым объектом. [3] Механическая (аналоговая) имитационная система предоставляет исследователю возможность изучать динамику артикуляционных движений в условиях, когда:

а) биомеханические свойства искусственных моделей не равноценны свойствам объекта оригинала, например, на гипсовых моделях невозможно учитывать экскурсии зубов в зубоальвеолярной связке, возникающие при приложении к зубам сил при жевании, глотании;

б) определение вводимых абстракций сопряжено с обязательным выполнением некоторых технических условий и методик на специальных вспомогательных приборах;

в) артикуляторы различных конструкций с разной степенью достоверности имитируют по вводным данным артикуляционные движения.

Перечисленные выше аспекты на практике влияют на точность функционального анализа окклюзии и артикуляции [4].

В то же время специалисты нуждаются в объективных знаниях о контактах между зубами (Рис. 4).

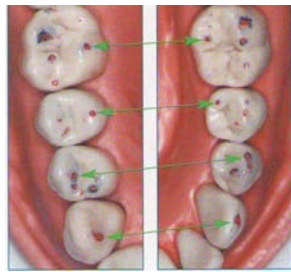


Рис. 4 - Контактные пункты на зубах.

Странники некоторых концепций окклюзии считают, что нужно придерживаться специфической модели множественных контактов для каждого зуба [5].

При преждевременных контактах скатов бугорков зубов, смещениях зубов, отсутствии бугорков или их нерациональных наклонах нарушаются нормальные бугорково-фиссурные соотношения зубов, их устойчивость, то есть нарушается гармоничная окклюзия. Последствия дефектов окклюзии отражаются на сосудисто-нервных механизмах пародонта. Значительные нарушения окклюзии сопряжены с явлениями мышечно-суставной дисфункции.

Недостаточная объективность механических (аналоговых) приборов (систем приборов) в части имитации динамических процессов в зубочелюстной системе подводит к идее разработки цифровых имитационных комплексов для решения задач расчета на компьютере артикуляционных движений, окклюзии и планирования лечебных конструкций на виртуальной модели. Логичность и формализованность компьютерных моделей позволяет выявить основные факторы, определяющие свойства изучаемого объекта, в частности, исследовать отклик моделируемой физической системы на изменение ее параметров и начальных условий. [3]

Цель разработки виртуальной имитации зубочелюстной системы – создание технологического комплекса, позволяющего повысить точность функционального анализа окклюзии и артикуляции. [6]

Основной задачей разрабатываемого комплекса является задача построения виртуальной имитации зубочелюстной системы пациента для проведения расчетных экспериментов на компьютере, цель которых – анализ жевательной функции, исследование отклика моделируемой системы на изменение ее параметров и начальных условий, сопоставление результатов моделирования с реальным поведением объекта-оригинала. [7]

2 Метод решения задачи

В компьютерном моделировании, в том числе для создания виртуальных интерактивных динамических моделей вещественных объектов (систем объектов) используются цифровые потоки данных, содержащие определенные сведения о физических объектах.

Получить необходимые цифровые данные на практике возможно с использованием сканирующих устройств. Такие методы реализуются с помощью компьютерной техники, поэтому любой из таких методов можно назвать компьютерным. Методика компьютерной визуализации математически описана в преобразовании Фурье и представляет собой математическое соотношение между реальным пространством и k -пространством. Применительно к визуализации в общем виде оно интерпретируется так:

$$S(\mathbf{k}) = \int d\mathbf{r} \rho(\mathbf{r}) \exp(i\mathbf{k}\mathbf{r}) \quad (1)$$

где k и g могут быть одно-, двух- или трехмерными, плотность $\rho(\mathbf{r})$ характеризует изображение в реальном пространстве, а $S(\mathbf{k})$ в k -пространстве, \mathbf{k} зависит от физических свойств, используемых в том или ином виде сканирования. Преобразование Фурье допускает обратное преобразование k -пространства в реальное g -пространство. [8]

Суть примененного в работе метода заключается в переносе цифровой информации об определенных гнатологических положениях, фиксируемых оптическим сканером (цифровая камера) на полигональную сетку трехмерной модели зубочелюстной системы пациента, полученную при помощи 3D лазерного сканера. На практике это выглядит, как наложение (проецирование) фотографической текстуры на трехмерную модель объекта. Используя синтезированные таким образом данные, при помощи алгоритмов твердотельного моделирования и реконструкции сложных высокодетализированных сцен, можно рассчитать и визуализировать траекторию движений нижней челюсти при жевании, определить возникающие нагрузки на зубы, выявить контактные точки и поля на окклюзионных поверхностях зубов с учетом экскурсий зубов в зубоальвеолярном аппарате. [9]

3 Принципы действия разработанных объектов, их характеристики

3.1 Полезная модель

Для реализации приведенного выше метода была разработана полезная модель «Программно-аппаратная система функционального анализа окклюзии и артикуляции» (патент РФ №80111). Полезная модель включает: блок сбора цифровых данных и средства организации интегрированной информационной среды,

объединяющей все процессы жизненного цикла системы. [9]

3.1.1 Блок сбора данных.

Блок сбора данных обеспечивает:

- а) сканирование гипсовых моделей исследуемых челюстей;
- б) построение полигональных сеток отсканированных моделей, формат STL (Рис. 5);



Рис.5

- в) сканирование (фотографирование) определенных окклюзионных точек на жевательных поверхностях зубов, формат JPG (Рис. 6).



Рис. 6

3.1.2 Средства организации интегрированной информационной среды.

Средства организации интегрированной информационной среды включают: ПК и программное обеспечение к программно-аппаратному комплексу окклюзии и артикуляции "Occlusion Navigator" (свидетельство о государственной регистрации № 2009616626).

Программное обеспечение к программно-аппаратному комплексу окклюзии и артикуляции "Occlusion Navigator" имеет следующие основные составляющие:

- а) пользовательский интерфейс;
- б) модуль визуализации:
 - подсистема адаптации данных сканирования;
 - компонент пространственного позиционирования челюстей;
 - библиотека OpenGL;
- в) подсистему численного моделирования жевательных движений н/ч, реакций зубов на силы, прилагаемые при окклюзии;
- г) подсистему расчета нагрузок на зубы;
- д) подсистему моделирования восстановительных конструкций (программный модуль подсистемы обратного (физического) моделирования по методу стереолитографии).

3.2 Модуль визуализации

Модуль визуализации выполняет основную функцию интегрированной информационной среды - построение виртуальной модели зубочелюстной системы конкретного пациента на основе потоков цифровых данных об объекте-оригинале, гипсовых моделях объекта-оригинала и определенных сведений о некоторых среднестатистических гнатологических показателях, включенных в базу данных.

В широком смысле компьютерная модель включает в себя все узлы, конечные элементы, свойства материала, реальные константы, граничные условия в зависимости от преобразования пространственной конфигурации объекта (системы), которые используются для отображения (имитации) физической системы. Построение виртуальной модели в данном контексте подразумевает визуализацию математической модели реального вещественного объекта (системы) в форме интерактивной компьютерной трехмерной графики.

3.2.1 Подсистема адаптации данных сканирования.

Поскольку речь идет о виртуальном моделировании динамической системы, необходимо было решить вопросы, связанные с адаптацией технологии визуализации в условиях данного технологического комплекса. [10]

Основным форматом для трехмерных моделей был выбран формат хранения результатов лазерного сканирования STL. Спецификация формата STL не поддерживает вхождение одной вершины в несколько полигонов, поэтому при загрузке данных модель автоматически конвертируется из формата STL в индексированный формат. Данное преобразование необходимо для корректной работы процедуры определения видимости полигонов на фотографии (depth map testing).

Для дальнейшего формирования текстурных координат (для точного наложения фотографической текстуры на сложную геометрию трехмерной модели, полученной в результате сканирования) был применен ручной режим.

В ручном режиме пользователь маркирует соответствующие точки на поверхности модели и на фотографии челюсти. По этим соответствиям (для однозначного определения положения камеры необходимо как минимум 4 соответствия) программа автоматически вычисляет проекционную матрицу камеры методом ресекции [11].

В процессе расчетов учитываются оптические искажения объектива камеры, которые были откалиброваны заранее с помощью программы автоматической калибровки объективов AgiSoft Lens. При этом по умолчанию определяются все коэффициенты матрицы внутренних параметров объектива, а также параметры дисторсии камеры (2 коэффициента радиальной составляющей дисторсии и 2 коэффициента тангенциальной составляющей).

В качестве альтернативного метода расчета параметров внешнего ориентирования фотокамеры, рассматривался метод стереозрения.

Преимуществом такого метода является полная автоматизация процедуры наложения фотографической текстуры на модель, а также более высокая точность наложения. Однако грубые геометрические модели, полученные методами стереозрения, обладают большим количеством шумовых дефектов, что в значительной степени осложняет работу алгоритмов

совмещения. Таким образом, этот подход выглядит перспективной заменой методу ручного совмещения.

3.2.2 Компонент пространственного позиционирования челюстей.

После наложения фотографической текстуры, содержащей данные о расположении окклюзионных точек и полей на геометрию трехмерных моделей челюстей антагонистов, полученных в результате сканирования, программа должна обеспечивать точное пространственное позиционирование моделей верхней и нижней челюстей относительно друг друга согласно специфике прикуса пациента.

Решение такой задачи требует:

- а) реализовать алгоритм определения положений точек окклюзии;
- б) реализовать алгоритм совмещения соответствующих точек окклюзии на трехмерных моделях челюстей антагонистов, при условии, что соответствия между совмещаемыми точками заранее не известны.

Действительно, фиксируя на фотографии зоны смыкания между зубами антагонистами, полученные в полости рта на зубах в виде отпечатка ультратонкой копировальной фольги, получаем достаточно обширные участки окрашивания. Для решения задачи выявления нахождения истинных контактных пунктов в зоне окрашивания от периферийного ореола следа копировальной фольги – ложных зон – необходимо наложить ограничение.

Для точного определения положений точек окклюзии был адаптирован алгоритм поиска особых точек на основе детерминанта матрицы Гессе, известный своей устойчивостью в составе детектора SURF. В процессе работы алгоритм находит экстремумы детерминанта в пространстве масштаба и координат. Найденные экстремумы затем интерполируются с субпиксельной точностью, как по координатам, так и по масштабу, что обеспечивает более точное определение положений и размеров окклюзионных зон (Рис. 7).



Рис. 7 - Результат работы детектора поиска окклюзионных зон.

Для фильтрации результатов работы детектора, найденные зоны анализируются по интенсивности и цвету, что в результате приводит к определению действительных точек окклюзии (Рис.8).



Рис.8 - Результат обработки текстуры с фильтрацией.

Пояснение: красным цветом выделены зоны окклюзии, оставшиеся после фильтрации.

Несмотря на то, что результат обработки выглядит вполне удовлетворительным, остается вероятность появления ложных зон окклюзии. Для надежной работы алгоритма совмещения челюстей, необходимым подход, который будет устойчив к ложным срабатываниям детектора. В качестве такого подхода был выбран метод, похожий на известный алгоритм RANSAC, отличающийся резистентностью к ошибкам во входных данных.

Совмещение соответствующих точек окклюзии реализуется за счет поиска максимального подмножества соответствий, удовлетворяющего преобразованию из заданного класса. Допустимым классом преобразований является поворот и перенос в трехмерном пространстве. Таким образом, задача совмещения сводится к поиску такого преобразования из заданного класса, при котором наибольшее количество пар найденных окклюзионных точек придет в соприкосновение друг с другом. В нашем случае задача свелась к поиску всех возможных преобразований по некоторым 3 парам соответствий, и выбору из них того преобразования, которое обеспечивает соответствие наибольшего количества из оставшихся пар. Поскольку количество окклюзионных зон относительно мало, метод полного перебора по всем возможным тройкам соответствий достаточно эффективен.

Оба, описанных выше алгоритма составляют компонент пространственного позиционирования челюстей в модуле визуализации. На Рис.9 показан результат совмещения челюстей с помощью компонента пространственного позиционирования в среде "Occlusion Navigator".

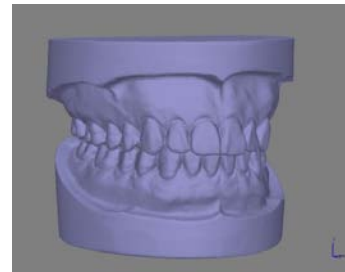


Рис. 9

3.3 Моделирование жевательных движений нижней челюсти

Виртуальная динамическая имитация зубочелюстной системы предполагает наличие подсистемы численного моделирования жевательных движений нижней челюсти, реакций зубов на силы, прилагаемые при окклюзии и визуализацию такого математического решения.

Общая траектория движений нижней челюсти описывается набором точек в пространстве согласно индивидуальному характеру максимальных положений нижней челюсти.

Справедливо утверждать, что, выбрав любую точку нижней челюсти на полигональной сетке виртуальной модели, можно проецировать ее перемещение на одну или другую плоскость и смоделировать таким образом характер индивидуальных движений максимальной амплитуды, т.е. характер «пограничных» движений. [12] Отсюда, математическая модель движений

нижней челюсти представляет собой алгоритм вычисления пространственных положений выбранной опытной точки на полигональной сетке трехмерной модели в различные фазы артикуляции.

На полигональной сетке нижней челюсти выбираем опытную точку – срединную точку нижних резцов (стнр). Определяем положение этой точки в трехмерном виртуальном пространстве при определенных положениях модели нижней челюсти.

Впервые все разнообразие движений нижней челюсти в трех плоскостях, прослеживая траекторию движения нижних резцов описал в 1952 г. Posselt.

Траектория движений нижней челюсти зависит от строения и физиологии ВНЧС. С точки зрения механики ВНЧС является шарниром, поэтому движения нижней челюсти в пространстве происходят дугообразно в сагиттальной (вертикальной), горизонтальной и фронтальной плоскостях (Рис. 10).

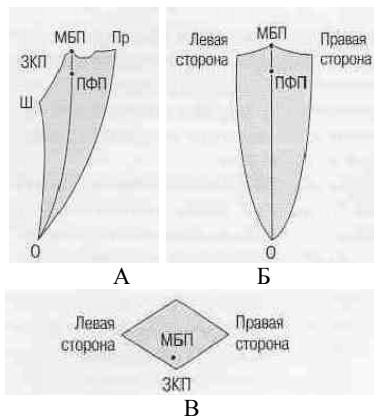


Рис. 10 – Схема пограничных движений нижней челюсти

Пояснения:

А – движения в сагиттальной плоскости;

Б – движения во фронтальной плоскости;

В – движения в горизонтальной плоскости;

ЗКП – положение центрального соотношения челюстей;

МБП - положение Центральной Окклюзии (ЦО);

ПФП - положение физиологического покоя;

Пр - крайнее переднее положение нижней челюсти;

О – положение нижней челюсти при максимально открытом рте;

ЗКП – Ш – терминальная дуга открывания и закрывания рта.

Для решения задачи виртуального моделирования артикуляционных движений нижней челюсти введем классификацию всех жевательных движений на:

а) подводящие движения - движения сближения нижней челюсти с верхней в отсутствие окклюзионного контакта;

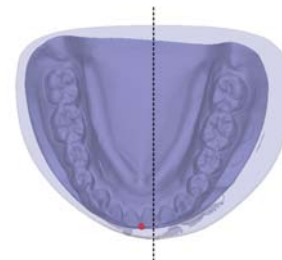
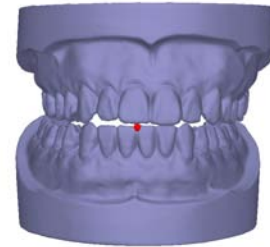
б) окклюзионных движений – движения, для которых характерно скольжение зубов по окклюзионному полю, другими словами - движения нижней челюсти при контакте между зубами антагонистами;

в) отводящие движения – движения разобщающие челюсти в отсутствие окклюзионного контакта.

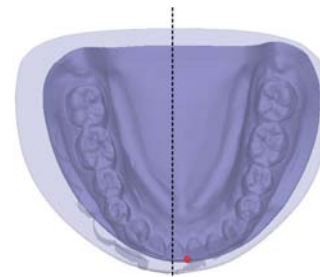
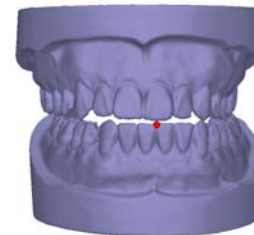
3.3.1 Моделирование подводящих движений.

Моделирование подводящих движений основано на вычислении результирующей подводящей траектории, которая, в свою очередь, складывается из вычисления дуг движения нижней челюсти в сагиттальной, фронтальной и горизонтальной плоскостях.

На Рис.11 приведены примеры изменения положения стнр в правой, левой боковой окклюзии.



А



Б

Рис. 11

Пояснение:

А – место нахождения стнр в положении челюстей в правой боковой окклюзии;

Б – место нахождения стнр в положении челюстей в левой боковой окклюзии.

3.3.2 Моделирование окклюзионных движений нижней челюсти.

Эти движения соответствует перемещению н/ч из боковой окклюзии в положение ЦО (когда челюсти сжаты таким образом, что не могут совершать боковых движений) и далее из положения ЦО вперед и в сторону размыкания зубов. Форма контура такого перемещения определяется ведущими зубами.

В описании полезной модели системы функционального анализа окклюзии и артикуляции [9] определено, что фотографическая текстура, содержит цифровую информацию об определенных точках на

окклюзионных поверхностях зубов. Эти точки определяются врачом в полости рта по общепринятой методике клинического обследования окклюзии. Таким образом, реализуется метод внутриротовой аксиографии, когда на жевательных поверхностях зубов с помощью копировальной бумаги «расчерчиваются» пути ведения и контакты между зубами в определенных фазах жевания.

Вычисление окклюзионной траектории движений нижней челюсти (Рис. 12) производится методом интерполяции между контрольными точками, с учетом участка траектории, где наблюдается контакт между зубами верхней и нижней челюстей.

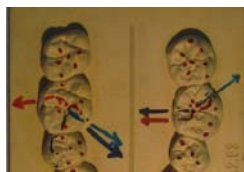


Рис. 12

На таком участке траектория нижней челюсти определяется геометрией коронковых частей зубов, находящихся в контакте друг с другом. Т.е. путь ведения задает кривизну траектории, которая обусловлена кривизной отмеченного участка - кривизной поверхности зуба, на которой определен путь ведения.

3.3.3 Моделирование отводящих движений нижней челюсти.

Моделируются отводящие движения трех типов:

- подготовка к жеванию;
- во время рабочего цикла;
- окончание жевания.

3.4 Моделирование реакций зубов на прилагаемые окклюзионные силы

Для того чтобы решить задачу моделирования реакций зубов на прилагаемые окклюзионные силы необходимо сегментировать цифровой массив трехмерной полигональной сетки модели челюсти на определенные потоки цифровых данных, соответствующие отдельным зубам, а затем задать каждому зубу локальную систему координат.

3.4.1 Алгоритм сегментации трехмерной модели челюсти.

Первоначально был разработан алгоритм, способной сегментировать цифровой массив трехмерной полигональной сетки модели челюсти на определенные потоки цифровых данных, соответствующие отдельным зубам, справедливый для любых форм зубов.

3.4.2 Локальная система координат зуба.

Теперь стало возможным задать для каждого зуба локальную систему координат. В качестве локальной системы координат справедливо использовать декартову прямоугольную систему координат.

Координатное поле зуба (Рис. 13):

- начало координат – находится в месте пересечения линий соединяющих крайние медиально-дистальную точки и вестибулярно - оральную точки на плоскости прилегания эмали зуба к десневому краю или в случае оголения в области шейки зуба;
- ось OY – продольная ось зуба – вдоль самого большого размера длины зуба;

- ось OX – в медио-дистальном направлении;
- ось OZ – в вестибулярно – оральном направлении.

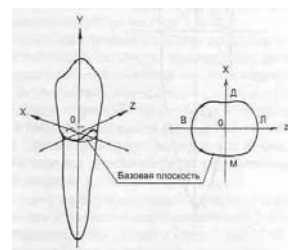


Рис. 13 – Декартова система координат для нижнего правого клыка. [13]

3.4.3 Моделирование реакций зубов.

Имея сегментированную модель челюстей и заданную для каждого зуба систему координат, можно переходить к описанию реакций зубов на окклюзионные нагрузки.

Биомеханические реакции зуба определяются строением тканей пародонта. Зуб в челюсти «установлен» не жестко, а удерживается зубоальвеолярной связкой, имеющей определенное строение. Таким образом, с точки зрения механики система зуб – челюсть имеет шесть степеней свободы.

Методика моделирования реакций зубов на окклюзионные силы построена на:

- вычислении центра сопротивления зуба;
- данных о точке приложения силы;
- разложении вектора силы, развиваемой жевательными мышцами при смыкании зубов (окклюзии), на составляющие;
- значениях нормальных перемещений зубов в горизонтальной и вертикальной плоскостях;
- вычислении центра вращения зуба.

3.4.3.1 Моделирование реакций зубов необходимо производить относительно центра сопротивления зуба. Центр сопротивления зуба - точка, через которую проходит равнодействующая всех сил реакции корня зуба, расположенного в кости.

Расчет центра сопротивления зуба происходит исходя из знаний о том, что центр сопротивления находится на продольной оси однокорневых зубов – посередине длины корня, на продольной оси зуба многокорневых зубов в области бифуркации корней [14].

3.4.3.2 Для дальнейшего моделирования примем за основу концепцию биомеханики зубов по Шварцу (Рис. 14)[14].

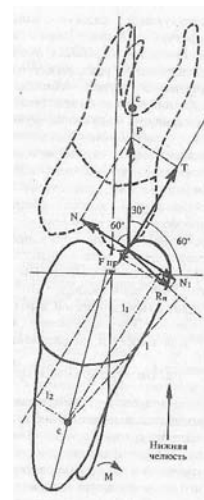


Рис. 14 – Изображение модели реакций боковых зубов.

Пояснение:

c – центр сопротивления зуба;

P – вектор действующей силы – силы мышечного сокращения;

T – вектор сдвигающего, скользящего усилия (сила скольжения по поверхности, образованной контактирующими скатами бугров);

N – вектор силы давления на пищевой комок и скат;

$N1$ – вектор нормальной реакции зуба (без пищевого комка);

$F_{тр}$ – вектор силы трения пищевого комка;

R_n – вектор результирующей силы или вектор полной реакции зуба указывает направление склонения зуба;

угол между вектором $N1$ и R_n есть поправка на вектор силы трения пищевого комка, принимаем за 8,5 градусов [14].

Для начала моделирования ответной реакции зуба на нагрузку при окклюзии определим направление склонения зуба, которое соответствует R_n (Рис. 14) в любой точке приложения силы на скате бугра.

Далее учитывается расстояние склонения зуба в горизонтальной плоскости от продольной оси зуба. Для чего следует задать зависимость расстояния перемещения зуба в сторону склонения в зависимости от величины прилагаемой силы.

При дальнейшем моделировании следует принять во внимание, что склонение зуба (горизонтальное перемещение) происходит не только трансляционно, но и вращательно относительно центра вращения зуба (Рис. 15).

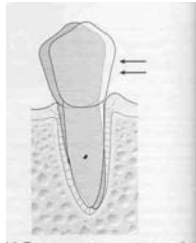


Рис. 15

Для этого необходимо рассчитать дугу, по которой вращается (склоняется) зуб.

Таким образом, пространственное положение зуба при горизонтальном смещении определяется:

- направлением склонения зуба;
- расстоянием смещения от продольной оси зуба при нормальной нагрузке;
- радиусом дуги склонения.

По мере скольжения зубов в положение максимального смыкания (скольжение к центру) происходит поступательное смещение центра вращения из центра сопротивления к верхушке корня (Рис. 16).

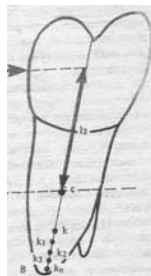


Рис. 16

Пояснение:

c – центр сопротивления;

$k...k_n$ – центры вращения (по Jacobson).

3.5 Визуализация полученных математических моделей

Базовой технологией для модуля визуализации была выбрана библиотека 3-мерного моделирования OpenGL. Она является кроссплатформенным решением и оптимизирована под задачи технического моделирования. Кроме того рассматривается возможность использования графического API OpenGL Volumizer. OpenGL Volumizer обеспечивает интерфейс высокого уровня для OpenGL, разрешает аппаратный рендеринг объемных данных – до нескольких гигабайт [15].

Формирование участка траектории в области контакта является достаточно ресурсоемкой задачей, т.к. с одной стороны, следует высчитывать кривизну пути ведения по кривизне соответствующей поверхности зуба, а с другой, результирующую реакцию зуба на окклюзионную нагрузку. Использование ресурсов графического процессора для выявления зон контакта и корректировки траектории по касательной к контактирующим поверхностям позволило:

- избавиться от процедуры предварительного просчета всех точек траектории;
 - корректировать положения контрольных точек или амплитуды движения челюсти "на лету".
- Такой подход обеспечил приемлемую скорость расчета траектории, что сделало возможным визуализировать окклюзионное движение в реальном времени, непосредственно в процессе движения.

Модуль визуализации поддерживает одновременное отображение нескольких моделей, как с наложенными текстурами, так и без.

Разрабатываются перспективы адаптации системы виртуальной имитации зубочелюстной системы к среде виртуального окружения АВАНГО. [16]

4 Сравнение с аналогичными системами

Актуальность разработки имитационного комплекса зубочелюстной системы с применением IBR методов подтверждается работами, которые ведутся специалистами компании Sirona (герм.) и некоторых других. Разработанная фирмой Sirona система Ceresc [17], построена на принципе обработки цифровых данных, полученных в результате оптического сканирования гипсовых моделей челюстей пациента или жевательных поверхностей зубов.

Остановимся подробнее на сравнении виртуальных моделей в Ceresc и Программно-аппаратной системе функционального анализа окклюзии и артикуляции (Табл.1).

Таблица 1.

	Алгоритм сегментации модели челюсти	Математическая модель экскурсий зубов	Учет индивидуальных положений путей ведения	Учет индивидуальных положений точек окклюзии
Сегес	нет	нет	нет	В положении ЦО
Программно-аппаратная система	есть	есть	есть	В положении ЦО; В положении боковых окклюзий; В положении размыкания зубов

Сведения, приведенные в Табл.1 позволяют говорить, что модель в Сегес может передавать биомеханические свойства объекта-оригинала с меньшей точностью, чем модель в Программно-аппаратной системе.

Следует также отметить, что в Программно-аппаратном комплексе впервые широко реализован способ внутриворотной аксиографии. Этот способ стал актуален благодаря применению IBR методов. На практике такое решение позволяет отказаться от вычисления многих вводных абстракций, связанных с положением элементов ВНЧС в пространстве, что значительно упрощает моделирование динамической имитации зубочелюстной системы.

5 Заключение

Комплекс виртуальной имитации зубочелюстной системы отличается тем, что предлагает потребителю (врачу, зубному технику) возможность проводить исследования виртуальной биомеханической модели зубочелюстной системы конкретного пациента с возможностью компьютерной обработки информации и интерактивного взаимодействия с таким объектом, в том числе - возможности ввода экспериментальных данных для последующих расчетов на модели.

При этом точность расчетов обеспечивается за счет:

- разрешающих возможностей сканирующих устройств и сенсоров, которые фиксируют геометрические параметры окклюзионного поля объекта-оригинала и контрольные точки окклюзии;
- включения в систему расчетов модели параметров экскурсий зубов при различных направлениях и силе нагрузок во время окклюзионных движений;
- алгоритмов твердотельного моделирования и реконструкции сложных высоко-детализированных сцен.

6 Литература

[1] L.Mc.Millan, An Image-Based Approach to Three-Dimensional Computer Graphics, Ph.D. Dissertation, UNC Computer Science Technical Report TR97-013, University of North Carolina, April 1997

[2] Клименко С.В., Никитин И.Н., Никитина Л.Д. АВАНГО: система разработки виртуальных

окружений. Москва – Протвино, Институт физико-технической информатики, 2006. - 252 с.

[3] Новик И.Б., О философских вопросах кибернетического моделирования. Знание, 1964.

[4] Бобров А.П., Смирнов А.Г., Падалко Е.В., Бя С.Ч., Суслов Д.П., Применение в клинической практике артикуляторов ASA Dental, Hager Werken, SAM 3 – сравнительный. Институт Стоматологии.- 2008. - №1(38). – С.126-127.

[5] Iven Klineberg R. G. Jagger. Occlusion and Clinical Practice: An Evidence-Based Approach. Published 2004-04-02, pp. 160

[6] Смирнов А.Г., Бобров А.П., Казанский И.П., Булгаков М.Г., Программно-аппаратный метод анализа окклюзии и артикуляции: общие контуры решения. Электронный многопредметный научный журнал "Исследовано в России". – 2008. – С.1110-1116. <http://zhurnal.ape.relarn.ru/articles/2008/101.pdf>

[7] Бобров А.П., Смирнов А.Г., Бельтюков П.П., Суслов Д.П., Нуштайкин А.С., Программно-аппаратный способ построения виртуальной имитации зубочелюстной системы пациента. Пародонтология. – 2009. - №4(53). – С.57-62.

[8] Мэнсфилд П. Быстрая магнитно-резонансная томография. Успехи физических наук. – 2005. - Т. 175, №10. - С. 1044—1052.

[9] Смирнов А.Г., Пат. 80111 Российская Федерация, МПК А61С 7/00. Программно-аппаратная система функционального анализа окклюзии и артикуляции № 2008138078/22 ; заявл. 24.09.2008 ; опубл. 27.01.2009, Бюл. № 3.

[10] Smirnov A., Kazansky I., IBR (Image Based Rendering) complex for virtual simulation dental-jaw system. Труды международной научной конференции Medias-Dialog. – 2009. – Т.2 – С.62-64.

[11] Hartley, R.I. and Zisserman, A., "Multiple View Geometry in Computer Vision", Second edition, Cambridge University Press, 2004.

[12] Хватова В.А. Клиническая гнатология. М.: ОАО «Издательство «Медицина», 2005. - 296 с.: ил

[13] Чуйко А.Н., Вовк В.Е., Особенности биомеханики в стоматологии: Монография. Х.: Прапор, 2006. – 304 с.

[14] Шварц А.Д. Биомеханика и окклюзия зубов. М.: Медицина, 1994. - 208 с. ил. 76

[15] <http://www.sgi.com/products/software/volumizer/>

[16] Казанский И.П., Смирнов А.Г. Система виртуальной имитации зубочелюстной системы пациента. Труды 52-й научной конференции МФТИ «Современные проблемы фундаментальных и прикладных наук». – 2009. - Часть VII. Управление и прикладная математика. - Том 2. – С.18-20.

[17] <http://www.cereconline.com/cerec/demo.html>

Алгоритм автоматического выделения лица на термографических изображениях

Наталья Баша, Людмила Шульга

ФГУП «Научно-исследовательский институт прикладной акустики», Центр высоких технологий, г. Дубна, Россия,

Международный Университет природы, общества и человека «Дубна», г. Дубна, Россия

{natalia.basha, luda.shulga}@niipa.ru

Аннотация

В статье представлен подход к исследованию термографических изображений человека для интеллектуального видеонаблюдения. В работе предложен алгоритм автоматического выделения лица в инфракрасном спектре излучения. Приведены результаты его работы и проведен анализ эффективности его на базе данных 103 термографических портретов 15 человек.

Ключевые слова: распознавание образов, анализ изображений, системы видеонаблюдения, термография, детекция лица.

1. ВВЕДЕНИЕ

Тепловидение является одной из самых перспективных наукоемких технологий XXI века. Являясь дальнейшим развитием приборов и систем технического зрения, тепловизоры в перспективе будут иметь более широкое распространение, чем предыдущее поколение, ориентированное на видимый диапазон. Термографический образ (термограмма) – изображение объекта, полученное камерой, улавливающей инфракрасное излучение объекта (человека). Для получения термографических изображений используются специальные тепловизионные камеры (или тепловизоры). Различные системы диагностики и контроля на базе тепловизора нашли широкое применение в промышленности и технике (неразрушающий контроль, техническая и медицинская диагностика), в силу того, что применение тепловизоров абсолютно безопасно и дает практически готовый анализ состояния оборудования (технических систем, кровоснабжения частей человеческого организма) уже на момент проведения инфракрасной диагностики.

Наряду с видеонаблюдением, тепловизионное наблюдение так же является бесконтактным методом. Существенными преимуществами его использования для автоматической идентификации тепловизионной камеры по сравнению с видеокамерой являются:

- нечувствительность к освещенности объекта и способность работать в полной темноте;
- способность давать вполне приемлемое для опознавания изображение даже при значительном удалении от человека;
- нечувствительность к внешней маскировке (например, элементам макияжа).

Эти отличительные черты позволяют применять тепловидение в условиях, когда получение изображения с видеокамеры недостаточно для реализации поставленных целей [1,2].

Хотелось бы отметить, что работы, связанные с задачами распознавания лиц в ближайшем ИК-диапазоне, ведутся последние 10 лет, однако возможность применять термограф для данного рода исследований появилась недавно. Отличие заключается в том, что данные, получаемые с ИК-камеры – это изображение, а получаемые с термографа – температурные карты, где графическое представление в цвете используется только для визуализации разницы температур.

Информационными признаками в термографии являются подкожные рисунки артерий и вен, которые являются уникальными и неизменными для каждого человека, так как сосудистый рисунок человека не подвергается изменениям ни при изменении температуры лица, ни при пластических операциях, ни от фактора старения человека. Это еще один довод, почему задача использования термографических изображений в системах интеллектуального видеонаблюдения является актуальной.

2. МЕТОДЫ ИССЛЕДОВАНИЯ

В работе был исследован вопрос автоматического выделения лиц на термографических изображениях. Для этого было проведено исследование с целью определения на лице человека стабильной температурной области, которую можно использовать как область интереса при создании индивидуального температурного портрета человека.

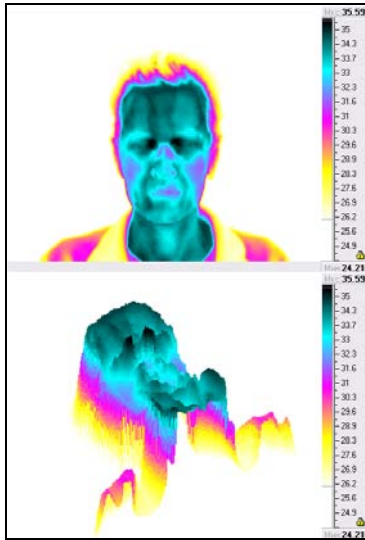


Рис. 1: Изображение лица, полученное тепловизионной камерой: сверху – двумерное изображение термограммы лица, снизу – 3D отображение термограммы лица; вертикальная шкала отображает цветом температуру объекта в градусах Цельсия.

Термографические данные (рис. 1) были получены термографом «ИРТИС-2000МЕ», представляют собой матрицу температур размером 240×320, с точностью изменения температур 0,01°С. База термографических данных состоит из 103 снимков 15 людей разного возраста, пола и телосложения. Термограммы получены в разное время суток, при разных условиях окружающей среды (в помещении и на улице). Сбор данных проводился еженедельно в течение 4 месяцев.

В результате исследования было выяснено, что внутренний угол глаза здорового человека дает отклик в ИК-диапазоне наиболее интенсивно (рис. 1, 2а).



Рис. 2: Изображения лица, полученное тепловизионной камерой: а) гипертермия области внутренних углов глаз (норма); б) симметричная гипертермия в области ключиц, на границе с одеждой; в) гипертермия в области носа и рта; г) гипертермия в области лба, частично разрываемая волосами; вертикальная шкала отображает температуру объекта в градусах Цельсия.

Экспериментально были замечены некоторые отклонения от общей тенденции. Основные причины: симметричные зоны гипертермии, связанные с воспалительными процессами, вегетосудистой дистонией, гипертермией на границе открытых и закрытых участков тела (рис. 2 б, в, г).

Результаты наблюдений были учтены при разработке алгоритма детекции лица.

Поэтому в качестве опорных точек области интереса были выбраны точки внутренних углов глаза. Использование данных меток эффективно потому, что с их помощью по коэффициентам пропорциональности на лице можно вычислить размеры самого лица и расположение основных деталей (рис. 3) [3].

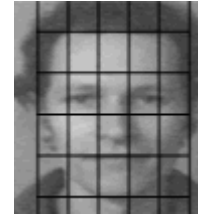


Рис. 3: Пропорции лица взрослого человека.

В настоящей статье предложен и разработан подход автоматического выделения области лица, базирующийся на детекции внутренних углов глаз.

Анализируя тепловые профили, соответствующие зоне внутренних углов глаз, набора термограмм из базы данных, было установлено, что данные точки на профиле представляются в виде двух четко выраженных пиков, симметричных относительно срединной линии лица. Тепловой профиль по строке, соответствующей зоне внутренних углов глаз, приведен на рис. 4.

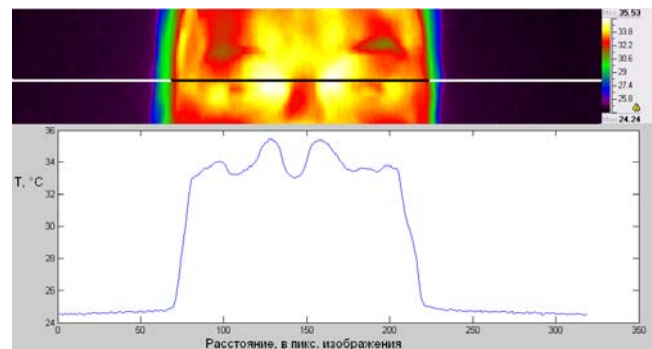


Рис. 4: Тепловой профиль по строке, соответствующей зоне внутренних углов глаз: сверху – часть исходного изображения термограммы лица (линией выделена строка, содержащая внутренние углы глаз), снизу – соответствующий ей тепловой профиль.

По результатам данных термограмм разработан алгоритм выделения области лица. Он состоит из нескольких этапов.

Первый этап: пороговая фильтрация

Первым этапом является предварительная обработка изображения методом пороговой фильтрации. Экспериментально было установлено значение этого порога *threshold*, равное $\frac{min+max}{3}$, где *min* и *max* –

минимальное и максимальное, соответственно, значения температур, соответствующих человеку. Это значение соответствует температуре кожи и не включает одежду и волосы. Для каждой строки изображения находится левая и правая границы области, подозрительной на лицо, а так же точка *top_head* для всего изображения, соответствующая верхней точке потенциальной области лица.

Второй этап: вычисление интервала для нахождения линии глаз

Вторым этапом вычисляется мода для левой и правой границы области, подозрительной на лицо, мода для средней линии области, *moda_middle_line* (как мода середины отрезка между правой и левой границей каждой конкретной строки выделенной области изображения) и мода ширины области, *moda_width* (как мода разности между правой и левой границей каждой конкретной строки выделенной области изображения). Экспериментально было установлено, что значение моды ширины области, подозрительной на лицо, соответствует ширине реального лица на уровне глаз).

Исходя из пропорций лица, значения верхней точки головы и значения средней ширины рассчитывается значение линии глаз *eye_line*:

$$eye_line = top_head + round(0.68 * moda_width).$$

Экспериментально было установлено, что для более надежного определения глаз, нужно задать некий интервал *borders* для «попадания» глаз.

Таким образом, выделяется интервал, находящийся между значениями $[eye_line - borders; eye_line + borders]$.

Третий этап: поиск реальной линии глаз

Было экспериментально установлено, что глаза являются зонами гипертермии. Поэтому на выделенном интервале ведется поиск значения максимума температуры. Затем осуществляется проход маской с двумя пиками всего выделенного интервала. Среди строк, удовлетворяющих условиям, выбирается строка, где:

- Значение одного из «пиков» равно максимуму на интервале;
- «Пики» симметрично расположены относительно средней линии;
- «Пики» симметрично расположены относительно соответствующих границ лица;
- «Впадина» расположена на средней линии;
- Разница температур между «пиком» и «впадиной» больше порогового значения (экспериментально было установлено пороговое значение, равное 0,5 °C).

Выделенная строка является линией глаз.

Четвертый этап: выделение лица

На изображении расставляются маркеры глаз. Вычисляется ширина области, подозрительной на лицо, на уровне глаз и по ней рассчитывается верхний угол прямоугольника, заключающего потенциальное лицо, его длина и ширина.

Выделенная таким образом область принимается за лицо.

Предложенный метод был проверен на базе из 103 изображений 15 человек. Результат выделения лица продемонстрирован на рис. 5. По данной выборке процент верного выделения лиц составил 98%, анализ ошибочно детектированных изображений показал, что, ошибка была связана с размытостью исходной термограммы (по причине движения объекта во время съемки). В экспериментальных исследованиях было установлено, что на качество детекции лица, для конкретно взятого человека, не влияют ни изменение прически и волосяного покрова лица (наличия и отсутствия бороды, усов), температурные условия сбора изображений (сбор проводился в помещении и зимой на открытом воздухе, и в обоих случаях выделение проходило успешно).

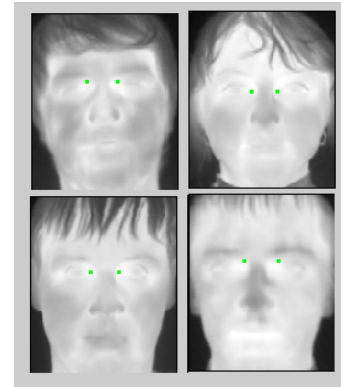


Рис. 5: Результат работы алгоритма автоматического выделения области лица.

Отдельно стоит отметить, что предложенный алгоритм демонстрирует свое быстроедействие: среднее время выделения одного лица составляет 2,07с (в среде MatLab).

С целью расширения класса термограмм, на которых применим разработанный метод выделения лица, приведем результаты детекции для термограмм, полученных в разных условиях. На рис. 6а, представлен термографический снимок, где лицо конкурирует с другими объектами. Для обработки этого термоснимка, сначала производится операция сегментирования областей, похожих на лицо (выделены 2 области: «лицо» и «чайник», находящиеся в одном температурном диапазоне и схожей площади). Далее применяется разработанный алгоритм выделения лица. Объект «чайник» был отбракован на третьем этапе: не были найдены строки, удовлетворяющие условиям линии глаз. Результат детекции лица представлен на рис. 6в.

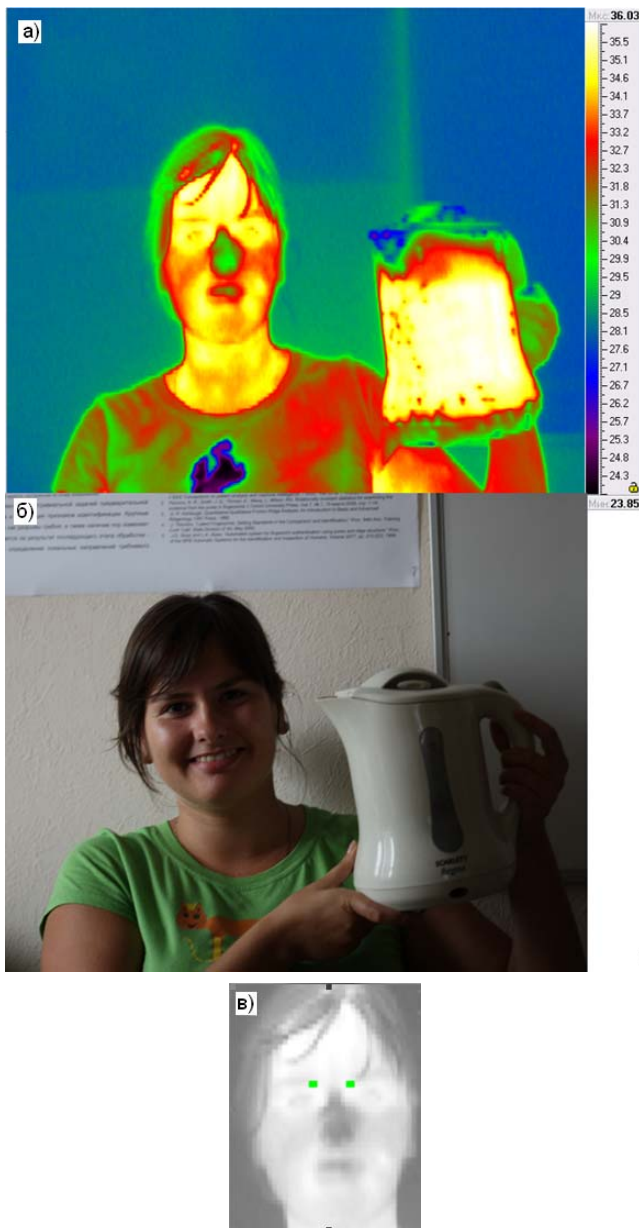


Рис. 6: Результат работы алгоритма автоматического выделения области лица: а) температурная карта, б) соответствующее ей изображение, в) выделенное лицо.

3. ВЫВОДЫ

Предложен алгоритм автоматического выделения лица на термографических изображениях. Описанный алгоритм базируется на поиске внутренних углов глаз, которые являются стабильной гипертермической областью на лице человека. Экспериментальные результаты показали, что выделенные предложенным алгоритмом точки термограммы, соответствуют внутренним углам глаз.

Стоит отметить, что данный алгоритм можно использовать как алгоритм автоматического определения присутствия человека в кадре, скажем, такая задача решается в системах охраны периметра.

В дальнейшем рассмотренный метод можно применять в задачах идентификации личности и построения интеллектуальных систем видеонаблюдения.

4. ЛИТЕРАТУРА

1 Evans, David C. «Infrared Facial Recognition Technology Being Pushed Toward Emerging Applications», Proc. SPIE Vol. 2962, 1997, p. 276-286.

2 Иваницкий Г.Р. «Современное матричное тепловидение в биомедицине», Успехи физических наук, т. 176, № 12, 2006, с. 1293-1320.

3 Куприянов В. В., Стовичек Г. В. Лицо человека: Анатомия, мимика. — М.: Медицина, 1988.

Сведения об авторах

Наталья Сергеевна Баша, аспирант кафедры Системного Анализа и Управления Международного Университета природы, общества и человека «Дубна», инженер-программист Центра высоких технологий ФГУП «Научно-исследовательского института прикладной акустики».

Людмила Александровна Шульга, кандидат технических наук, начальник лаборатории Центра высоких технологий ФГУП «Научно-исследовательского института прикладной акустики».

Алгоритмы биометрической идентификации личности человека на основе инвариантных признаков профильного изображения лица

Мумтозали Тухтаинов
Факультет информатики

Наманганский инженерно - педагогический институт, Наманган, Узбекистан
mumtozali@yahoo.com

Аннотация

В данной работе предложены алгоритмы биометрической идентификации личности человека на основе инвариантных признаков профильного изображения лица. Локализация профиля лица и лицевых элементов на изображении осуществляется с помощью традиционных методов, основанных на анализе цвета кожи и интегральных проекций. Для определения идентификационных признаков профильного изображения лица предложены новые подходы, с помощью которых можно сформулировать пространства инвариантных признаков. На основе корреляционного и геометрического подхода выполнено само распознавание.

Ключевые слова: биометрия, идентификация, распознавание, профильное изображение лица, инвариантные признаки, корреляция, геометрия.

1. ВВЕДЕНИЕ

Известно, что количество научных исследований в области биометрической идентификации личности человека достаточно велико; наиболее популярной темой является идентификация человека по изображению лица [1-6].

Необходимо отметить, что до настоящего времени исследования в данной области проводились в основном для фронтальных изображений лиц. Но теперь появляются научные и практические задачи исследования профильного изображения лица, которые необходимо решать [5, 6].

Для решения поставленной задачи, необходимо решить следующие подзадачи.

- локализация профильного изображения лица и лицевых элементов на изображении;
- формирование признакового пространства;
- распознавание профильного изображения лица на основе идентификационных признаков.

В данной работе предложены алгоритмы биометрической идентификации личности человека на основе инвариантных признаков профильного изображения лица. Локализация профильного изображения лица и лицевых элементов на изображении осуществляется с помощью традиционных методов, основанных на анализе цвета кожи и интегральных проекций. Для определения идентификационных признаков профильного изображения лица предложены новые подходы, с помощью которых можно сформулировать пространства инвариантных признаков. На основе корреляционного и геометрического подхода выполнено само распознавание.

2. ЛОКАЛИЗАЦИЯ ПРОФИЛЯ ЛИЦА НА ИЗОБРАЖЕНИИ

На рис.1. приведен общий вид цветного изображения профиля лица, на котором необходимо определить области лица для дальнейшего выполнения поставленной задачи.



Рисунок 1: Цветное изображение профиля лица.

Для локализации лицевой части можно использовать метод, основанный на анализе цвета кожи в цветовой модели RGB. С помощью нижеследующих предикатов можно выделить области кожи на цветном изображении:

$$R > 95 \text{ and } G > 40 \text{ and } B > 20 \text{ and } \max(R, G, B) - \min(R, G, B) > 15 \\ \text{ and } |R - G| > 15 \text{ and } R > G \text{ and } R > B \text{ and} \\ [(R * 100) / (R + G + B)] < 57 \text{ and } [(G * 100) / (R + G + B)] < 35 \text{ and} \\ [(B * 100) / (R + G + B)] < 35.$$

Здесь R, G, B – значения каналов проверяемого пикселя, соответствующие красному, зеленому и голубому цветам.

Результат алгоритма показан на Рис.2а.

Для определения лицевой части, выполнен анализ интегральных проекций на области цвета кожи. Алгоритм состоит из следующих шагов.

1-шаг. По формуле $IP = \sum I_{skin}$ вычисляются вертикальные и горизонтальные интегральные проекции для пикселей, содержащие цвет кожи (Рис.2б, в).

2-шаг. Определяются максимальные значения IP_{Max} от вертикальных и горизонтальных интегральных проекций. Результативные линии изображены красным цветом (Рис.2б, в).

3-шаг. Определяются координаты (области) профильного изображения лица, удовлетворяющие условию $IP > \frac{IP_{Max}}{5}$.

Результативные линии от вертикальных и горизонтальных интегральных проекций выделены синим цветом. Остальные линии выделены черным цветом (Рис.2б, в).

Граничные (верхние, нижнее, левые и правые) координаты, т.е. общий результат локализации профильного изображения лица, проиллюстрирован на Рис.2а пресекающимися линиями зелёного цвета.

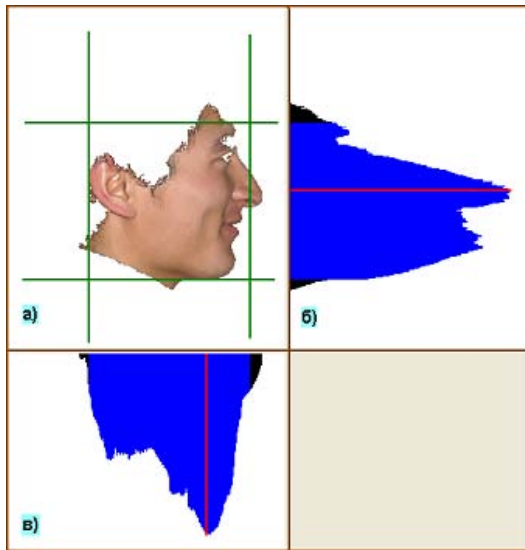


Рисунок 2: Процесс выделения области профильного изображения лица на основе анализа интегральных проекций.

3. ОПРЕДЕЛЕНИЕ ИДЕНТИФИКАЦИОННЫХ ТОЧЕК ПРОФИЛЬНОГО ИЗОБРАЖЕНИЯ ЛИЦА

В данной работе предусмотрено использовать, в основном четыре идентификационные точки для дальнейшего распознавания профильного изображения лица (Рис.3). Ими являются: козелок ушной раковины (точка A); кончик носа (точка B); переносица (точка C); подбородок (точка D). Из них A и B являются также опорными точками.

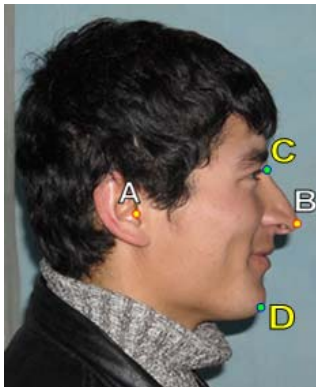


Рисунок 3: Идентификационные точки профильного изображения лица.

3.1. Определение координат кончика носа.

Вертикальную линию кончика носа можно определить также на основе анализа интегральной проекции и анализа области цвета кожи. Можно предположить, что вертикальная линия кончика носа расположены в самой крайней точке выделенной передней части лица, позиция которой (позиция горизонтальной линии) лежит в позиции максимального значения интегральной проекции.

3.2. Локализация ушной раковины и определение координат козелка.

Козелок ушной раковины является опорной точкой, поэтому необходимо найти его координаты.

В работе [4] рассмотрены задачи локализации фронтального изображения лица, где используется метод корреляционного анализа по маске для поиска области лица. Для локализации ушной раковины можно также использовать алгоритмы этого метода. Нужно только изменить маску поиска, т.е. вместо маски лица возьмем маску ушной раковины (Рис.4).



Рисунок 4: Маска поиска ушной раковины.

После локализации ушной раковины координаты козелка можно определить на основе маски, т.е. в найденной области выберем координаты, соответствующие координатам козелка по маске.

3.3. Определение координат переносицы и челюсти.

Ранее найденные опорные точки A и B помогут нам определить координаты переносицы (точка C) и челюсти (точка D). Таким образом, базируясь на углах между линиями, соединяющими идентификационные точки, можно уточнить соответствующие координаты точки C и D (Рис.5).

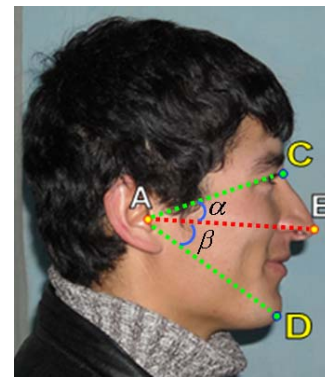


Рис.5. Определение координат точки C и D.

Эту операцию можно выполнять автоматически, используя ранее определенные параметры. В качестве параметров мы выберем координаты опорных точек (A и B) и контурной (граничной) линий передней части профильного изображения

лица. Координаты контурных линий (точек) лица были определены в процессе выделения цвета кожи.

Экспериментально установлено, что угол между линией AB и AC составляет примерно на 22° . А угол между линией AB и AD составляет примерно на 35° . То есть, $\alpha \approx 22$ и $\beta \approx 35$.

4. ОПРЕДЕЛЕНИЕ ИДЕНТИФИКАЦИОННЫХ ТОЧЕК И ФОРМИРОВАНИЕ ПРИЗНАКОВОГО ПРОСТРАНСТВА

В качестве признаков выберём евклидовы расстояния между точкой A и точкой граничных линий передней части лица (Рис.6). Эти расстояния обозначим через d_k . Точки граничных линий передней части лица определяются с помощью нумерованной шкалы, которую мы условно назовем «сеткой». Интервалы шкалы этой сетки будут одинаковыми (равномерными). Интервалы может установить каждый самостоятельно. На рис.7 показан график расстояний по узлам сетки, количество которых составляет 14.

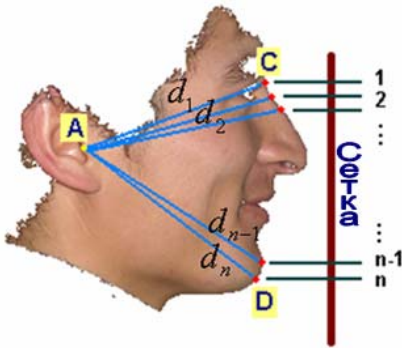
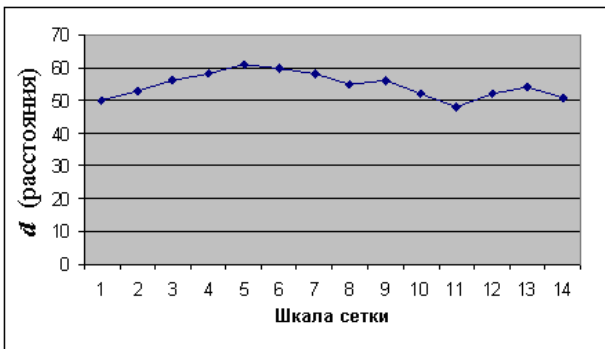


Рисунок 6: Общая схема определения идентификационных признаков профильного изображения лица.



Рисинук 7: График расстояний по номерам шкалы сетки.

Допустим, что количество узлов сетки равно n . Тогда, вектор признаков можем обозначить как $F_d = \{d_1, d_2, \dots, d_n\}$.

5. РАСПОЗНАВАНИЕ ПРОФИЛЬНОГО ИЗОБРАЖЕНИЯ ЛИЦА

Пусть, имеются два (A и B) изображения профильного лица, принадлежащие разным людям. Пространства признаков разных лиц, можно сопоставить различными способами. На основе наших исследований рекомендуется использовать статистические и геометрические методы.

В качестве статистического метода мы выбрали корреляцию, которая весьма успешно используется в разных системах распознавания. Ниже рассмотрим формулу вычисления коэффициентов корреляции для ранее сформированных признаков.

Допустим, дан набор чисел по признаку для двух (A и B) человек. $A_d = \{A_{d_i}\}_{i=1}^n$ и $B_d = \{B_{d_i}\}_{i=1}^n$. Для этих чисел определяем математическое ожидание.

$$MA = \frac{1}{n} \sum_{i=1}^n A_{d_i}, \quad MB = \frac{1}{n} \sum_{i=1}^n B_{d_i}$$

После этого образуем новый набор \bar{A} , \bar{B} , математические ожидания, которых равны нулю.

$$\bar{A}_i = (A_{d_i} - MA), \quad \bar{B}_i = (B_{d_i} - MB).$$

Дискретная формула корреляционной функции для набора A , B принимает следующий вид:

$$K = \frac{\sum_{i=1}^n \bar{A}_i \cdot \bar{B}_i}{\sqrt{\sum_{i=1}^n \bar{A}_i^2 \cdot \sum_{i=1}^n \bar{B}_i^2}}$$

Результат корреляции дает степень сравнения профилей лиц. Если величина K достаточно велика ($K \geq e$), то два изображения принадлежат одному человеку, в противном случае изображения принадлежат разным лицам.

Для сравнения профильных изображений лиц можно также использовать геометрический подход. В данном случае набор признаков немного изменяется, так как $F_d = \{d_T / d_1, d_T / d_2, \dots, d_T / d_n\}$. Здесь d_T - фиксированное расстояние. Формулы сравнения имеют следующий вид:

$$K = \frac{1}{n} \sum_{i=1}^n (A_i - B_i)^2 \quad \text{или} \quad K = \frac{1}{n} \sum_{i=1}^n |(A_i / B_i) - 1|.$$

В данном случае, если $K < e$, то изображения лица относятся к одному человеку.

6. ЗАКЛЮЧЕНИЕ

Алгоритм может дать ошибочные результаты в том случае, когда положение рта (закрытого или широко открытого) обоих проверяемых лиц будут различными. Усы и борода на лице также влияют на результаты алгоритма.

В общем, можно заключить, что предложенный алгоритм дает возможность создания компьютерной системы

биометрической идентификации личности человека на основе профильного изображения лица.

7. ЛИТЕРАТУРА

- [1] Кухарев Г.А. Биометрические системы: Методы и средства идентификации личности человека. -СПб.: Политехника, 2001. -240 с.
- [2] Тухтасинов М.Т. Алгоритмы определения лица и лицевых элементов на изображении // Узб. журнал «Проблемы информатики и энергетики», –Ташкент. -2004. -№4. с.-5-10.
- [3] Фазылов Ш.Х., Тухтасинов М.Т., Старовойтов В.В., Самаль Д.И., Ригол Г. Локализация фрагментов лица на цветных фотопортретах // Обработка информации и управление в чрезвычайных и экстремальных ситуациях: доклады 4-й Международной научной конференции. -Минск, 2004. -С.218-223.
- [4] Фазылов Ш.Х., Мирзаев Н.М., Тухтасинов М.Т. Об одном алгоритме определения местонахождения лица и координат зрачков на изображении // Доклады 13-й Всероссийской конференции "Математические методы распознавания образов" (ММО-13), Москва, 2007, -С. 409-411.
- [5] M. Pantic et al., Facial mimics recognition from face profil image sequences. Technical Report TR-DKS-02-01, DKS group, Delft University of Technology, Delft, 2002.
- [6] Maja Pantic et al., Facial gesture recognition in face profile image sequences. Delft University of Technology ITS / Mediamatics Dept. Delft, the Netherlands.

Автор

Тухтасинов Мумтозали Тулкиналиевич, к.т.н., доцент факультета Информатики Наманганского инженерно - педагогического института.

E-mail: mumtozali@yahoo.com

Fingerprint image quick processing

Vladimir Gudkov, Chelyabinsk State University, Chelyabinsk, Russia

diana@sonda.ru

Maxim Bokov, South Ural State University, Chelyabinsk, Russia

guardianm@mail.ru

Abstract

The article briefly describes an approach of features extraction from fingerprint image with time restriction. Minutiae are saved in an image template. Templates are used for fingerprint identification.

Keywords: *Fingerprint, minutiae, image flow, image processing.*

1. INTRODUCTION

More than a century has passed since investigations in the field of biometrics started. Fingerprints have been used for over a century and are the most widely used form of biometric identification. Fingerprint identification is commonly employed in forensic science to support criminal investigations, and in biometric systems such as civilian and commercial identification devices. Criminal system must be very reliable whereas civilian and commercial systems must be fast and reliable [7]. It influences deeply on choice of fingerprint recognition methods, for example in the systems like Control and Access systems, Time and Attendance systems.

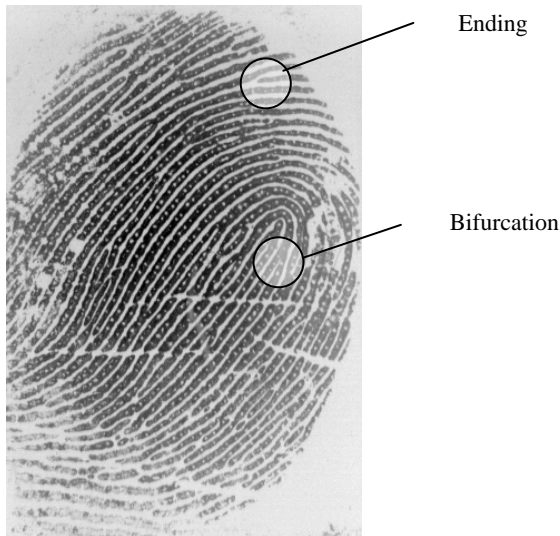


Figure 1: Fingerprint image.

The two most prominent ridge characteristics, called minutiae are: ridge endings and ridge bifurcations. Ridge endings are the points where the ridge curve terminates, and bifurcations are the ones where the ridge splits from a single path into two paths. Figure 1 illustrates an example of a ridge ending and a bifurcation. In the fingerprint image, ridges are dark whereas valleys are bright.

Quick processing is realized as a series of techniques, which include estimation of local ridge orientation, smoothing along ridges, estimation of local ridge period, segmentation, binarization and skeletonization. Although a minutiae-based approach is characterized by a high saliency, reliable automatic minutiae extraction can be problematic in extremely low-quality fingerprints devoid of any ridge structure. Time restriction limits a class of fingerprint images to medium and high quality fingerprint images.

The primary aim of this article is to implement a series of techniques for fingerprint image enhancement and minutiae extraction. Experiments with real fingerprint images were used to assess the performance of the implemented techniques.

Usually steps of preprocessing and enhancement are performed in order to simplify the fingerprint recognition. Therefore fingerprint image is presented in rectangular area G with a power $|G| = x_0 y_0$ as $F = \{f(x, y) \in 0..2^b - 1 | (x, y) \in X \times Y\}$, where b – represents the number of code bits allocated to each pixel (bit depths); $X = 0..x_0 - 1$ and $Y = 0..y_0 - 1$.

Consider two-dimensional images. An image represents an important class of data structures. Data objects may be taken as pixels, but it is more meaningful for image interpretation if we try, in some appropriate way, to take regions of the image as the data objects. Such regions may be approximate. One approach is to recursively divide the image into smaller regions. Such regions may be square or rectangular, to facilitate general implementation. Fingerprint image processing is represented structurally by pyramid \mathfrak{R} of correlated hierarchies [1, 2, 6]. Image segmentation is performed for any layer of arbitrary hierarchy. For instance l -th layer of k -th hierarchy $F_k^{(l)}$ is segmented $x_h y_h$ square blocks $S_{hk}^{(l)}(x, y)$ with side length 2^{h-k} and vertices $(x, y) \in X_h \times Y_h$, where $k < h$, h is hierarchy number and $X_h = 0..x_h - 1$, $Y_h = 0..y_h - 1$.

Access to any point of the block $S_{hk}(x, y)$ is written in coordinates $(u, v) \in \bar{X}_{hk} \times \bar{Y}_{hk}$ as:

$$\begin{cases} \bar{X}_{hk} = \{u + x2^{h-k} | x \in X_h \wedge u \in 0..2^{h-k} - 1\}, \\ \bar{Y}_{hk} = \{v + y2^{h-k} | y \in Y_h \wedge v \in 0..2^{h-k} - 1\}. \end{cases} \quad (1)$$

Access to the central point of the block $S_{hk}(x, y)$ is written in coordinates $(u, v) \in \hat{X}_{hk} \times \hat{Y}_{hk}$ as:

$$\begin{cases} \hat{X}_{hk} = \{2^{h-k-1} + x2^{h-k} | x \in X_h\}, \\ \hat{Y}_{hk} = \{2^{h-k-1} + y2^{h-k} | y \in Y_h\}. \end{cases} \quad (2)$$

Block size of h -th hierarchy is: $x_h = \left\lceil \frac{y_0}{2^h} \right\rceil$ and $y_h = \left\lceil \frac{y_0}{2^h} \right\rceil$,

where $\lceil a \rceil$ – ceiling a .

With hierarchical segmentation, blocks of the layer F_k map to blocks' vertices of the layer F_h of pyramid \mathfrak{R} , where $k < h$. Correspondingly blocks' vertices map to blocks situated closer to base of pyramid [2]. Block size affects execution time and processing quality deeply. Assume that $S_h(x, y) = S_{h0}(x, y)$ and vertices $S_h(x, y) \in F_h$.

Pyramid's layers can be represented as a set of real numbers and initial image can be represented as a set of nonnegative real numbers [1, 6]. It gets rid of tiresome integer representation of a signal and simplifies expression, though image discretization (pyramid's layers) is kept.

Apertures are used widely for compact mathematical formalization. Meanwhile, slotted rectangular aperture $A_h(x, y, \alpha, w)$, $A_h^-(x, y, \alpha, w)$ and circular aperture $A_h(x, y, w)$ play a crucial part. These apertures are represented by set of the layer's points of h -th hierarchy and associated angles as (u, v, β) . Aforesaid aperture can be defined as:

$$\begin{cases} A_h(x, y, \alpha, w) = \{(u, v, \beta) = (x +]w \cos(\alpha), y +]w \sin(\alpha), \beta) | w \in Z_w^+\}, \\ A_h^-(x, y, \alpha, w) = \{(u, v, \beta) = (x +]w \cos(\alpha), y +]w \sin(\alpha), \beta) | w \in Z_w^-\}, \end{cases} \quad (3)$$

$$A_h(x, y, w) = \bigcup_{\alpha \in Z^*} A_h(x, y, \alpha, w), \quad (4)$$

where $(x, y) \in X_h \times Y_h$ is the centre of aperture; $(u, v) \in X_h \times Y_h$ is the point of aperture; w is the size of aperture; $Z_w^+ = 1..w$; $Z_w^- = -w..-1 \cup 1..w$; α is the direction of aperture; $]a[$ - the nearest integer part of a . The direction from the centre of aperture (x, y) to the point (u, v) is defined as:

$$\beta = \arctg\left(\frac{v-y}{u-x}\right) + \pi m \mid m \in 0..1.$$

Fingerprint image preprocessing and enhancement must meet time restriction requirements and also algorithm must ensure acceptable quality of minutiae recognition, which can be verified with test base. List of minutiae is defined as:

$$L_m = \{M_i = \{x_i, y_i, \alpha_i, t_i\} \mid i \in 1..n\}, \quad (5)$$

where M_i – minutiae with i index number; (x_i, y_i) , α_i , t_i – coordinates, direction and type of minutiae; $n = |L_m|$ – cardinal number. Minutiae are detected in informative area of an image.

Then trade-off decision of speed – quality violation is implementation of six steps of fingerprint image processing, like this:

- orientation estimation;
- smoothing;
- period estimation;
- segmentation;
- binarization;
- skeletonization and minutiae recognition.

2. QUICK PROCESSING

The main goal of most fingerprint recognition algorithms [8] is minutiae, which can be used to calculate additional parameters of the fingerprint image. Quick processing is executed for initial fingerprint image $F_0^{(0)} = \{f_0^{(0)}(x, y)\}$ (see Figure 1).

2.1 Integral Image

An integral image (also known as a summed-area table) is a tool that can be used whenever we have a function from pixels to real numbers $f(x, y)$ (for instance, pixel intensity), and we wish to compute the sum of this function over a rectangular region of the image.

Without an integral image, the sum can be computed in linear time per rectangle by calculating the value of the function for each pixel individually. However, if we need to compute the sum over multiple overlapping rectangular windows, we can use an integral image and achieve a constant number of operations per rectangle with only a linear amount of preprocessing.

To compute the integral image, we store at each location, $I(x, y)$ the sum of all $f(x, y)$ terms to the left and above the pixel (x, y) . This is accomplished in linear time using the following equation for each pixel (taking into account the border cases),

$$I(x, y) = f(x, y) + I(x-1, y) + I(x, y-1) - I(x-1, y-1). \quad (6)$$

Figure 2 illustrates the computation of an integral image. Once we have the integral image, the sum of the function for any rectangle with upper left corner (x_1, y_1) , and lower right corner (x_2, y_2) can be calculated in constant time using the following equation,

$$\sum_{x=x_1}^{x_2} \sum_{y=y_1}^{y_2} f(x, y) = I(x_2, y_2) - I(x_1-1, y_2) - I(x_2, y_1-1) + I(x_1-1, y_1-1). \quad (7)$$

4	1	2	2
0	4	1	3
3	1	0	4
2	1	3	2

4	5	7	9
4	9	12	17
7	13	16	25
9	16	22	33

Figure 2: The integral image. Left: A simple input of image values. Right: The computed integral image.

2.2 Orientation estimation

The orientation estimation is a fundamental step in the processing and enhancement. It consists of two successive procedures. The simplest and most natural approach for extracting local ridge orientation is based on computation of gradients in the fingerprint image. The gradient $\nabla_x(x, y)$ at point (x, y) of $f_0^{(0)}(x, y)$, is a two-dimensional vector $[\nabla_x(x, y), \nabla_y(x, y)]$ where ∇_x and ∇_y components are the partial derivatives of $f_0^{(0)}(x, y)$ in (x, y) with respect to the x and y directions, respectively.

Evaluation of the orientation matrix. The essence of the method is to divide fingerprint image into nonoverlapping blocks $S_h(x, y)$ corresponding to Equation (1) and estimate the orientation $0 \leq \delta_h^{(0)}(x, y) < 180$ block-wise on hierarchy $h = 3(8 \times 8)$. An orientation image is then calculated, which is a matrix $\Lambda_h^{(0)}$ of direction vectors representing the ridge orientation at each

location in the image. The matrix $\Lambda_h^{(0)}$ can be estimated using following equation:

$$\Lambda_h^{(0)} = \left\{ \delta_h^{(0)}(x, y) \right\} = \left\{ \frac{\pi}{2} + \frac{1}{2} \arctg \left(\frac{2J_{12}(x, y)}{J_{22}(x, y) - J_{11}(x, y)} \right) \right\}, \quad (8)$$

where

$$\begin{aligned} J_{12}(x, y) &= \sum_{(u,v) \in S_h(x,y)} \nabla_x \nabla_y, \\ J_{11}(x, y) &= \sum_{(u,v) \in S_h(x,y)} \nabla_x \nabla_x, \\ J_{22}(x, y) &= \sum_{(u,v) \in S_h(x,y)} \nabla_y \nabla_y. \end{aligned}$$

Here ∇_x and ∇_y are the x - and y -gradient components in $(u, v) \in \bar{X}_{hk} \times \bar{Y}_{hk}$ corresponding to Equation (1) of the blocks $S_h(x, y)$ computed through 3×3 Sobel masks as:

$$\begin{aligned} \nabla_x &= \mathbf{H}_x ** f_0^{(0)}(u, v), \\ \nabla_y &= \mathbf{H}_y ** f_0^{(0)}(u, v), \end{aligned}$$

where \mathbf{H}_x and \mathbf{H}_y is direction error reduced convolution kernel [6]

$$\mathbf{H}_x = \begin{bmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{bmatrix}, \quad \mathbf{H}_y = \begin{bmatrix} -3 & -10 & -3 \\ 0 & 0 & 0 \\ 3 & 10 & 3 \end{bmatrix}.$$

The steps for calculating the orientation matrix $\Lambda_h^{(l)}$ are as follows:

1. Firstly, integral images $IG_{xy}(x, y), IG_{xx}(x, y)$ and $IG_{yy}(x, y)$ must be computed according to Equation (6) in order to compute $J_{12}(x, y), J_{11}(x, y), J_{22}(x, y)$ for different-sized rectangular apertures $A_h(x, y, w)$ in constant time.

For instance:

$$IG_{xy}(x, y) = G_{xy}(x, y) + IG_{xy}(x-1, y) + IG_{xy}(x, y-1) - IG_{xy}(x-1, y-1),$$

where $G_{xy}(x, y) = \nabla_x(x, y) \nabla_y(x, y)$. The integral images $IG_{xx}(x, y)$ и $IG_{yy}(x, y)$ can be calculated similarly.

2. After computing $IG_{xy}(x, y), IG_{xx}(x, y)$ and $IG_{yy}(x, y)$ we can calculate $J_{12}(x, y), J_{11}(x, y), J_{22}(x, y)$ by using following equations:

$$J_{12}(x, y) = IG_{xy}(x_2, y_2) - IG_{xy}(x_1 - 1, y_2) - IG_{xy}(x_2, y_1 - 1) + IG_{xy}(x_1 - 1, y_1 - 1),$$

$$J_{11}(x, y) = IG_{xx}(x_2, y_2) - IG_{xx}(x_1 - 1, y_2) - IG_{xx}(x_2, y_1 - 1) + IG_{xx}(x_1 - 1, y_1 - 1),$$

$$J_{22}(x, y) = IG_{yy}(x_2, y_2) - IG_{yy}(x_1 - 1, y_2) - IG_{yy}(x_2, y_1 - 1) + IG_{yy}(x_1 - 1, y_1 - 1),$$

where the point (x_1, y_1) is the upper left corner of the aperture $A_h(x, y, w)$ and the point (x_2, y_2) is the lower right corner of the aperture $A_h(x, y, w)$.

Actually, elements in $\Lambda_h^{(l)}$ are computed by averaging in blocks $\{S_h(x, y)\}$ by means of structural point-wise tensor operator J as:

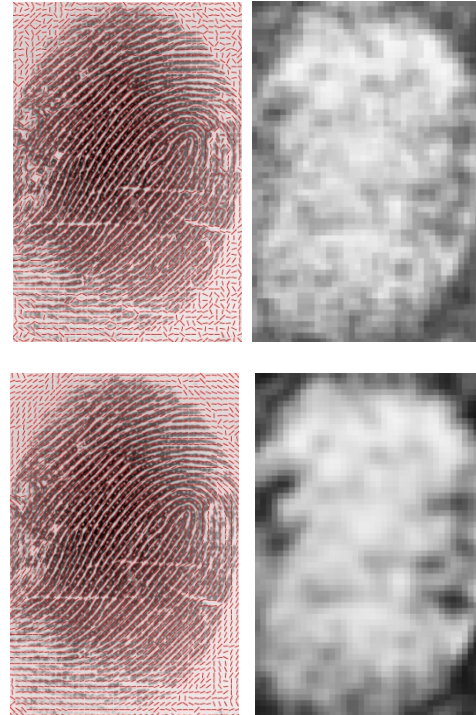
$$J = \begin{bmatrix} J_{11} + J_{22} \\ J_{22} - J_{11} \\ 2J_{12} \end{bmatrix}. \quad (9)$$

Analyze and correction the orientation matrix. The reliability of the estimate $\delta_h^{(0)}(x, y)$ can be derived by the coherence of the orientation vectors in the local apertures $A_h(x, y, w)$. For the gradient-based approach corresponding to Equation (8) we can calculate coherence matrix, which defined as:

$$M_h^{(0)} = \left[\mu_h^{(0)}(x, y) \right] = \left[\frac{\sqrt{(J_{22}(x, y) - J_{11}(x, y))^2 + 4J_{12}^2(x, y)}}{J_{11}(x, y) + J_{22}(x, y)} \right]. \quad (10)$$

An example of local coherence map computed with Equation (10) is shown in Figure 3. The coherence of ideal lines is equal to one, but the coherence of isotropic structures is equal to zero [6].

Due to the integral images $IG_{xy}(x, y), IG_{xx}(x, y)$ and $IG_{yy}(x, y)$ we can compute orientation matrices $\Lambda_h^{(0)}, \Lambda_h^{(1)}, \Lambda_h^{(2)}, \Lambda_h^{(3)}$ and coherence matrices $M_h^{(0)}, M_h^{(1)}, M_h^{(2)}, M_h^{(3)}$ with different apertures $A_h(x, y, w)$ like these $A_h(x, y, 8), A_h(x, y, 24), A_h(x, y, 40), A_h(x, y, 56)$ respectively ($h=3$). Figure 3 illustrates the results of these calculations.



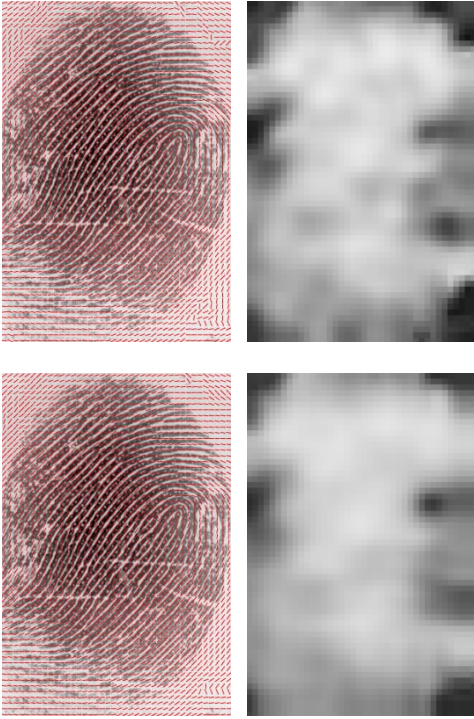


Figure 3: Top-down: matrices $\Lambda_h^{(0)}$, $\Lambda_h^{(1)}$, $\Lambda_h^{(2)}$, $\Lambda_h^{(3)}$ and their coherence matrices $M_h^{(0)}$, $M_h^{(1)}$, $M_h^{(2)}$, $M_h^{(3)}$ respectively.

After calculation of the orientation matrices $\Lambda_h^{(0)}$, $\Lambda_h^{(1)}$, $\Lambda_h^{(2)}$, $\Lambda_h^{(3)}$ and their coherence matrices $M_h^{(0)}$, $M_h^{(1)}$, $M_h^{(2)}$, $M_h^{(3)}$ we can analyze them. Let's compute a derivatives matrix $N_h^{(0)}$ and an orientation matrix $O_h^{(0)}$, which contain the orientation with maximum coherence from different orientation matrices. Matrices $N_h^{(0)}$ and $O_h^{(0)}$ can be defined as:

$$N_h^{(0)} = [v_h^{(0)}(x, y)] = \left[\frac{1}{n} \sum_{l=1}^{n-1} (\mu_h^{(l)}(x, y) - \mu_h^{(l-1)}(x, y)) \right],$$

$$O_h^{(0)} = [o_h^{(0)}(x, y)] = [\lambda_h^{(c)}(x, y)],$$

where $c = \arg \max_l (\mu_h^{(l)}(x, y))$; n is a quantity of the layers, in this implementation $n=4$.

Thus, $N_h^{(0)}$ helps us assess how the coherence changes. It allows finding blocks with singularities like delta, loop and whorl.

Then, the orientation matrix $\Lambda_h^{(4)}$ containing reliable and accurate orientations can be computed by analyzing matrices $N_h^{(0)}$ and $O_h^{(0)}$. Matrix $\Lambda_h^{(4)}$ is defined as:

$$\Lambda_h^{(4)} = [v_h^{(4)}(x, y)] = \begin{cases} \Lambda_h^{(3)}, & \text{if } N_h^{(0)} > T, \\ O_h^{(0)}, & \text{else,} \end{cases}$$

where T is a threshold, which can be calculated experimentally (in this implementation $T = -0.1$).

Elements of $\Lambda_h^{(4)}$ can be corrected by using following equation:

$$\Lambda_h^{(5)} = \left\{ \delta_h^{(5)}(x, y) \right\} = \left\{ \frac{1}{2} \arctg \left(\frac{im_h^{(0)}(x, y)}{re_h^{(0)}(x, y)} \right) \right\}, \quad (11)$$

where

$$re_h^{(0)}(x, y) = \sum_{k=-1}^1 \sum_{m=-1}^1 \mu_h^{(0)}(x+k, y+m) \cos(2\delta_h^{(4)}(x+k, y+m)),$$

$$im_h^{(0)}(x, y) = \sum_{k=-1}^1 \sum_{m=-1}^1 \mu_h^{(0)}(x+k, y+m) \sin(2\delta_h^{(4)}(x+k, y+m)).$$

Thereby, the orientation matrix $\Lambda_h^{(4)}$ is more accurate and more reliable than the matrix $\Lambda_h^{(3)}$. Figure 4 illustrates the matrix $\Lambda_h^{(4)}$ and its correction, the matrix $\Lambda_h^{(5)}$.

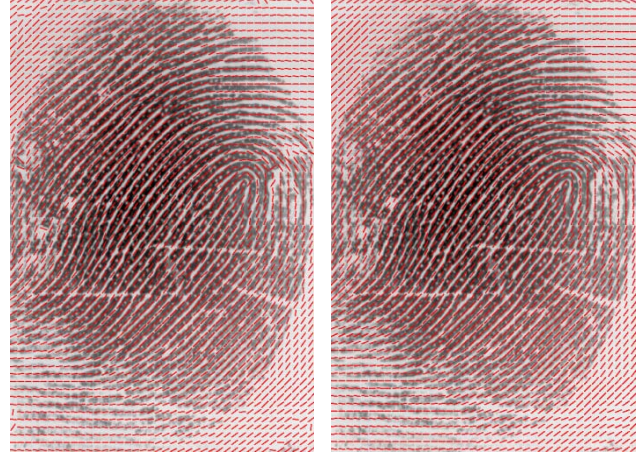


Figure 4: Fingerprint's orientation. Left: the orientation matrix $\Lambda_h^{(4)}$. Right: the orientation matrix $\Lambda_h^{(5)}$.

2.3 Smoothing

Smoothing eliminates ruptures and smudges. Before smoothing irregularity matrix $IR_h^{(0)}$ is computed by using following equation:

$$IR_h^{(0)} = [ir_h^{(0)}(x, y)] = \left[1 - \frac{\left\| \sum_{m=-1}^1 \sum_{n=-1}^1 d(x+n, y+m) \right\|}{\sum_{m=-1}^1 \sum_{n=-1}^1 \|d(x+n, y+m)\|} \right],$$

where $d(x, y) = [\cos 2\delta_h^{(5)}(x, y), \sin 2\delta_h^{(5)}(x, y)]$.

After that apply smoothing filter for each block $S_h(x, y)$ of the image $F_0^{(0)}$. Values from the base of the block $S_h(x, y)$ are smoothed by using following equation:

$$F_0^{(1)} = \begin{cases} f_0^{(1)}(x, y) = \mathbf{H}_1 * \Xi_0^{(\alpha)}(x, y), & \text{if } ir_h^{(0)}(x, y) < t, \\ f_0^{(1)}(x, y) = f_0^{(0)}(x, y), & \text{else,} \end{cases} \quad (12)$$

where \mathbf{H}_1 is a kernel of the one dimensional convolution; t is a threshold (in this implementation $t=0.5$); a set $\Xi_0^{(\alpha)}(x, y) = \{\xi_0^{(\alpha)}(u, v)\}$ consists of the elements, which are chosen from $F_0^{(0)}$ with aperture according to Equation (3), as $\{\xi_0^{(\alpha)}(u, v)\} = \{f_0^{(0)}(u, v) | (u, v) \in A_0^-(x, y, \alpha, w) \cup (x, y)\}$;

$\alpha = \delta_h^{(5)}(x, y) \in \Lambda_h^{(5)}$ – direction of the aperture, that is identical for all elements of the block $S_h(x, y)$; w – size of the aperture.

Renumber ordered counts of the set $\Xi_0^{(\alpha)}(x, y) = \{\xi_0^{(\alpha)}(u, v)\}$, which is generated by slotted rectangular aperture according to Equation (3), as $k \mapsto (u_k, v_k)$, where $k \in 0..N$; $N = 2w + 1$. Then the convolution kernel \mathbf{H}_1 is calculated as:

$$\mathbf{H}_1 = \exp\left(-\frac{(w-k)^2}{2\sigma^2}\right),$$

where σ is the standard deviation, that defines Gaussian steepness [8] (2-4 in this implementation); $k \equiv w$ is center of the aperture.

Smoothing filter is inherently first factor of an even-symmetric Gabor Filter [8]. Smoothing result is shown in Figure 5.



Figure 5: Smoothed image.

2.4 Period estimation

In addition to the orientation matrix, another important parameter that is used in the fingerprint recognition is the local ridge period. This step of fingerprint image processing is also consists of two successive procedures and performed in the same hierarchy $h = 3$.

Evaluation of the period matrix. Method is based on autocorrelation function.

Definition 1. The local ridge period is a value $t = w/n$ inversely to the number of ridges per unit length along a hypothetical segment centered at (x, y) and orthogonal to the local ridge orientation [8].

Specify a slit $C(x, y) = A_0^-(x, y, \alpha, w) \cup (x, y)$, which is generated by slotted rectangular aperture according to Equation (3), and renumber counts $(u, v) \in C(x, y)$ as $k \mapsto (u_k, v_k)$, where $k \in 0..N$; $N = 2w + 1$. In the slit $C(x, y)$ with center $k \equiv w$ k -ordered values of the image $f_0^{(1)}(k)$ is collected ($w = 16$ in this implementation). Orientation of the slit $C(x, y)$, that is represented angle α , can be defined as:

$$\alpha = \frac{\pi}{2} + \delta_h^{(5)}(x, y).$$

The autocorrelation function can be calculated by using following equations [5]:

$$r(i) = \frac{1}{N} \sum_{k=0}^{N-i-1} \widehat{f}_0^{(1)}(k) \widehat{f}_0^{(1)}(k+i), \quad (13)$$

$$\widehat{f}_0^{(1)}(k) = f_0^{(1)}(k) - \bar{f},$$

$$\bar{f} = \frac{1}{N} \sum_{k=0}^{N-1} f_0^{(1)}(k).$$

Set difference as: $\Delta r(i) = r(i+1) - r(i)$; N is quantity counts for calculating of autocorrelation function. Then elements of the period matrix $\mathbf{T}_h^{(0)} = \{t_h^{(0)}(x, y)\}$ can be estimated as:

$$t_h^{(0)}(x, y) = \operatorname{argmin}_j \{|\Delta r(0), \dots, \Delta r(j)| \mid \Delta r(j-1) > 0 \wedge \Delta r(j) \leq 0\}. \quad (14)$$

Actually, for each block $S_h(x, y)$, where $h = 3$, we find the center of the block and get the slit perpendicular to flow. Thereafter the slit is analyzed by the autocorrelation function according to Equation (13) and local period is calculated from Equation (14). Figure 6 illustrates results of the autocorrelation function for the white slit.

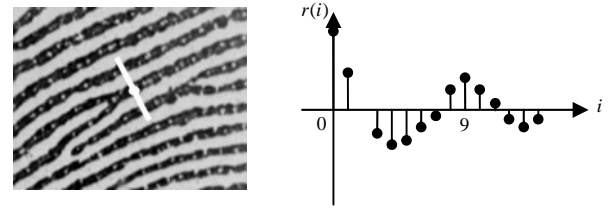


Figure 6: Calculating a local period.

Analyze and correction of the period matrix. On the principle that ratio of the constant component autocorrelation function and its value $r(t_h^{(0)}(x, y))$ equal 1 for sine or cosine, we can calculate reliability matrix $Z_h^{(0)} = \{\zeta_h^{(0)}(x, y) = r(t_h^{(0)}(x, y)) / r(0)\}$, $h = 3$.

It is well known that the local period can be in the range $4 \leq t_h^{(0)}(x, y) \leq 17$ for fingerprint image with resolution 500 dpi [8]. It allows deleting some mistakes of the local period calculations, by setting $t_h^{(0)}(x, y) = 0$.

The corrected period matrix can be calculated by using following equation:

$$t_{h,j}^{(1)}(x, y) = \frac{\sum_R t_{h,j-1}^{(1)}(u, v) \zeta_h^{(0)}(u, v)}{\sum_R \zeta_h^{(0)}(u, v)}, \quad (15)$$

where condition of summing is $R = \{t_{h,j-1}^{(1)}(u, v) > 0 \text{ and } n = \sum_R 1 \text{ is quantity of nonzero elements (in this implementation } n > 4)\}$.

Thereby, calculation errors are deleted, ridge periods are averaged and are predicted. The result of the calculated period matrix and its reliability matrix is shown in Figure 7. Period with value equal zero is black.

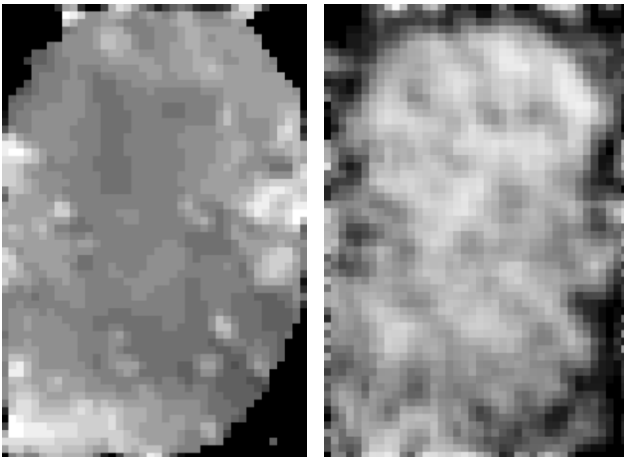


Figure 7: Left: the period matrix. Right: The reliability matrix.

2.5 Segmentation

Segmentation is the process of separating the foreground regions in the image from the background regions. The foreground regions correspond to the clear fingerprint area containing the ridges and valleys, which is the area of interest. The background corresponds to the regions outside the borders of the fingerprint area, which do not contain any valid fingerprint information. When minutiae extraction algorithms are applied to the background regions of an image, it results noisy and false minutiae. Thus, segmentation is employed to discard these background regions, which facilitates the reliable extraction of minutiae.

Segmentation is accomplished in the same hierarchy $h=3$ and consists in calculating a matrix $C_h^{(0)} = \{c_h^{(0)}(x, y)\}$ as:

$$c_h^{(0)}(x, y) = \begin{cases} 1, & \text{if } k_1 \zeta_h^{(1)}(x, y) + k_2 \mu_h^{(1)}(x, y) > \kappa_0, \\ 0, & \text{else,} \end{cases} \quad (16)$$

where $\zeta_h^{(1)}(x, y)$ – period reliability value, smoothed with 3×3 mask; $\mu_h^{(1)}(x, y)$ – coherence according to Equation (10), smoothed with 3×3 mask; k_0, k_1, k_2 – learning coefficients.

Essentially, segmentation relies on two characters: coherence and period reliability. These characters are complex and their combination allows enhancing segmentation accuracy.

After segmentation islands of “diverse” regions can appear. They can be processed with mathematical morphology operations [1, 6], but under time restriction it is not advisable.

2.6 Binarization

Most minutiae extraction algorithms operate on binary images where there are only two levels of interest: the black pixels that represent ridges, and the white pixels that represent valleys. Binarization is the process that converts a grey level image into a binary image. This improves the contrast between the ridges and valleys in a fingerprint image, and consequently facilitates the extraction of minutiae.

Binarization is based on two earlier calculated matrices $T_h^{(1)}$ and $C_h^{(0)}$. The binarization process involves examining the grey-level value of each pixel of the block $S_h(x, y)$ of the image $F_0^{(1)}$ with

mark $c_h^{(0)}(x, y) \in \{1\}$, and, if the value is greater than the local threshold, then the pixel value is set to a binary value one; otherwise, it is set to zero.

$$f_0^{(2)}(x, y) = \begin{cases} 1, & \text{if } f_0^{(1)}(x, y) < km, \\ 0, & \text{else.} \end{cases}$$

Threshold km is calculated from an aperture $A_h(x, y, w)$, where m is a mean value of $A_h(x, y, w)$; k – some coefficient (in implementation $k=0.98$); w is size of the aperture $A_h(x, y, w)$, that can be get from the matrix $T_h^{(1)}$.

The mean value is calculated with the integral image, which allows accelerating binarization, according to Equation (6) and Equation (7). Figure 8 illustrates the result of binarization.



Figure 8: Binary image.

2.7 Skeletonization and minutiae recognition

The final fingerprint image quick processing step performed prior minutiae recognition is skeletonization (thinning). Skeletonization is the process of getting a skeleton by thinning of binary image (see Figure 8). In order to get accurate skeleton we can use mathematical morphology [1] to fill holes, remove small break and other artifacts. Work in some definitions.

Definition 2. Skeleton is a simple chain $\langle u, v \rangle$ with nodes u and v in eight-neighborhood and each node $p_1 \in \langle u, v \rangle$ has two adjacent to it nodes p_2 and p_3 , though nodes p_2 and p_3 are non-adjacent.

Definition 3. Ending is a node p_1 , which has only one adjacent node p_2 .

Definition 4. Bifurcation is a node p_1 , which has only three adjacent nodes p_2, p_3 and p_4 , though any two nodes of a set $\{p_2, p_3, p_4\}$ are pair-wise non-adjacent.

Skeletonization is based on painting the points of $f_0^{(2)}(x, y) \in \{0\}$ according to the rules $P(\xi(x, y))$ defining in special table form. The rules $P(\xi(x, y))$ are based on a neighborhood identifier as:

$$\xi(x, y) = \sum_{i \in I} f(i) \cdot 2^i,$$

where $f(i)$ is 1 for line (skeleton), otherwise 0; $i \in I = 0..7$ is a number of a aperture sector according to Equation (4). Value $\xi(x, y) \in 0..255$ determines table cell. According to [3], iterative application the rules $P(\xi(x, y))$ allows us to compute the skeleton. Endings and bifurcations are got from the skeleton's vertices [4], which situated in fingerprint image informative area, and placed to a list according to Equation (5). Skeleton and minutiae with their directions are shown in Figure 9. Endings are drawn with black color and bifurcations are drawn with grey color.

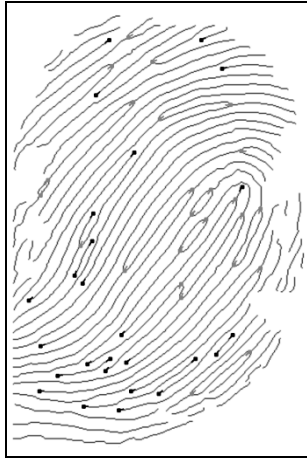


Figure 9: Skeleton and minutiae.

3. CONCLUSION

The primary focus of the work is on the quick processing of fingerprint images. In the article, it is proposed the group of techniques, which provide acceptable quality recognition with time restriction. Techniques include: estimation and correction the orientation matrix and its coherence, smoothing along ridges, estimation and correction period matrix and its reliability, segmentation of fingerprint image, binarization, skeletonization and minutiae recognition. Calculation of the orientation matrix is based on tensor analysis. Calculation of the period matrix is based on autocorrelation function. Total calculation time is about 100 milliseconds (Intel Pentium 4 CPU 3.0 GHz).

The algorithm also identifies the unrecoverable corrupted regions in the fingerprint image and removes them from further processing. This is a very important property because such unrecoverable regions appear in some of the corrupted fingerprint images and they are extremely harmful to minutiae extraction.

Also processing is based on convolutions, that allows integrating the processing into boards TMS and processors DSP [5] and using embedded convolution. The last makes possible realization of simple portable biometric systems, which work online.

Similarly to smoothing, one-dimensional Gabor filter can be used to filter image perpendicular to the orientation matrix. It recovers image and reduces time processing. Computed binary image can be smoothed again for more reliable and more accurate results.

4. REFERENCES

- [1] R.C. Gonzales, R.E. Woods. Digital image processing, second edition. Prentice Hall, 2002.
- [2] V.U. Gudkov. The methods of the first processing of dactiloscopic images, monography. Geotur, Miass, 2008.
- [3] V.U. Gudkov, A.A. Kolyada, A.V. Chernyavskiy. New technology of dactiloscopic images skeleton formation. Methods, algorithms and software of flexible information technology for automated identification systems, Minsk, BSU, 1999, pp. 71-82.
- [4] F.A Novikov. Discrete mathematics for programmers. Piter, 2008.
- [5] A.B. Sergienko. Digital signal processing. Piter, 2002.
- [6] B. Jahne. Digital image processing. Technosphere, 2007.
- [7] R.M. Bolle, J.Y. Connel, S. Pankanti, N.K. Ratha. Guide to biometrics. Springer Verlag, 2004.
- [8] D. Maio, D. Maltoni, A. K. Jain, and S. Prabhakar. Handbook of Fingerprint Recognition. Springer Verlag, 2003.

About the author

Vladimir Gudkov is a doctor at Chelyabisk State University, Department of Applied Mathematics. His contact e-mail is <mailto:diana@sonda.ru>.

Maxim Bokov is a Ph.D. applicant at South Ural State University, Department of Applied Mathematics. His contact email is <mailto:guardianm@mail.ru>.

Система идентификации дикторов по голосу для конкурса NIST SRE 2010

д.т.н. Матвеев Ю.Н., Симончик К.К.
 ООО «Центр Речевых Технологий», Санкт-Петербург, Россия
matveev@speechpro.com, simonchik@speechpro.com

The speaker identification system for the NIST SRE 2010
 Matveev Yu.N., Simonchik K.K.

ABSTRACT (ENG)

The speaker identification system, submitted for the NIST Year 2010 Speaker Recognition Evaluation (SRE), is described.

Keywords: *Biometry, speaker identification, voice recognition, speech features, pitch, formants, GMM, SVM, NIST.*

Аннотация

Приведено описание системы идентификации дикторов по голосу, разработанной для конкурса по оцениванию систем распознавания дикторов NIST SRE 2010.

Ключевые слова: *Биометрия, идентификация диктора, распознавание по голосу, основной тон, форманты, СГР, SVM, NIST.*

1. ВВЕДЕНИЕ

Системы идентификации (распознавания, верификации) дикторов по голосу относятся к классу биометрических систем, достоинством которых является то, что они чаще всего не требуют дополнительного оборудования для регистрации голоса и могут быть реализованы с использованием телефонных сетей или устройств ввода-вывода разных типов (микрофонов).

Область использования такого рода приложений обширна:

- автоматическая идентификация подозреваемого в телефонном канале;
- автоматическая верификация клиентов при удаленном доступе по телефонному каналу;
- обработка речевых баз данных;
- криминалистические исследования.

В данной работе представляется описание текстонезависимой системы автоматической идентификации дикторов по голосу, разработанной ООО «Центр Речевых Технологий» для участия в международном конкурсе по оцениванию систем распознавания дикторов NIST SRE 2010.

В профессиональной среде NIST SRE (Speaker Recognition Evaluation) называют неофициальным чемпионатом мира по голосовой идентификации. Начиная с 1996 года, этот конкурс ежегодно проводится американским Национальным Институтом Стандартов и Технологий (англ. National Institute of Standards and Technology, NIST). Его основная цель – оценить уровень существующих технологий и определить перспективные направления развития индустрии. Регулярно в

конкурсе принимают участие ведущие компании, университеты и лаборатории со всего мира. В 2010 году в конкурсе участвовало 46 научных команд.

Первой особенностью оценивания NIST SRE 2010 являлось использование баз речевых данных, собранных по различным каналам связи и в акустике помещений, и характеризующимися широким диапазоном отношений сигнал-шум и уровней реверберации (см. рисунок 1). Точка на скаттерграмме обозначает присутствие в речевом корпусе фонограммы с определенным уровнем шума (SNR, dB) и уровнем реверберации (Reverberation time, sec).

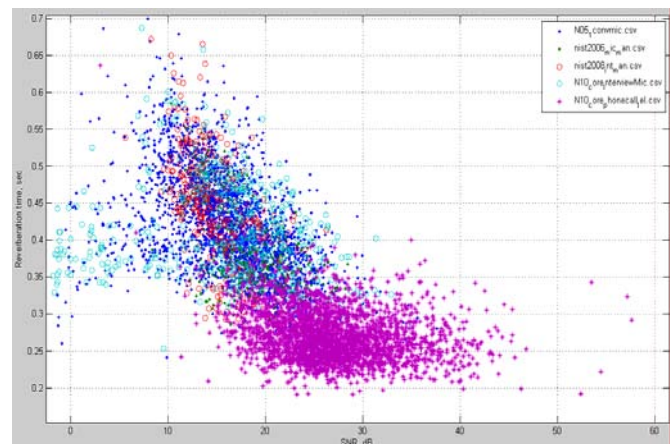


Рисунок 1: Скаттерграмма корпусов речевых данных NIST различных годов: фиолетовый цвет – сотовый корпус 2010 г.; остальные – корпуса речевых данных в акустике помещений 2005-2010 г.г.

Второй особенностью NIST SRE 2010, в сравнении с предыдущими конкурсами, являлось то, что в этом году организаторы задали новую функцию минимизации ошибки идентификации, суть которой состояла в крайне высокой стоимости ошибки ложного пропуска нецелевого диктора:

$$DCF = FR + 999 FA,$$

где FR - false rejection error rate (вероятность ошибки ложного отклонения);

FA - false acceptance error rate (вероятность ошибки ложного пропуска);

Введение новых значений весов параметров требует значительных объемов данных для калибровки порога принятия решения системы идентификации, в связи с тем,

что количество попыток идентификации нецелевого диктора должно быть достаточно большим для статистически устойчивой оценки *DCF*.

Точность калибровки особенно важна при применении голосовой идентификации в реальных условиях и зачастую играет критическую роль, так как позволяет максимально точно адаптироваться под прикладные задачи.

В данной работе описывается система, которая показала один из лучших результатов по качеству идентификации, в том числе, заняла первое место по уровню калибровки среди коммерческих систем.

2. МЕТОДЫ ИДЕНТИФИКАЦИИ ДИКТОРА

Принцип работы системы идентификации диктора основан на выделении из фонограмм речи и последующем попарном сравнении биометрических признаков (содержащихся в голосе индивидуальных, идентификационно значимых, признаков личности).

Выделение и сравнение биометрических признаков производится с использованием различных языко- и текстонезависимых методов идентификации дикторов по голосу. Система распознавания диктора называется текстонезависимой, если она не содержит информации о том, что именно диктор будет произносить (система обучается и тестируется на произвольных речевых данных).

На данный момент наиболее распространённым подходом к решению задач текстонезависимой идентификации является подход на основе использования моделей гауссовых смесей (англ. Gaussian Mixture Models, GMM) [1]. В качестве речевых признаков в подавляющем числе систем идентификации используются мэл-частотные кепстральные коэффициенты (англ. Mel-Frequency Cepstral Coefficients, MFCC) [2]. Модель голоса диктора представляет собой аппроксимацию распределения используемых речевых признаков смесью гауссовых распределений (GMM-модель). Значения равновероятной ошибки (англ. Equal Error Rate, EER) принятия чужого и отбрасывания своего диктора для метода на основе MFCC-GMM зависят от длительности сравниваемых речевых фрагментов и могут достигать величины ~4–5%.

Однако, при относительно высокой точности идентификации, по сравнению с другими популярными методами, такими как спектрально-формантный (СФ) метод [3] и метод идентификации на основе статистик основного тона (СОТ) [4], метод MFCC-GMM предъявляет высокие требования к качеству сигнала, обладает сильной зависимостью от вида обучающего материала, а также требует относительно больших временных затрат на выделение биометрических признаков. Сравнительные характеристики перечисленных методов приведены в таблице 1 (количество знаков «+» отражает степень зависимости метода от параметров сигнала).

Таблица 1

Метод	Параметры сигнала		
	Продолжительность	Качество сигнала	Физическое и эмоциональное состояние
СФ	+++	++	+
СОТ	++	+	++++
MFCC-GMM	+++	++++	++

Для улучшения показателей системы идентификации на основе GMM-моделей голосов дикторов обычно используются:

- канало-компенсация [5],
- классификатор на основе машины опорных векторов (англ. support vector machine, SVM),

Следует отметить, что совместное использование GMM-SVM моделей является на сегодняшний день доминантным в задачах верификации/идентификации/распознавания дикторов по голосу [1].

3. ОПИСАНИЕ СИСТЕМЫ

3.1 Структура системы

В системе автоматической идентификации личности по голосовым признакам в естественной речи осуществляется сравнение одной или нескольких записей (моделей) голоса неизвестного диктора с одной или несколькими записями (моделями) голоса известного диктора. В результате такого сравнения определяется, насколько похож голос неизвестного диктора на голос известного и, следовательно, принадлежат ли записи речи одному человеку или разным людям.

Если тестируемая фонограмма речи диктора может не принадлежать ни одному из кандидатов, то в систему дополнительно вводится фоновая модель (модель «самозванца»), и схема такой системы, называемой системой идентификации дикторов на открытом множестве, имеет вид, представленный на рисунке 2.

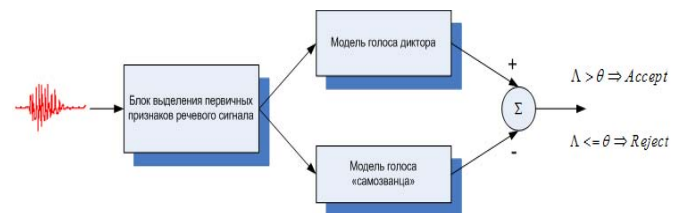


Рисунок 2: Структурная схема системы идентификации дикторов по голосу на открытом множестве.

Для каждого диктора-кандидата проводится сравнение речевого сигнала с моделью голоса данного диктора. Однако, кроме этого, вычисляется также вероятность того, что речевой сигнал принадлежит какому-нибудь другому диктору – на основании так называемой универсальной фоновой модели (англ. Universal Background Model, UBM) или модели «самозванца» (англ. impostor) на рисунке 2, которая описывает некоторые усредненные характеристики всех дикторов по используемой речевой базе.

3.2 Описание системы

Представленная на конкурсе NIST SRE 2010 система состояла из 6 различных гендеро- и канало-зависимых подсистем. Подсистемы адаптировались для различных каналов получения фонограмм:

- телефонного;
- микрофонного;

- смешанного (телефон-микрофон).

Кроме того, в рамках одного канала производилось дополнительное деление подсистем на две гендеро-зависимые (для женских и мужских голосов) подсистемы.

В качестве обучающих данных были взяты речевые базы NIST SRE прошлых лет (2004, 2006 и 2008 гг.) общим объемом более 20 тыс. фонограмм.

Для повышения надежности системы в качестве дополнительных речевых признаков были использованы линейно-частотные кепстральные коэффициенты (англ. Linear-Frequency Cepstral Coefficients, LFCC) [2], что обеспечило повышение качества идентификации в микрофонном канале.

Результирующее решение (“decision”) по обоим признакам (MFCC и LFCC) получалось путем вычисления «обобщенного решения» (“fusion”), получаемого методом взвешенного голосования, когда результату работы каждой подсистемы присваивается некоторый вес (см. рисунок 3). Для определения этих весов на этапе обучения системы использовался инструментарий собственной разработки. Обучение и точная калибровка системы производилась на речевой базе NIST SRE 2005.

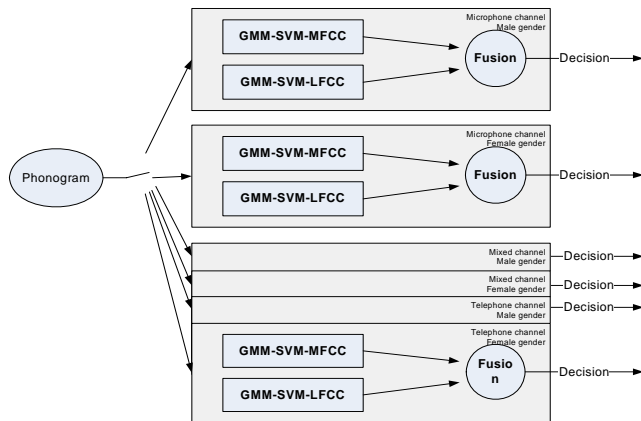


Рисунок 3: Структура системы идентификации диктора ООО «Центр Речевых Технологий».

GMM-модели голосов дикторов обучались методом максимального правдоподобия Фогта [5] с использованием компенсации канальных искажений. Размерность пространства собственных каналов для разных подсистем варьировалось от 50 до 80. В качестве классификатора использовался SVM с классической zt-нормализацией [5], которая представляла собой нормирование выходной дистанции SVM по случайным произнесениям дикторов из речевой базы объемом 1000-2000 фонограмм.

4. СМЕШИВАНИЕ ПОДСИСТЕМ И СИСТЕМ

4.1 Основные характеристики систем и подсистем

На конкурс NIST SRE 2010 были предоставлены три системы (SVID-1, SVID-2, SVID-3), которые отличаются корпусами речевых данных, используемых для обучения и настройки систем.

4.1.1 Primary system (SVID-1)

Базовая (primary) система является комбинацией двух подсистем, каждая из которых строилась на отдельных наборах речевых признаков:

- Первая подсистема: на базе 39-мерных векторов признаков, составленных из 13 MFCC-коэффициентов, дополненных их первыми и вторыми производными. Для каждого из векторов применялась процедура вычитания кепстрального среднего (CMS).
- Вторая подсистема: на базе 39-мерных векторов признаков, составленных из 13 LFCC-коэффициентов, дополненных их первыми и вторыми производными. Для каждого из векторов применялась процедура вычитания кепстрального среднего (CMS).

Каждая из подсистем, в свою очередь, имеет 6 гендеро- и канало-зависимых UBM. При обучении UBM использовались 1024-компонентные гауссовы смеси. База обучения UBM состояла из речевых корпусов Switchboard II Phases 2&3, Switchboard Cellular Parts 1&2, NIST SRE 2004, 2006&2008, из которых отбирались фонограммы дикторов, имеющих по 5–10 сессий записи их речи. Характеристики базы обучения представлены в таблице 2.

Таблица 2

Пол	Общее кол-во	Каналы		
		тел-тел	мик-мик	тел-мик
муж	дикторов	788	158	280 (153 тел+158 мик)
	фонограмм	6546	1516	2290
жен	дикторов	1042	203	371 (201 тел+203 мик)
	фонограмм	8589	1955	3050

Импостеры отбирались из РБД NIST SRE 2006, 2008. Характеристики базы импостеров представлены в таблице 3.

Таблица 3

Пол	Общее кол-во	Каналы		
		тел	мик	тел-мик
муж	дикторов	1070	1450	1000
жен	дикторов	1227	2236	1000

Для сегментации дикторов использовалась информация из ASR (Automatic Speech Recognition) транскрипции, предоставленной NIST.

Для получения обобщенного по всем подсистемам результирующего решения использовался инструментарий для смешивания по критерию минимизации функции стоимости DCF.

4.1.2 Secondary system (SVID-2)

Вторичная (secondary) система отличается использованием на этапе обучения UBM речевой базы NIST SRE 2005 вместо NIST SRE 2008.

4.1.3 Secondary system (SVID-3)

Еще одна вторичная (secondary) система была сформирована путем комбинирования первичной (SVID 1) и вторичной систем (SVID 2).

4.2 Смешивание подсистем

Обобщенное решение по всем подсистемам было основано на методе взвешенного голосования:

$$d(x) = \sum_{i=1}^S \alpha_i d_i(x) + \Theta$$

где $d_i(x)$ - выходное значение i -й подсистемы, α_i - весовой коэффициент для i -й подсистемы, Θ - пороговое значение, S - число подсистем.

Калибровка общего решения производилась на речевой базе NIST SRE 2005, которая не использовалась для обучения базовых GMM-SVM подсистем.

В связи с тем, что оценка DCF при высокой стоимости ошибки ложного пропуска не является статистически устойчивой, оптимизация коэффициентов α_i производилась простейшим методом перебора. При этом в качестве функции минимизации ошибки использовалась непосредственно функция DCF .

5. РЕЗУЛЬТАТЫ

В таблицах 4-5 приведены характеристики систем идентификации дикторов ООО «Центр Речевых Технологий» до конкурса и представленной на конкурсе NIST SRE 2010.

Как следует из приведенных данных, было обеспечено:

- повышение точности идентификации (понижение EER) в 3-4 раза;
- робастность идентификации в различных условиях (шумы, реверберация);
- сохранение достигнутых параметров точности и робастности при кросс-канальных сравнениях.

Таблица 4

Каналы	Система до конкурса EER, %	Система после конкурса EER, %
Микрофон-микрофон (различные микрофоны)	15-18	6,0
Микрофон-телефон (различные телефонные каналы)		4,9
Телефон-телефон (различные телефонные трубки и каналы)		5,0

Таблица 5 - Смешанный корпус микрофон – GSM-канал

Система до конкурса	Система после конкурса
---------------------	------------------------

Длительность, сек	16	32	80	Длительность, сек	17	29	77
16	15,2	14,0	14,0	16	8,0	6,3	3,8
32		12,9	11,4	32		4,5	2,7
80			8,9	80			1,3

Исходя из официально предоставленных NIST материалов (см. рисунок 4), система идентификации дикторов ООО «Центр Речевых Технологий» заняла на конкурсе NIST SRE 2010:

- 2-е место по уровню калибровки (1-е место среди коммерческих компаний);
- 7-е место по фактической стоимости (actual cost) технологии - официальной метрике NIST (2-е место среди коммерческих компаний).

На рисунке 4 используются следующие обозначения:

Min DCF – минимальное значение DCF, порог определяется NIST;

Act DCF – реальное значение DCF, порог определен участником;

Min DCF / Act DCF – степень калибровки системы

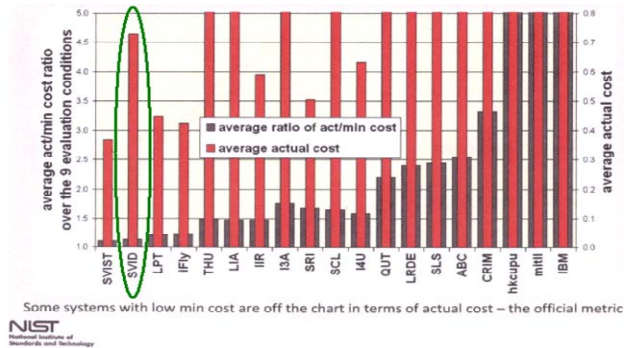


Рисунок 4: Фактическая стоимость (actual cost) детектирования системы по всем 9-ти DET-граммам.

6. ЗАКЛЮЧЕНИЕ

В рамках подготовки к конкурсу NIST SRE 2010 была произведена модернизация системы идентификации дикторов ООО «Центр Речевых Технологий», что обеспечило значительное повышение точности идентификации (понижение EER) в 3-4 раза, повышение робастности идентификации в различных условиях (шумы, реверберация) и повышение скорости системы при построении голосовых моделей по фонограмме.

В настоящее время предложенный в рамках NIST SRE 2010 подход к идентификации дикторов используется в системе VoiceNet ID ведения и автоматизации национального фоноучета Мексики [7].

7. СПИСОК ЛИТЕРАТУРЫ:

- [1] Bimbot F. et al. A Tutorial on Text-Independent Speaker Verification. - EURASIP Journal on Applied Signal Processing, 2004, №4, p.p. 430–451.
- [2] Reynolds D. Experimental evaluation of features for robust speaker identification. – IEEE Trans. On Speech and Audio Processing, 1994, vol. 2, №4, p.p. 639-643.
- [3] Коваль С.Л., Лабутин П.В., Раев А.Н. Патент РФ 2230375 от 10.06.2004 «Метод распознавания диктора и устройство для его осуществления».
- [4] Коваль С.Л., Лабутин П.В., Малая Е.В., Прощина Е.А. Идентификация дикторов на основе сравнения статистик основного тона голоса // В сб. трудов XV международной научной конференции «Информатизация и информационная безопасность правоохранительных органов». – М.: Академия управления МВД России, 2006. С. 324-327.
- [5] Vogt R., Baker B., Sridharan S. Modeling session variability in text-independent speaker verification // In Proc. Eurospeech, Lisbon, Portugal, Sept. 2005, pp. 3109–3112.
- [6] Rosenberg A. et al. Cepstral channel normalization techniques for HMM-based speaker verification // In Proc. ICSLP-94, pp. 1835-1838.
- [7] Тимофеев А.В. Распределённая система фоноучёта «VoiceNet ID». – Речевые технологии, 2009, №2, с. 69-73.

Об авторах

Юрий Николаевич Матвеев – д.т.н., руководитель отдела верификации и идентификации диктора ООО «Центр Речевых Технологий», Санкт-Петербург.

Электронный адрес matveev@speechpro.com.

Константин Константинович Симончик – научный сотрудник отдела верификации и идентификации диктора ООО «Центр Речевых Технологий», Санкт-Петербург.

Электронный адрес simonchik@speechpro.com.

An Adaptive Algorithm of Iris Image Key Points Detection

Elena A. Pavelyeva, Andrey S. Krylov

Laboratory of Mathematical Methods of Image Processing

Faculty of Computational Mathematics and Cybernetics

Lomonosov Moscow State University

E-mail: paveljeva@yandex.ru, kryl@cs.msu.ru

Abstract

A modification of iris recognition algorithm using Hermite transform to find iris key points has been developed. The iris image mask for the areas free of eyelids, eyelashes and glares is constructed. The key points are found inside the masked iris region. The use of eye rotation compensation based on Polar Hermite Transform before iris parameterization makes the method more effective. The proposed algorithm is tested with CASIA-IrisV3 database. The estimation of optimal number of iris key points is analyzed.

Keywords: biometrics, iris recognition, Hermite transform, iris key points, mask.

1. INTRODUCTION

The extraction of iris feature parameters consists of 2 steps: iris segmentation and iris parameterization. At the segmentation step [1] the iris areas which are free of glares, eyelashes and eyelids are determined (Fig. 1). These areas are called as iris mask. The parameterization step [2, 3] extracts the iris informative parameters from the masked iris area. These parameters are used at the iris comparison and iris identification stage. These techniques are under intensive investigation now. Special contest had been organized by SOCIA Lab for best mask construction for UBIRIS.v2 database [4] of visible wavelength iris images captured at-a-distance and on-the-move [5].



Figure 1: The examples of eye images and their masks.

In our previous work [6] the key points based iris recognition method using Hermite transform [7] was proposed. This method showed good results for CASIA-IrisV3 [8] database, although 0.23% of iris images were recognized incorrectly. CASIA database contains iris images (Fig. 2) with infrared illumination.

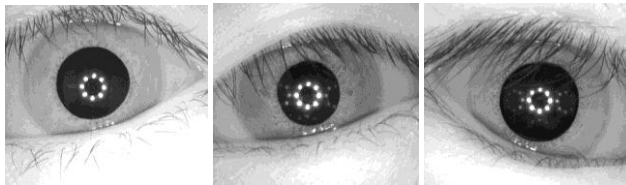


Figure 2: The examples of CASIA iris images.

This work proposes the modification of the previous iris parameterization method. In this work the iris mask which determines the iris areas free of eyelids, eyelashes and glares is built automatically for each eye from the database. The intersection of taken eyes masks forms the common pair mask for each eyes pair. The key points are then selected only inside the common mask region and these key points are unique for the taken pair.

The structure of the article is follows: section 2 describes the algorithm of iris mask construction. Section 3 proposes Polar Hermite Transform algorithm for eye rotation determination. The composition of common images mask is described in section 4. In section 5 we estimate the optimal number of key points for iris parameterization and present the experimental results for CASIA-IrisV3 database.

2. IRIS MASK CONSTRUCTION

The general scheme of iris recognition method described in [6] is the following:

- After iris localization the iris is mapped to a rectangular image [9] (Fig.3).
- Then the iris key points are selected using the Hermite transform method in the parameterization region of the normalized iris image.
- At the identification stage the comparisons between key points matrixes are used [6, 9].

In [6] we used only the top right quarter of normalize iris image as the parameterization region (Fig. 3). In this work we construct the parameterization region (the iris mask) using the whole normalized iris image.



Figure 3: Iris localization and normalization.

To compare iris images correctly the iris mask must exclude the areas occluded by eyelids, eyelashes and glares. Otherwise many key points will correspond to these bad areas (Fig.4). This is caused by the form of Hermite transform function $\varphi_{1,0}$ found optimal for iris convolution analysis in [6] (Fig.5).

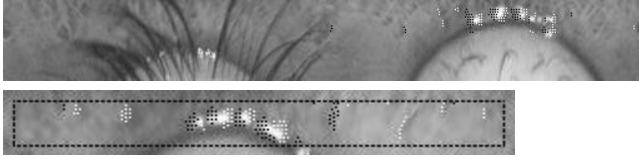


Figure 4: The normalized iris image which parametrization region (inside the dotted line) is occluded by eyelid and eyelashes.

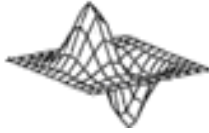


Figure 5: Hermite transform function

$$\varphi_{1,0}(x, y) = -\frac{\sqrt{2}}{\pi} \cdot x \cdot e^{-(x^2+y^2)}.$$

The mask value is equal 1 in the point (i, j) if this point is not occluded by glares, eyelids or eyelashes. Otherwise the mask value is 0. The mask construction consists of two steps. We find:

1. the points occluded by glares and eyelashes,
2. the points occluded by eyelids.

To find glares and eyelashes Canny Edge detector [10] is used (Fig. 6). The Canny algorithm thresholds in this work are TL=0, TH=50. With these values we detect glares and eyelashes edges and suppress the detection of the edges in the iris texture regions.

The eyelid is often less contrast than glares so the eyelid edges are only partially detected by Canny edge detector with the chosen thresholds. To find the whole eyelid in the normalized iris image we use the following formula (the same formula as is used to find the iris boundary at the iris segmentation stage):

$$\max_{r, x_0, y_0} \left| G_\sigma(r) * \frac{\partial}{\partial r} \int_{r, x_0, y_0} \frac{I(x, y)}{2\pi r} ds \right|.$$

The results of iris mask detection are shown in Fig.6.

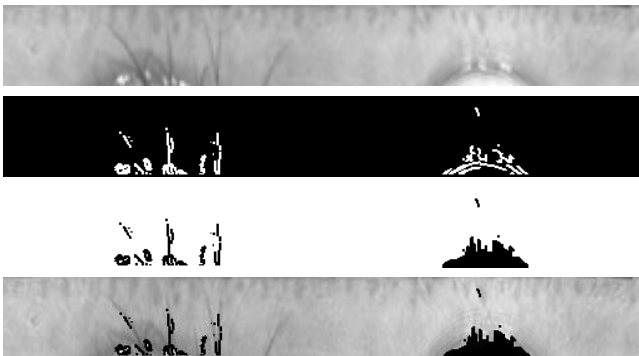


Figure 6: The normalized iris image, its edges by Canny edge detector, obtained iris mask and the result of masking (black areas are excluded by the mask).

3. EYE ROTATION ESTIMATION USING POLAR HERMITE TRANSFORM

If one of the images is rotated through the angle θ relative to the other one then this angle must be taken into account at the comparison stage. So we have to shift the rotated normalized iris image cyclically before parameterization. This operation is equivalent to the initial image rotation (Fig. 7).



Figure 7: Normalized iris images before rotation.

To determine the rotation angle we use the Polar Hermite Transform. Let $l_{i,j}$ be the cartesian Hermite coefficients [11, 12] of the image, i.e. the coefficients of expansion of iris image intensity information in series of Hermite functions ψ which are eigenfunctions of the Fourier Transform.

$$l_{i,j} = \int_{-\infty}^{\infty} dy \int_{-\infty}^{\infty} I(x, y) \cdot \psi_{i,j}(x, y) dx.$$

The polar Hermite coefficients $l_{n-k,k}^p$ are calculated from the cartesian Hermite coefficients $l_{n-m,m}$ using formulae [11, 12]:

$$l_{n-k,k}^p = \sum_{m=0}^n G_n^{cp}(m, k) \cdot l_{n-m,m}$$

$$G_n^{cp}(m, k) = \frac{1}{2\pi} \int_0^{2\pi} \alpha_{n-m,m}^c(\omega) \cdot \tilde{\alpha}_{n-k,k}^p(\omega) d\omega,$$

$$\alpha_{n-m,m}^c(\omega) = \sqrt{\frac{n!}{(n-m)!m!}} \cdot \cos^{n-m} \omega \cdot \sin^m \omega, \quad m=0, \dots, n,$$

$$\tilde{\alpha}_{n-k,k}^p(\omega) = \sqrt{\frac{2^n (n-k)!k!}{n!}} \cdot \begin{cases} \sqrt{2} \cos(n-2k)\omega, & 0 \leq k < \frac{n}{2} \\ 1, & k = n-k \\ \sqrt{2} \sin(n-2k)\omega, & \frac{n}{2} < k \leq n \end{cases}$$

The very useful property of polar coefficients is that the polar coefficients of rotated through the angle θ image are calculated fast:

$$\begin{bmatrix} l_{n-m,m}^p(\theta) \\ l_{m,n-m}^p(\theta) \end{bmatrix} = \begin{bmatrix} \cos(n-2m)\theta & \sin(n-2m)\theta \\ -\sin(n-2m)\theta & \cos(n-2m)\theta \end{bmatrix} \cdot \begin{bmatrix} l_{n-m,m}^p \\ l_{m,n-m}^p \end{bmatrix}.$$

So to calculate the polar coefficients for the image rotated through the angle θ we just multiply the rotation matrix and the polar coefficients of non-rotated image.

To calculate the angle of eye image rotation the images are first transformed to the normalized template ($N \times N$ square). The template has the pupil in the center of the iris and the pupil have

the fixed radius ($r' = N/3$). Then the iris is reduced to the fixed width of 64 pixels. The part of image which is inside the $N \times N$ square is used as normalized template image (Fig. 8b) and we use for the rotation estimation only the ring marked in Fig. 8c.

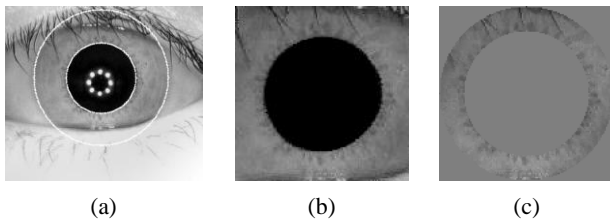


Figure 8: (a) – the input image, (b), (c) – the normalized template construction.

Then we calculate the polar Hermite coefficients $I_{n-i,i}$ for $n = 0, 1, \dots, 31$, $i = 0, 1, \dots, n$ for the inspected normalized eye image and polar coefficients for the same image rotated through the different angles $\varphi = \pm 2^\circ, \pm 4^\circ, \dots, \pm 20^\circ$. All these coefficients are compared with polar coefficients of the rotated image. As images comparison metric we use the sum of square differences between the corresponding coefficients of images. The angle with the smallest comparison distance is taken as rotation angle between the images. An example of the proposed algorithm for the images given in Fig. 7 is shown in Fig. 9.



Figure 9: The normalized iris images after rotation.

4. COMMON MASK OF TWO IRISES

After rotation correction the common mask of two images is formed. The common mask is an intersection of the masks of each iris image (Fig. 10). To reduce the influence of glares, eyelashes and eyelids areas at the iris key points extraction stage we narrow the common mask by the fixed number of pixels.

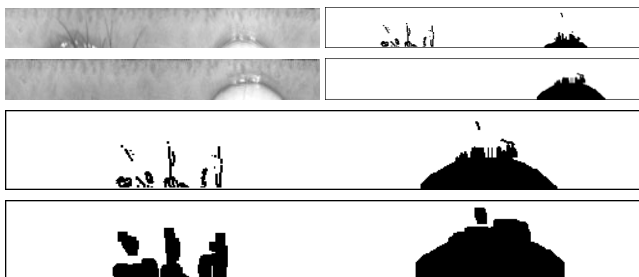


Figure 10: The common mask of two iris images after rotation elimination (the mask is marked by white color) and the narrowed common mask.

5. PRACTICAL APPLICATION OF THE ADAPTIVE ALGORITHM OF IRIS IMAGE KEY POINTS DETECTION

Before the iris key points extraction the histogram of normalized iris image is equalized inside the common mask region (Fig. 11). Key points are found for the histogram equalized iris images. The value $F(x_0, y_0) = (I(x, y) * \varphi_{1,0}(x, y))(x_0, y_0)$ is calculated in each point of masked image for key points extraction, where $\varphi_{1,0}$ is the Hermite transform function. As an iris code (iris key points) we take N points ($N = 300$, the choose of this N value will be argued below), which form two groups: $N/2$ points with the maximal F values (black points in Fig. 12) with the condition that the minimal distance between these points is 2 pixels, and similarly $N/2$ points with the minimum of F values (white points).

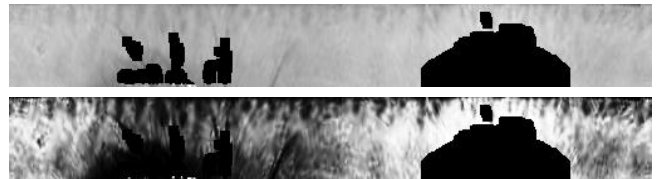


Figure 11: The normalized iris image with the common mask before and after histogram equalization.



Figure 12: The iris key points.

5.1 Estimation of optimal number of key points

In previous work [6] it was shown that the optimal number of iris key points is close to 150 points for the case of the top right quarter used as the parameterization region (Fig. 13).

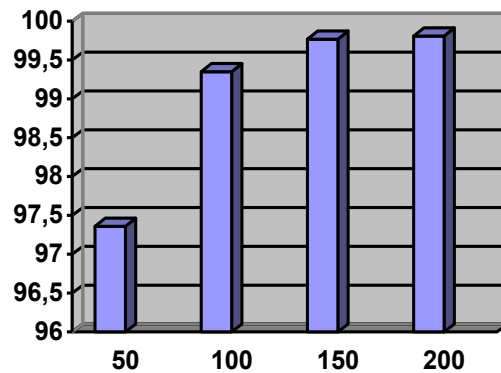


Figure 13: The diagram of correct recognition rate (in %) depending on number of key points.

Using this diagram the optimal number of key points for the masked parameterization region was selected. We realized the next calculations for each of two points group. Let M_i is the maximum of convolution modulus value for the image i for the taken key points group and the convolution value for the last

taken key point (150^{th} key point) from this group is $\frac{M_i}{k_i}$. We

averaged k_i values through the whole database and obtained the average value $k_{mean} = 2.08$. Thus for image i we found N_i

key points in the common mask region in the range $\left[\frac{M_i}{k_{mean}}, M_i \right]$.

Averaging of N_i values for the whole CASIA-IrisV3 database gave us the average value $N \approx 300$.

5.2 Results for CASIA-IrisV3 iris database

To evaluate the effectiveness of the proposed key points method all irises which were falsely recognized in [6] (using the fixed parameterization region) were tested. All these eyes are recognized correctly using the masked parameterization regions approach. So the iris key points recognition algorithm allows us to obtain 100% correct recognition rate with CASIA-IrisV3 database.

6. CONCLUSION

A modification of previously proposed iris recognition algorithm using Hermite transform to find iris key points has been developed. The iris mask which determines the iris areas free of eyelids, eyelashes and glares is constructed automatically for each eye. The key points are found inside the masked iris region. Eye rotation compensation was performed using Polar Hermite Transform. The proposed algorithm is tested with CASIA-IrisV3 database and showed 100% recognition rate. The future work will include speed optimization of the proposed algorithms and their application to visible wavelength iris data.

The work was supported by RFBR grants 10-07-00433, 10-01-00535 and by federal target program "Scientific and scientific-pedagogical personnel of innovative Russia in 2009-2013".

7. REFERENCES

- [1] H. Proenca. Iris Recognition: On the Segmentation of Degraded Images Acquired in the Visible Wavelength, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 32, no.8, pp. 1502-1516, Aug. 2010.
- [2] J. Daugman. How iris recognition works, IEEE Transactions on Circuits and Systems for Video Technology, vol. 14, no.1, pp. 21-30, 2004.
- [3] K. Hollingsworth, K. Bowyer, P. Flynn. The Best Bits in an Iris Code, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 31, no. 6, pp. 964-973, June 2009.
- [4] UBIRIS.v2 database. <http://iris.di.ubi.pt/ubiris2.html>
- [5] Special Issue on the Segmentation of Visible Wavelength Iris Images Captured At-a-distance and On-the-move, Image and Vision Computing, vol. 28, is. 2, pp. 213-284, February 2010.
- [6] E. Pavelyeva, A. Krylov. Hermite Transform Based Iris Key Points Selection and Analysis, Informatics and applications, vol.4, is. 1, 79-82, 2010 (in Russian).
- [7] J.-B. Martens. The Hermite transform-theory, IEEE Transactions on Acoustics, Speech, and Signal Processing, 1990. Vol. 38. No. 9. P. 1595-1606.

[8] CASIA-IrisV3 database.

<http://www.cbsr.ia.ac.cn/IrisDatabase.htm>.

[9] E. Pavelyeva, A. Krylov. Iris Identification Algorithm Using the Most Informative Iris Points, GraphiCon'2009 Conference proceedings, pp. 228-231, 2009 (in Russian).

(http://www.graphicon.ru/proceedings/2009/conference/se8/99/99_Paper.pdf)

[10] J. Canny. A computational approach to edge detection, IEEE Trans. PAMI, 1986, vol. 8. pp. 34-43.

[11] J.-B. Martens. Local Orientation Analysis in Images by Means of the Hermite Transform, IEEE Transactions On Image Processing, vol. 6, no. 8, August 1997, pp. 1103-1116, 1997.

[12] A.V. Kutovoi, A.S. Krylov. A New Method for Texture-Based Image Analysis, GraphiCon'2006 Conference proceedings, Novosibirsk, pp. 235-238, 2006.

(http://www.graphicon.ru/2006/proceedings/papers/we07_45_Kutovoi.pdf)

About the authors:

Elena A. Pavelyeva is a PhD student of Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University.

E-mail: paveljeva@yandex.ru.

Andrey S. Krylov is professor, head of the Laboratory of Mathematical Methods of Image Processing, Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University.

Email: kryl@cs.msu.ru

Averaging of Fingerprint Template with Respect to Elastic Deformations

Ushmaev Oleg, Institute of Informatics Problems, Russian Academy of Sciences, oushmaev@ipiran.ru
 Arutyunyan Artem, Nuclear Safety Institute, Russian Academy of Sciences, artem@ibrae.ac.ru

Abstract

The paper describes a technique of generation of an “undeformed”, i.e. averaged with respect to elastic deformations, fingerprint template. We offer to calculate the average “undeformed” fingerprint from multiple impressions. The averaging is based on the elastic model of fingerprint. Experiments show that the “undeformed” template improves fingerprint identification performance as measured by the FAR and FRR.

Key words: *fingerprint recognition, elastic deformations, template improvement.*

1. INTRODUCTION

The performance of fingerprint recognition algorithms is negatively affected by different factors: noises, differences in skin conditions, elastic distortions of fingerprints (fig.1) etc. In this paper we consider elastic deformations. The aim of our research is prior compensation (during enrollment) of elastic deformations of fingerprints.

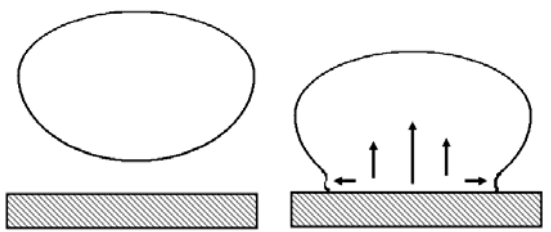


Figure 1: Origin of elastic deformations

There are different approaches to registration and modeling of elastic deformations. One of the first approaches was introduced by D.J. Burr [9], and used the concept of rubber masks. The way suggested by A.M. Bazen and S.H. Gerez [10, 11] is based on the thin-plate spline (TPS) models, firstly applied to biological objects by F.L. Bookstein [12]. This method requires determining correspondent points in two compared images (matching point) and it suffers from the lack of precision in case of few matching points. Modifications of TPS (approximate thin-plate splines and radial based function splines) were introduced by M. Fornefett, K. Rohr and H. Stiehl [13],[14]. They consider deformations of biological tissues. But this way also requires many matching points (more than 100) what is virtually impossible in fingerprint applications, because number of minutiae in fingerprint image rarely exceeds 50. This fact makes TPS and its variants hardly applicable to fingerprint deformations registration. Very interesting empirical approach has been suggested by R. Cappelli, D. Maio and D. Maltoni [15]. They developed analytical model of fingerprint deformation, however not specifying the algorithm for its registration.

The listed approaches and techniques don't provide us with direct tools for prior registration of fingerprint deformations. The only approach to prior deformation estimation has been proposed in [11]. Authors suggest that a fingerprint template can be averaged on-line. Additional information for averaging is obtained during each genuine match.

In this paper we combine averaging technique [11] and elastic model [16] of fingerprint deformation. The paper is organized as follows. Section 2 describes mathematical model of fingerprint deformations based on the elasticity theory. The technique of deformation averaging is in Section 3. Section 4 contains experimental results.

2. MODEL OF DEFORMATION

Let's represent a deformation of fingerprints as a mapping of images:

$$\mathbf{u} : \text{Im}_1 \rightarrow \text{Im}_2, \quad (1)$$

$$\text{Im}_2(x, y) = \text{Im}_1(x + \mathbf{u}(x, y), y + \mathbf{u}(x, y)). \quad (2)$$

where Im_1, Im_2 - fingerprint images, \mathbf{u} - displacement field.

Let's assume that fingerprint is elastic body. Dynamics of its deformation are defined by the following equations [17]:

$$\begin{aligned} \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xx}}{\partial x} + f_x &= \rho \frac{\partial^2 u_x}{\partial t^2}; \\ \frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + f_y &= \rho \frac{\partial^2 u_y}{\partial t^2}; \end{aligned} \quad (3)$$

$$\varepsilon_{xx} = \frac{\partial u_x}{\partial x}; \quad \varepsilon_{yy} = \frac{\partial u_y}{\partial y}; \quad \varepsilon_{xy} = \varepsilon_{yx} = \frac{1}{2} \left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \right);$$

where u_x, u_y are the x - and y - components of displacements,

$\mathbf{f} = (f_x, f_y)$ is external forces, ε is the strain tensor, σ is the stress tensor, ρ is the mass density.

Relations between tensors ε and σ obey Hooke's law:

$$\begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix} = \mathbf{C} \begin{pmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{pmatrix}, \quad (4)$$

$$\mathbf{C} = \frac{1}{E} \begin{pmatrix} 1 & -\nu & 0 \\ -\nu & 1 & 0 \\ 0 & 0 & (1+\nu) \end{pmatrix}. \quad (5)$$

where \mathbf{C} is the stiffness matrix, E is the Young's modulus, ν is the Poisson's ratio.

As known from the elasticity theory [17], steady-state solution (when the fingerprint is immobile in the surface of contact with a scanner) of (3) minimizes the following energy function:

$$\begin{aligned} E &= - \iint_S (u_x f_x + u_y f_y) dx dy + \\ &+ \frac{1}{2} \iint_S (\varepsilon_{xx} \sigma_{xx} + \varepsilon_{yy} \sigma_{yy} + \varepsilon_{xy} \sigma_{xy}) dx dy. \end{aligned} \quad (6)$$

As shown in [16], the energy function can't be directly calculated from (6), because external forces $\mathbf{f} = (f_x, f_y)$ are unknown. We offer approximate solution of (6) that based on measurable fingerprint features.

Let's assume, that we have found paired minutiae sets $\{\mathbf{p}_i\}_{i=1}^m$ and $\{\mathbf{q}_i\}_{i=1}^m$ on the first and second images respectively (fig.2). We replace action of forces $\iint_S (u_x f_x + u_y f_y) dx dy$ with mean-square discrepancy:

$$L^2(\mathbf{u}) = \frac{1}{m} \sum_{i=1}^m \|\mathbf{p}_i + \mathbf{u}(\mathbf{p}_i) - \mathbf{q}_i\|^2. \quad (7)$$

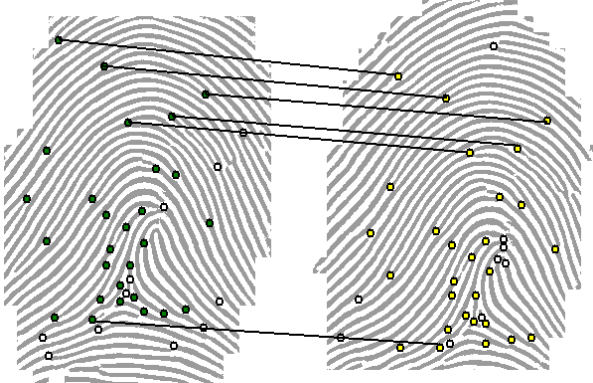


Figure 2: Minutiae correspondence

The final energy functional is a sum of the internal deformation energy and the substitute of the external forces work (7):

$$D(\mathbf{u}) = E_d(\mathbf{u}) + \alpha L^2(\mathbf{u}), \quad (8)$$

where α is the weight coefficient, E_d is the internal deformation energy:

$$E_d = \frac{1}{2} \iint_S (\varepsilon_{xx} \sigma_{xx} + \varepsilon_{yy} \sigma_{yy} + \varepsilon_{xy} \sigma_{xy}) dx dy. \quad (9)$$

Minimum of (8) defines relative deformation of the first image Im_1 to the second image Im_2 . An example of fingerprint deformation is in the Fig. 3.

3. MODEL OF UNDEFORMED FINGERPRINT

Let's assume that we enrolled $k+1$ fingerprint images: Im_0, Im_1, \dots, Im_k . Relative deformation of Im_j to Im_0 is denoted as \mathbf{u}_j . Let's define an "undeformed" fingerprint image as a minimum of the following function:

$$\mathbf{u}_{\min} = \min_{\mathbf{u}} \left(\sum_{j=1}^k E_d(\mathbf{u} - \mathbf{u}_j) + E_d(\mathbf{u}) \right). \quad (10)$$

Physically the function (10) is the total energy of deformation from condition \mathbf{u}_{\min} to every impression Im_0, Im_1, \dots, Im_k . "Average" fingerprint image is result of the following mapping $Im_{\min} = \mathbf{u}_{\min}(Im_0)$ (fig.4).

The solution of (10) depend only on positions of minutiae points in fingerprint images. We can calculate the "undeformed" condition not only for fingerprint image but also for the ISO-compliant fingerprint template. It allows adding the proposed technique on operating fingerprint identification systems.

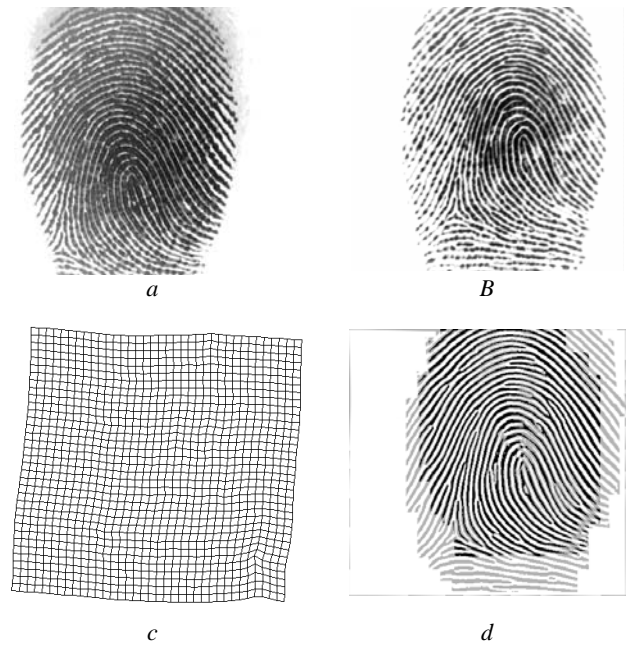


Figure 3: Example of deformation calculation (a,b – source images, c – displacement field, d- binarized image overlap).



Figure 4: Deformation averaging (undeformed image is on the right)

4. EXPERIMENTS

Let's briefly describe our experiments with the "averaging" of fingerprint templates. We employ the Biolink fingerprint recognition SDK and the public available FVC2002DB1 dataset [18]. The FVC2002DB1 contains 8 impressions of 100 different fingerprints (as total 800 images). Let's denote it as Im_j^i , where i is the impression index, j is the fingerprint index. We randomly select 4 impression subset i_1, i_2, i_3, i_4 . The query dataset contains averaged fingerprint templates generated from $Im_{j_1}^{i_1}, Im_{j_2}^{i_2}, Im_{j_3}^{i_3}, Im_{j_4}^{i_4}$ (100 templates). The target dataset contains 400 templates (4 impressions of 100 fingerprints). As total, we have 400 genuine and 39600 impostor matches for each subset i_1, i_2, i_3, i_4 . The we measure FAR and FRR. DET curves of fingerprint recognition are in the Fig. 5. The "undeformed" fingerprint template significantly improves performance as measured by the DET.

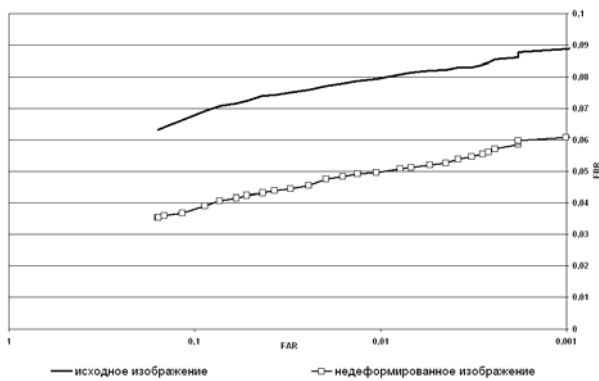


Figure 5: Performance improvement (FVC2002DB1)

CONCLUSION

We proposed the technique of generation of “undeformed” fingerprint image and generation of “undeformed” fingerprint template. The technique allows improving fingerprint recognition performance as measured by the FAR and FRR. In further we plan fusion of our fingerprint “averaging” with mosaicing technique [19,20].

ACKNOWLEDGEMENTS

This work was supported by RFBR, grant #10-07-00433.

REFERENCES

1. Wilson C.L., Watson C.I., Garris M.D., and Hicklin A., Studies of Fingerprint Matching Using the NIST Verification Test Bed (VTB) // available at http://sequoyah.nist.gov/pub/nist_internal_reports/ir_7020.pdf
2. Lee H.C. and Gaenssley R.E., Advances in Fingerprint Technology, Elsevier, New York, 1991.
3. S. Pankanti, S. Prabhakar and A.K. Jain, On the Individuality of Fingerprints, IEEE Trans. PAMI, 2002, 24(8), pp. 1010-1025.
4. Eleccion M. Automatic Fingerprint Identification, IEEE Spectrum, 1973, 10, p.36-45.
5. Halici U., Jain L.C., Erol A. Introduction to Fingerprint Recognition, Intelligent Biometric Techniques in Fingerprint and Face Recognition, CRC Press, 1999.
6. Wayman J., et al. Biometric Systems: Technology, Design and Performance Evaluation. – London: Springer Verlag, 2005.
7. Watson C., Wilson C., Indovina M., Cochran B. NISTIR 7249, Two Finger Matching With Vendor SDK Matchers, July 2005.
8. Jain A.K., Hong L., Pankanti S., Bolle R. An Identity-Authentication System Using Fingerprints // Proc. of IEEE, 1997, 85(9), p.1365-1388.
9. Burr D.J., "A Dynamic Model for Image Registration" Computer Graphics Image Processing Vol. 15 pp. 102-112, 1981
10. Bazen A.M., Gerez S.H., Thin-Plate Spline Modelling of Elastic Deformation in Fingerprints, Proceedings of 3rd IEEE Benelux Signal Processing Symposium, 2002.
11. A. Ross, S. Dass and A. K. Jain, Estimating Fingerprint Deformation, Proc. of International Conference on Biometric Authentication (ICBA), (Hong Kong), LNCS vol. 3072, pp. 249-255, Springer Publishers, July 2004.
12. Bookstein F.L., Principal Warps: Thin-Plate Splines and the Decomposition of Deformations, IEEE Trans. PAMI, 1989, 11(6), pp. 567-585.
13. M. Fornefett, K. Rohr and H.S. Stiehl, Elastic Medical Image Registration Using Surface Landmarks with Automatic Finding of Correspondences, In A. Horsch and T. Lehmann, editors Proc.

Workshop Bildverarbeitung für die Medizin, Informatik aktuell, München, Germany, Springer-Verlag Berlin Heidelberg, 2000, pp. 48-52.

14. M. Fornefett, K. Rohr and H.S. Stiehl, Radial Basis Functions with Compact Support for Elastic Registration of Medical Images, Image and Vision Computing, 19 (1-2), 2001, pp. 87-96.
15. Raffaele Cappelli, Dario Maio, Davide Maltoni, Modelling Plastic Distortion in Fingerprint Images, ICAPR2001, pp. 369-376.
16. Sergey Novikov and Oleg Ushmaev, "Registration and Modelling of Elastic Deformations of Fingerprints". Biometric Authentication (ECCV 2004 International Workshop, BioAW2004), Prague, Czech Republic, May2004, Proceedings. Springer, Eds. Davide Maltoni, Anil K. Jain. pp. 80-88.
17. Landau L.D., Lifshitz E.M. Theory of elasticity, 3rd ed. Butterworth-Heinemann, 1986.
18. FVC2002, the Second International Competition for Fingerprint Verification Algorithms, bias.csr.unibo.it/fvc2002
19. Choi K., Choi H., Kim J. Fingerprint Mosaicing by Rolling and Sliding // Proc. AVBPA2005, p.260-269
20. Ratha N., Connell J., Bolle R. Image Mosaicing for Rolled Fingerprint Construction // Proc. ICPR1998, p. 1651-1653.

About authors



Oleg Ushmaev received the MS degree in Math from Lomonosov Moscow State University in 2002 and the PhD degree in Computer Science from the Institute of Informatics Problems, Russian Academy of Sciences, in 2004. He is leading scientist in the Institute of Informatics Problems. Until 2009 he worked as R&D Director with fingerprint vendor Biolink Solutions. His research interests include multimodal biometrics, fingerprint recognition and statistical pattern recognition.



Artem Arutyunyan received the MS degree in Computer Science for Lomonosov Moscow State University in 2002 and the PhD degree in Computer Science from the Institute of Informatics Problems, Russian Academy of Sciences, in 2010. Now he is scientist in the Nuclear Safety Institute, Russian Academy of Sciences. His research interests include fingerprint recognition, biometric systems performance evaluation and statistical pattern recognition.

Artem Arutyunyan received the MS degree in Computer Science for Lomonosov Moscow State University in 2002 and the PhD degree in Computer Science from the Institute of Informatics Problems, Russian Academy of Sciences, in 2010. Now he is scientist in the Nuclear Safety Institute, Russian Academy of Sciences. His research interests include fingerprint recognition, biometric systems performance evaluation and statistical



Young Scientists School

GraphiCon'2010

September 20-24, 2010
Saint-Petersburg, Russia

Оценивание ожидаемой работоспособности алгоритмов измерения координат объектов

Павел Бабаян, Сергей Смирнов

Кафедра Автоматики и информационных технологии в управлении
Рязанский государственный радиотехнический университет, Рязань, Россия
aitu@rsreu.ru

Аннотация

В работе рассмотрена методика оценивания ожидаемой работоспособности алгоритмов измерения координат объектов. Она может быть использована при разработке программного обеспечения для автономных комплексов обнаружения и сопровождения объектов, ориентированных на работу в сложной фоновой обстановке. В частности, данная методика использовалась при разработке алгоритма слежения за объектами с автоматическим выбором метода измерения координат при изменении фоновой обстановки [1].

Ключевые слова: изображение, измерение координат объектов, оценка работоспособности.

1. ВВЕДЕНИЕ. ОБЩИЙ ПОДХОД К РЕШЕНИЮ ЗАДАЧИ

Для повышения помехоустойчивости сопровождения объектов требуется эффективно решать задачу измерения координат объектов. Однако каждый алгоритм измерения координат характеризуется определенными требованиями к условиям наблюдения, при выполнении которых он работает наиболее эффективно. Зачастую условия наблюдения заранее не известны, поэтому нет возможности задать алгоритм, обеспечивающий наилучшее качество измерения координат. Таким образом, возникает задача оценивания ожидаемой работоспособности алгоритма измерения координат с целью последующего принятия решения о его использовании.

Оценивание работоспособности алгоритма измерения координат в зависимости от характеристик фоновых условий связано с введением характеристик, описывающих качество работы алгоритма. Рассмотрим два вида таких характеристик: количественные характеристики работоспособности (КХР) и признаки работоспособности (ПР). Для использования КХР необходимо знать эталонные данные о траектории движения объектов, что невозможно в процессе функционирования систем реального времени. Для вычисления ПР не требуются эталонные данные, однако ПР дают лишь оценочную, не всегда адекватную характеристику работоспособности. Для использования этих характеристик предварительно необходимо установить статистическую связь между КХР и ПР. Таким образом, на основе данной связи можно по значениям ПР получить оценки КХР.

Ввиду чрезвычайной сложности построения моделей фоновой обстановки, практически отсутствует возможность аналитического построения функции, устанавливающей соответствие между значениями ПР и КХР. По этой причине разработана процедура нахождения КХР на основе обучения на большом количестве видеосюжетов.

Для проведения такого обучения заранее формируется база

данных видеосюжетов с различной фоновой обстановкой. Для каждого объекта экспертом фиксируется эталонная траектория и размеры объекта. Каждый видеосюжет обрабатывается алгоритмом измерения координат объектов, и формируются измеренные траектории. На основе сопоставления эталонных и измеренных траекторий вычисляются КХР.

Затем производится статистический анализ КХР и ПР, рассчитанных по первым кадрам видеосюжетов. Опираясь на предположение о наличии статистической связи между ними можно найти характер этой зависимости для алгоритма измерения координат.

Таким образом, после нахождения зависимости между ПР и КХР можно в режиме реального времени при поступлении очередного кадра видеосюжета произвести оценку КХР и объективно оценить работоспособность алгоритма.

На рисунке 1 схематично приведена описанная выше методика оценивания работоспособности алгоритма измерения координат.



Рисунок 1: Методика оценивания работоспособности алгоритмов измерения координат.

2. ВЫБОР КХР И ПР

В качестве КХР необходимо выбирать характеристики, описывающие измерение координат с точки зрения точности или продолжительности слежения.

В данной работе используются две характеристики: двоичная КХР и числовая КХР, характеризующая число кадров до первого срыва. Под срывом будем понимать превышение ошибочной измерения координат размеров объекта. Двоичная КХР может принимать значения «следит»/«не следит». В случае если для заданного числа кадров не наблюдается срыв, то КХР имеет значение «следит» (1), в противном случае критерий имеет значение «не следит» (0).

Признаки работоспособности должны быть связаны с теми особенностями объекта, которые тот или иной алгоритм измерения координат использует для отделения объекта от фо-

на. В данной работе рассматриваются четыре алгоритма измерения координат:

- алгоритм на основе корреляционного совмещения изображений (алгоритм I);
- алгоритм на основе байесовской сегментации (алгоритм II);
- алгоритм на основе пространственной фильтрации (алгоритм III);
- алгоритм на основе пространственно-временной фильтрации (алгоритм IV).

ПР для алгоритма I может быть основан на сравнении временной изменчивости объекта и степени отличия объекта от фона [1]. Работоспособность алгоритма II главным образом определяется различиями статистических характеристик фона и объекта. В качестве признака оценивающего работоспособность данного алгоритма может использоваться разность энтропий [2]. Признак для алгоритма III базируется на оценке амплитуды яркостного импульса, представляющего объект на изображении [1]. Для алгоритма IV ПР основан на оценке степени временной изменчивости изображения объекта [1].

3. НАХОЖДЕНИЕ ВЗАИМОСВЯЗИ МЕЖДУ КХР И ПР

Нахождение взаимосвязи между двоичной КХР и ПР

Переход от ПР к двоичной КХР реализуется путём сравнения признаков работоспособности с пороговыми значениями.

Рассмотрим процедуру нахождения порогового значения для произвольного алгоритма измерения координат. Всё множество видеосюжетов можно разбить на подмножество X_1 , для которых КХР приняла значение 1, и подмножество X_2 , включающее все остальные видеосюжеты.

В качестве критерия выбора порога используется критерий максимального правдоподобия. Для этого приближенно опишем плотность распределения ПР внутри каждого множества X_1 и X_2 с помощью нормального распределения и найдём выборочные средние m_i и дисперсию σ_i для каждого множества X_i . Поиск порогового значения сводится к вычислению точки пересечения гауссоид по соотношениям

$$\alpha_{1,2} = \frac{-(2m_1\sigma_2^2 - 2m_2\sigma_1^2) \pm \sqrt{D}}{2(\sigma_1^2 - \sigma_2^2)}, \text{ где}$$

$$D = (2m_1\sigma_2^2 - 2m_2\sigma_1^2)^2 - 4(\sigma_1^2 - \sigma_2^2) \cdot (\sigma_1^2 m_2^2 - \sigma_2^2 m_1^2 - 2\sigma_1^2 \sigma_2^2 \log \sqrt{\sigma_1 / \sigma_2}).$$

Из полученных $\alpha_{1,2}$ выбирается в качестве порогового значения то, которое лежит в диапазоне $[m_1, m_2]$.

Нахождение взаимосвязи между числовой КХР и ПР

Будем предполагать, что связь между КХР и ПР описывается корреляционной зависимостью. Таким образом, для нахождения зависимости нужно решить задачу нахождения линии регрессии. Учитывая тот факт, что длина видеосюжетов ограничена, можно предполагать, что на некоторых сюжетах момент срыва не будет достигнут. Непосредственно использовать значения КХР полученные для этих сюжетов для расчета линии регрессии нельзя, потому что продолжительность сюжета не позволяет установить фактическое значение признака. Полученные значения КХР дают лишь нижнюю оценку КХР. Однако исключение этих сюжетов приведет к значительному уменьшению выборки.

Таким образом, методику построения линии регрессии можно описать в виде последовательности шагов:

- 1) построить линию регрессии по точкам, содержащей только те значения КХР, в которых срыв произошел раньше окончания сюжетов;
- 2) составить выборку, включающую точки, в которых срыв произошел раньше окончания сюжетов, и все точки, лежащие выше новой линии регрессии. Перестроить прямую;
- 3) повторять шаг № 2 до тех пор, пока разница коэффициентов линии регрессии при соседних итерациях не окажется меньше заданного порога.

Найденные коэффициенты линии регрессии могут использоваться для расчёта КХР, основываясь на значениях ПР.

4. АНАЛИЗ ТОЧНОСТИ ОЦЕНИВАНИЯ КХР

При оценивании двоичной КХР возможны ошибки двух видов:

- ошибка первого рода – ошибочное принятие решения, что алгоритм неработоспособен,
- ошибка второго рода – ошибочное принятие решения, что алгоритм работоспособен.

В таблице 1 приведены значения ошибок классификации для каждого алгоритма.

Также в таблице 1 приведены значения среднего квадрата отклонения от линии регрессии для точек, участвующих в построении линии регрессии на последней итерации при определении числовой КХР по значениям ПР.

Таблица 1: Ошибки определения КХР для различных алгоритмов

Алгоритм	Двоичная КХР		Числовая КХР СКО, в кадрах
	Ошибка первого рода, %	Ошибка второго рода, %	
I	0	3,64	102,56
II	0	6,383	82,34
III	0	7,89	60,9
IV	0	18,2	79,2

5. ЗАКЛЮЧЕНИЕ

Применение описанной методики позволяет обоснованно подходить к выбору алгоритма измерения координат объекта в текущих условиях наблюдения. Проведенный анализ точности оценивания КХР показывает, что СКО оценивания числовой КХР оказалось высокой, что затрудняет её практическое применение. В случае двоичной КХР точность оценивания составляет 3,6-18%, что является приемлемым для принятия решения о работоспособности алгоритма.

6. БЛАГОДАРНОСТИ

Работа проводилась при поддержке Федерального агентства по науке и инновациям (госконтракт № 02.740.11.0002).

7. ЛИТЕРАТУРА

- [1] Алпатов Б.А., Бабаян П.В., Смирнов С.А. *Автоматическое слежение за объектами при отсутствии априорных сведений о фоноцелевой обстановке // Цифровая обработка сигналов. – 2009. – №3. – С. 52-56.*
- [2] Бакут П.А., Лабунец В.Г. *Телевизионная следящая система с байесовским дискриминатором цели // Зарубежная радиоэлектроника. – 1987. – №10 – С. 81-93.*

ИСПОЛЬЗОВАНИЕ ИНФОРМАЦИИ О ЦВЕТЕ В АЛГОРИТМЕ ВЫДЕЛЕНИЯ ЛИЦ НА ИЗОБРАЖЕНИЯХ

Шмаглит Л.А., Голубев М.Н.
Кафедра динамики электронных систем

Ярославский государственный университет им. П.Г. Демидова, Ярославль, Россия
connect@piclab.ru

АННОТАЦИЯ

Система обнаружения лиц решает следующую задачу: по произвольному изображению на входе системы определить имеются ли на этом изображении лица, и если да, то указать, где находится каждое лицо и каковы его размеры. За последние несколько лет для решения этой задачи было предложено множество алгоритмов, использующих различные подходы. Однако, как бы ни был хорош алгоритм, на его выходе возникают ошибки двух видов – пропущенные лица и ложные срабатывания. В данной работе делается попытка уменьшить число ошибок второго рода путем использования информации о цвете, которая наряду с информацией о яркости пикселей присутствует в полноцветных изображениях.

Ключевые слова: алгоритм выделения лиц, кластер кожи, ошибки второго рода, уровень ложных срабатываний.

1. ВВЕДЕНИЕ

В качестве алгоритма выделения лиц в работе рассматривается алгоритм, предложенный П. Виолой и М. Джонсом [1], который использует процедуру обучения, основанную на бустинге [2]. Данный алгоритм детектирования лиц на цифровом изображении, основан на выделении признаков и учитывает лишь яркостную составляющую изображения, полностью игнорируя информацию о цвете. Однако из анализа примеров детектирования следует, что рассмотренный алгоритм в некоторых случаях допускает ошибки второго рода (ложные срабатывания) на объектах, которые по цветовой гамме значительно отличаются от цвета лица. Эти ошибки можно значительно уменьшить, добавив к алгоритму детектирования этап проверки областей-кандидатов на цветное соответствие [3].

Для модификации алгоритма детектирования лиц были выбраны два подхода. Рассмотрим их более подробно.

2. ЯВНОЕ ЗАДАНИЕ КЛАСТЕРА КОЖИ

Первый подход использует классификатор, который определяется явными численными правилами, ограничивающими кластер цвета кожи в пространстве RGB. Пиксель (R, G, B) относится к коже, если для него выполнены все следующие условия:

- $R > 95$;
- $G > 40$;
- $B > 20$;

- $\max\{R, G, B\} - \min\{R, G, B\} > 15$;
- $|R - G| > 15$;
- $R > G$;
- $R > B$.

Данный алгоритм не требует перехода в цветовую систему, отличную от цветовой системы исходного изображения, не требует больших вычислительных ресурсов, прост и надежен. Простота классификатора и применение его только к областям-кандидатам позволили сократить количество ошибок второго рода в алгоритме детектирования лиц, практически не увеличив вычислительную нагрузку. Доля пикселей, которая должна быть детектирована как кожа, чтобы алгоритм отнес данное окно к классу лиц, задается в параметре STDR (Skin Tone Detection Rate).

3. НЕПАРАМЕТРИЧЕСКАЯ МОДЕЛЬ РАСПРЕДЕЛЕНИЯ ОТТЕНКА КОЖИ

Второй подход использует разработанный нами алгоритм и относится к непараметрическим моделям распределения оттенка кожи. Он основан на вычислении вероятности принадлежности к оттенку кожи каждой точки дискретизированного цветового пространства. Для достижения робастности алгоритма он обучался на базе, которая состояла из изображений кожи 6 различных оттенков.

Первым этапом процедуры обучения алгоритма являлся переход от цветового пространства RGB к цветовому пространству $K_1K_2K_3$, которое представляет собой куб в пространстве RGB, размеры которого в 5 раз меньше исходного. Подобный переход необходим по двум причинам: во-первых, для обучения алгоритма требуется набрать меньшую статистику, а во-вторых, уменьшение числа элементов способствует повышению быстроты действия алгоритма. К тому же человек не может различить пиксели, отстоящие друг от друга по яркости своих цветовых компонент меньше, чем на 5 единиц.

Далее для каждого набора была получена функция плотности распределения вероятности $W(k_1, k_2, k_3)$, описывающая статистическое распределение пикселей в цветовом пространстве $K_1K_2K_3$, по которой находился порог детектирования для данного оттенка кожи. С данными порогами сравниваются вероятности, полученные в процессе работы алгоритма. Поскольку пороги находились для изображений размером 150×150 пикселей, то первым этапом работы алгоритма является масштабирование изображения до указанных размеров.

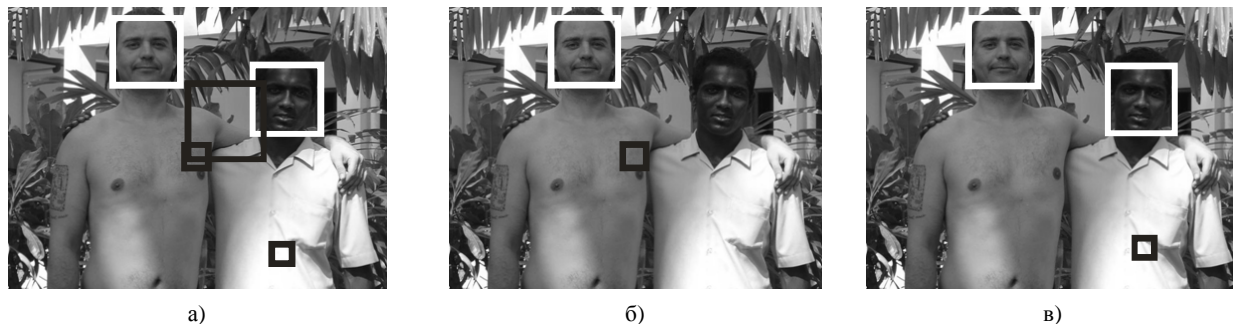


Рис. 1. Сравнение работы алгоритмов: а) исходный алгоритм; б) явное задание кластера кожи; в) непараметрическое задание кластера кожи

4. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

Тестирование работы алгоритма проводилось на базе полноцветных изображений, полученных из сети Интернет. При этом число лиц на изображениях, их размер и условия съемки различны и максимально приближены к реальным. Коэффициент масштабирования для алгоритма выделения лиц был взят равным 1,01. При использовании коэффициента масштабирования близкого к 1 достигается наибольший уровень выделения, но и происходит наибольшее число ложных срабатываний. Результаты работы алгоритмов сведены в табл. 1. В скобках указано количество выделенных областей.

Таблица 1

Тестируемый алгоритм	Уровень выделения	Уровень ложных срабатываний
Только яркость	92 % (217)	51 % (230)
Явное задание кластера кожи STDR = 0,4	85 % (201)	26%(72)
Непараметрическая модель кожи	88 % (209)	17%(44)

Из таблицы видно, что с помощью первого алгоритма удается почти в два раза уменьшить уровень ложных срабатываний. Однако при этом и уровень выделения снижается на 7%. Предлагаемый нами алгоритм показывает на 9% меньше ложных срабатываний при большем уровне детектирования (на 3%).

Среди тех ложных срабатываний, которые не удалось отсеять с помощью данных алгоритмов, часть содержит кожу и по определению не должна отфильтровываться. Результаты работы алгоритмов без учета таких ложных срабатываний представлены в табл. 2. Из данной таблицы видно, что «чистая» ошибка нашего алгоритма составляет 4%.

Таблица 2

Тестируемый алгоритм	Доля ложных срабатываний, которые не содержат кожу
Явное задание кластера кожи STDR = 0,4	13%(36)
Непараметрическая модель кожи	4%(10)

На рис. 1 приводится визуальный пример того, как модификации алгоритма уменьшают количество ложных срабатываний. Из рисунка видно, что оба алгоритма успешно отсеяли по два ложных срабатывания. Однако первый алгоритм также допустил ошибку на одном из лиц.

5. ЗАКЛЮЧЕНИЕ

Применение детектора кожи позволяет уменьшить количество ложных срабатываний на цветных изображениях при небольшом уменьшении уровня выделения. Алгоритм, использующий явное задание цвета кожи, ограничен одной областью цветовых оттенков. Предлагаемый алгоритм, использующий непараметрическую модель кластера кожи, является более гибким, и его работа определяется набором обучающих изображений.

6. ЛИТЕРАТУРА

- [1] Viola P., Jones M. Rapid object detection using a boosted cascade of simple features // Proc. Int. Conf. on Computer Vision and Pattern Recognition, 2001. № 1. P. 511-518.
- [2] Баринаова О.В., Ветров Д.П. Оценки обобщающей способности бустинга с вероятностными входами // Материалы 14-й Всероссийской конференции «Математические методы распознавания образов», Суздаль, 2009. С. 81-84.
- [3] Vezhnevets V., Sazonov V., Andreeva A. A survey on pixel-based skin color detection techniques // GRAPHICON03. 2003. P. 85-92.

Об авторах

Шмаглит Лев Александрович аспирант Ярославского государственного университета, кафедра динамики электронных систем.

Голубев Максим Николаевич аспирант Ярославского государственного университета, кафедра динамики электронных систем.

Адаптивный метод выделения движущихся объектов

Салимзибаров Рустам*, Разумков Артем**, Коробков Александр**

*Кафедра компьютерных систем и телекоммуникаций, Пермский Государственный Университет, Пермь, Россия

**ООО «Сателлит», Пермь, Россия

{rustamse, artem.razumkov, koravl1983}@graphicon.ru

Аннотация

Для систем видеоанализа и распознавания объектов в видеопотоках крайне важным является правильное выделение движущихся объектов. При работе систем видеонаблюдения в реальных условиях (при наличии шумов, теней, изменений освещенности и заднего плана, интенсивном движении) существующие методы выделения движущихся объектов часто работают недостаточно точно для последующего видеоанализа. В статье предложен алгоритм выделения движущихся объектов в видеопотоке с камеры видеонаблюдения в реальных условиях. Алгоритм основан на методе «вычитания фона»: строится модель переднего плана (движущийся объект) и заднего плана (статического фона). Уровень шума автоматически рассчитывается для разных частей изображения. Для определения уровня шума и заднего плана на каждом участке изображения применяются методы статистического анализа. Предложенный алгоритм обеспечивает выделение движущихся объектов, производит удаление теней и адаптируется к изменениям освещенности и изменениям заднего плана. Алгоритм был интегрирован в реальную систему охранного видеонаблюдения «MACROSCOP», что позволило производить более качественный видеоанализ.

Ключевые слова: Автоматический уровень шума, вычитание фона, выделение движущихся объектов.

1. ВВЕДЕНИЕ

Для систем видеоанализа и распознавания объектов в видеопотоках крайне важным является правильное выделение движущихся объектов. При работе систем видеонаблюдения в реальных условиях (при наличии шумов, теней, изменений освещенности и заднего плана, интенсивном движении) существующие методы выделения движущихся объектов часто работают недостаточно точно для последующего видеоанализа.

Большинство существующих методов выделения движущихся объектов [1] учитывают наличие шума, но не учитывают изменение уровня шума в пространстве и времени [2].

Многие существующие методы [1] адаптируются к резкому изменению освещенности или заднего плана, но после изменения (освещенности или заднего плана) эти методы продолжительное время работают некорректно, что приводит к неправильному выделению объектов и как следствие некорректным результатам видеоанализа.

К предложенному в настоящей статье методу предъявлялись следующие требования:

1. К предложенному в настоящей статье методу предъявлялись следующие требования: Устойчивая работа при наличии шума, автоматическое вычисление уровня шума для разных частей изображения;
2. Автоматическая адаптация к медленно меняющимся условиям освещенности; быстрая адаптация к резко меняющимся условиям освещенности;

3. Устойчивость к изменению заднего плана;
4. Автоматическое определение тени движущегося объекта и ее удаление;
5. Высокая скорость обработки данных (для параллельной обработки множества видеопотоков реального времени).

2. АЛГОРИТМ

Пусть задана видеопоследовательность кадров со стационарной камеры видеонаблюдения.

Для выделения областей движения используется модификация метода «вычитания фона»: строится передний план (движущиеся области) и задний план (статический фон). Для уменьшения влияния шума и увеличения быстродействия алгоритма изображение разбивается на блоки размером $S \times S$ пикселей, где S - параметр алгоритма. Дальнейший анализ изображения сводится к анализу блоков, где любая характеристика блока (например цвет) – среднее значение входящих в него пикселей.

2.1 Определение уровня шума и заднего плана

Уровень шума и задний план определяются для каждого блока в отдельности по схеме описанной ниже.

Чтобы обновить задний план и уровень шума для конкретного блока изображения находится такая непрерывная последовательность кадров, для которой с высокой вероятностью можно сделать вывод, что в рассматриваемом блоке изображение изменяется незначительно (изображение изменяется в заданных пределах).

Использовался следующий критерий незначительного изменения изображения в определенном блоке для непрерывной последовательности из K - кадров (условие 1):

$$\left\{ \begin{array}{l} \left| \mu_{i,j}^{НАЧ} - \mu_{i,j}^{КОН} \right| < \mu^C, \left| \mu_{i,j}^{НАЧ} - \mu_{i,j}^{СРЕД} \right| < \mu^C \\ \left| \sigma_{i,j}^{НАЧ} - \sigma_{i,j}^{КОН} \right| < \sigma^C, \left| \sigma_{i,j}^{НАЧ} - \sigma_{i,j}^{СРЕД} \right| < \sigma^C \end{array} \right. \quad (1)$$

, где $\mu_{i,j}^{НАЧ}$, $\mu_{i,j}^{КОН}$, $\mu_{i,j}^{СРЕД}$ - среднее значение яркости первой, второй половины последовательности и средней части последовательности соответственно для блока с координатами i, j ; $\sigma_{i,j}^{НАЧ}$, $\sigma_{i,j}^{КОН}$, $\sigma_{i,j}^{СРЕД}$ - аналогичные среднеквадратичные отклонения, где μ^C, σ^C - параметры алгоритма.

Если условие 1 выполняется, то считаем, что изображение изменяется незначительно, поэтому можно определить уровень шума и задний план.

Из вышесказанного следует, что могут существовать блоки с незадаанным уровнем шума и задним планом.

- а) Если условие (1) выполняется для некоторого блока, то задаем уровень шума для соответствующего блока

$$N_{i,j} = \text{Max}_{l=M-K+1}^{l=M} (I_{i,j}^l) - \text{Min}_{l=M-K+1}^{l=M} (I_{i,j}^l), \text{ задаем задний}$$

$$\text{план } B_{i,j} = \frac{1}{K} \sum_{l=M-K+1}^{l=M} I_{i,j}^l, \text{ где } I_{i,j}^l - \text{ яркость } l\text{-го кадра}$$

в видеопоследовательности для блока с координатами i, j , M – номер анализируемого кадра в видеопоследовательности, K – количество анализируемых кадров в непрерывной последовательности.

- б) Если условие (1) не выполняется, то уровень шума и задний план для данного блока оставляем неизменным.
- с) В случае если условие (1) не выполняется длительное время, а именно при обработке определенного количества подряд идущих кадров, считаем, что у блока не задан уровень шума и задний план.

2.2 Выделение движущихся объектов

Для выделения переднего плана анализируется каждый блок изображения. Критерий принадлежности блока к переднему плану (движущемуся объекту) зависит от того задан ли для конкретного блока уровень шума и задний план.

- а) Блоки с заданным уровнем шума и задним планом.

Для выделения движущихся объектов соответствующий блок заднего плана сравнивается с блоком в текущем кадре с учетом уровня шума.

Если $|I_{i,j}^M - B_{i,j}| > N_{i,j} \cdot Q$, то блок с координатами i, j принадлежит переднему плану, где Q – параметр алгоритма.

- б) Блоки с незадаанным уровнем шума и задним планом.

Если межкадровая разница велика, т.е. $|I_{i,j}^M - I_{i,j}^{M-1}| > R$, то блок с координатами i, j принадлежит переднему плану, где R – параметр алгоритма.

Далее блоки группируются в объекты, критерий группировки – пространственная близость.

Для исключения выделения неподвижных объектов из переднего плана (например, выделения остановившегося человека) анализируется межкадровая разница. В случае если ни для одного блока некоторого движущегося объекта разность яркостей на двух последних кадрах (межкадровая разность) не превышает определённого порога (R), объект считается неподвижным и исключается из списка движущихся объектов.

2.3 Удаление теней

Для правильного выделения движущихся объектов необходимо определить и исключить тени, отбрасываемые движущимися объектами. Для удаления теней использовалась методика удаления теней сходная с [3]. Данная методика основывается на анализе яркости и оттенка блоков. Падение тени в большинстве случаев приводит к значительному изменению (уменьшению) яркости при минимальном изменении оттенка. Также наша методика использует особенность теней, заключающуюся в том, что падение яркости в области тени относительно однородно.

3. ЭКСПЕРИМЕНТАЛЬНЫЕ РЕЗУЛЬТАТЫ

На рисунке 1.А представлен график скорости обучения алгоритма при $K=10$, где M – номер анализируемого кадра. Из графика видно, что 40% блоков имеют определенный уровень шума после анализа первого кадра (10-го кадра, т.к. первые $K=10$ кадров не анализируются), а 95% – через 50 кадров. На рисунке 1 Б, В, Г представлен результат работы алгоритма выделения движущихся объектов.

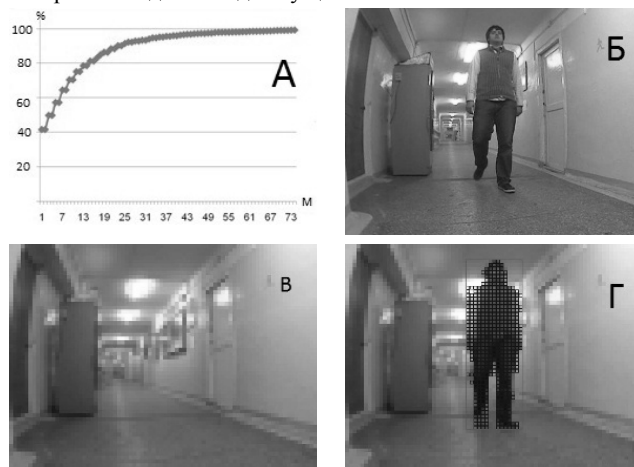


Рисунок 1: А) График скорости обучения алгоритма при $K = 10$ (зависимости количества блоков изображения с определенным значением уровня шума и задним планом от количества обработанных кадров). Б) Кадр с движущимся объектом В) Неподвижный задний план Г) Пример работы алгоритма выделения движущегося объекта.

4. ЗАКЛЮЧЕНИЕ

Предложенный алгоритм обеспечивает определение движущихся объектов, производит удаление теней и адаптируется к изменениям освещенности и изменениям заднего плана.

Предложенный алгоритм выделения движущихся объектов был реализован и протестирован на видеопоследовательностях с реальных камер видеонаблюдения, как на улице, так и в помещении. Алгоритм был интегрирован в реальную систему охранного видеонаблюдения «MACROSCOP», что позволило производить более качественный видеонализ.

5. СПИСОК ЛИТЕРАТУРЫ

- [1] M. Piccardi. *Background Subtraction Techniques: A Review IEEE SMC/ICSMC, vol. 4, pp. 3099–3104, 2004.*
- [2] П. Харевов, С. Новиков. *Проблемы выделения объектов в компрессированном потоке изображений. Proceedings of Graphicon, 2009.*
- [3] Xu, L., J.L. Landabaso, and M. Pardo. *Shadow Removal with Blob-Based Morphological Reconstruction for Error Correction. Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 2, 2005.*

Анализ использования адаптивного дискретного косинусного преобразования для улучшения качества изображений

Гущина О.Н., Аминова Е.А.

Физический факультет, кафедра Динамики электронных систем
Ярославский государственный университет им П.Г. Демидова, Ярославль, Россия

Аннотация

В рамках исследования была проанализирована возможность использования адаптивного дискретного косинусного преобразования в задачах удаления шума и подавления артефактов блочности. Рассмотренный алгоритм обработки изображений с использованием адаптивного дискретного косинусного преобразования был реализован в исследовательской среде MatLab и использован для улучшения качества сжатых и зашумленных изображений.

Ключевые слова: адаптивное дискретное косинусное преобразование, удаление блочности, фильтрация

1. ВВЕДЕНИЕ

Блочное двумерное дискретное косинус-преобразование (Б-ДКП) долгое время успешно использовалось в качестве ключевого элемента в задачах сжатия изображений и удаления шума. Однако известно, что применение Б-ДКП нередко приводит к возникновению артефактов блочности, выражающихся в том, что изображение распадается на визуально заметные квадратные области размером 8×8 пикселей.

В последние годы особое внимание уделяется разработке методов анализа изображений и видеопоследовательностей с помощью окон произвольной формы. Оптимальным по вычислительным затратам является адаптивное дискретное косинусное преобразование (А-ДКП), предложенное Сикорой [1].

2. ИСПОЛЬЗОВАНИЕ А-ДКП В ЗАДАЧАХ ОБРАБОТКИ ИЗОБРАЖЕНИЙ

А-ДКП вычисляется путем применения одномерного дискретного косинусного преобразования переменной длины последовательно к столбцам и строкам исследуемой области изображения. А-ДКП не требует каких-либо дополнительных преобразований над полученной матрицей коэффициентов и может рассматриваться как обобщение классического Б-ДКП. Использование А-ДКП совместно с анизотропной локальной полиномиальной аппроксимацией позволяет адаптировать форму окна при обработке каждого пикселя изображения [2]. Примеры формирования окна произвольной формы представлены на рис. 1 [4].

2.1. Удаление шума из изображений

Рассмотрим задачу удаления аддитивного гауссова шума из изображений:

$$z(x) = y(x) + \eta(x), \quad (1)$$

где z - зашумленное изображение, y - исходное изображение, η - матрица независимых отсчетов нормального гауссова распределения с нулевым средним и дисперсией σ^2 . А-ДКП и обратное А-ДКП обозначим как $T(y)$ и $T^{-1}(y)$ соответственно.

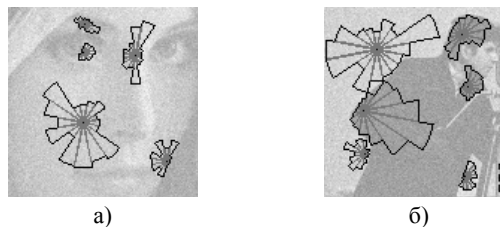


Рис. 1. Примеры формирования окна произвольной формы на тестовых изображениях «Лена» (а) и «Фотограф» (б)

Для каждого пикселя изображения с координатами x посредством ЛПА вычисляется окрестность U_x . Затем производится вычисление А-ДКП с последующей жесткой пороговой фильтрацией γ с порогом $\sigma \sqrt{2 \ln(N^2)}$, где N – спектральная плотность мощности шума [3]. Использование обратного А-ДКП позволяет получить восстановленное изображение.

Общая схема алгоритма фильтрации представлена на рис. 2.

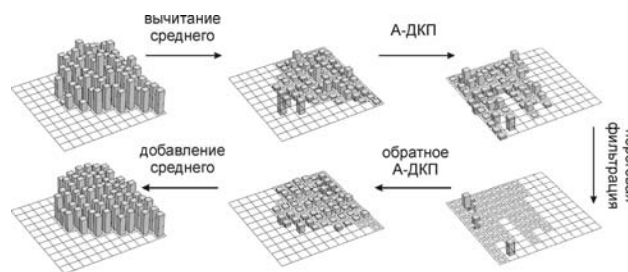


Рис. 2. Алгоритм фильтрации на основе А-ДКП

В математической форме эту операцию можно представить следующим образом [4]:

$$\tilde{y}_U = T_U^{-1}(\gamma(T_U(z_U - m_U(z)))) + m_U(z), \quad (2)$$

где $m_U(z)$ - среднее значение яркости внутри области U_x .

В дополнение к этому для каждого пикселя на основе коэффициентов преобразования вычисляется вес для последующего восстановления:

$$w_x = \frac{\sigma^{-2}}{(1 + N_x^{\text{har}})|U_x|}, \quad (3)$$

где N_x^{har} - число ненулевых коэффициентов после пороговой фильтрации, а $|U_x|$ - число пикселей в области U_x .

Процедуру восстановления можно представить следующим выражением:

$$\tilde{y} = \frac{\sum_{x \in X} w_x \tilde{y}_U}{\sum_{x \in X} w_x \chi_U}, \quad (4)$$

где X - множество всех координат пикселей изображения, а χ_U - функция, показывающая принадлежность пикселя к области U_x .

2.2. Подавление артефактов блочности

Рассмотрим сжатое изображение $z(x)$, заданное выражением (1). Для того чтобы применить А-ДКП, нам необходимо располагать значением дисперсии σ^2 , которое может быть получено из матрицы квантования коэффициентов $Q = [q_{i,j}]$ с использованием эмпирической формулы:

$$\sigma^2 = 0.69(\tilde{q})^{1.3}, \quad (5)$$

где $\tilde{q} = \frac{1}{9} \sum_{i,j=1}^3 q_{i,j}$ - среднее значение девяти

коэффициентов матрицы квантования, соответствующих гармоникам ДКП наиболее низких частот [4].

Процедура нахождения дисперсии σ^2 , определенная выражением (5), может быть применена непосредственно, т.к. матрицы квантования коэффициентов ДКП являются известными и могут быть найдены в соответствующих спецификациях для стандартов сжатия. Поэтому процедура восстановления изображения при помощи А-ДКП, может быть использована в задачах удаления артефактов блочности из изображений.

3. АНАЛИЗ РАБОТЫ АЛГОРИТМА С А-ДКП ДЛЯ ВОССТАНОВЛЕНИЯ ИЗОБРАЖЕНИЙ

Рассмотренный в работе алгоритм обработки изображений был реализован в среде MatLab. Для проведения исследований использовался набор тестовых изображений размером 512×512 пикселей. Работа алгоритма фильтрации оценивалась посредством метрики пиковое отношение сигнал/шум (ПОСШ).

Для проведения исследования к выбранным тестовым изображениям добавлялся аддитивный белый гауссов шум с различными значениями дисперсии σ^2 . Результаты, полученные при тестировании алгоритмов фильтрации, для тестового изображения «Лена» приведены на рис. 3.

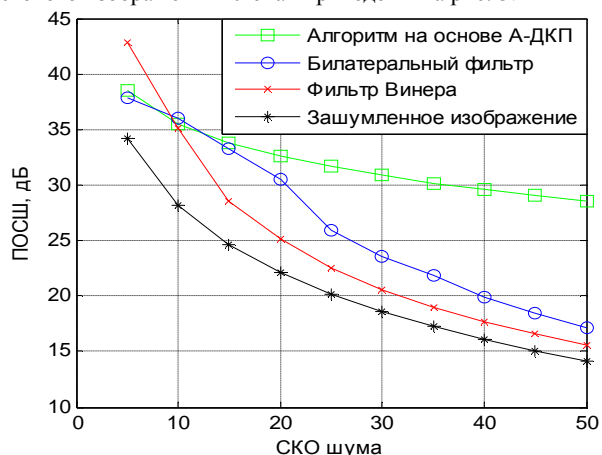


Рис. 3. Результаты восстановления тестового изображения «Лена» по шкале ПОСШ

Таким образом, алгоритм на основе А-ДКП позволяет значительно улучшить качество зашумленного изображения и повысить значение ПОСШ на величину порядка 10-14 дБ

практически при любых значениях дисперсии шума σ^2 . Сравнительный анализ результатов восстановления различных тестовых изображений показывает превосходство рассмотренного алгоритма на основе А-ДКП по сравнению с известными алгоритмами фильтрации (фильтр Винера, билатеральный фильтр и др.).

Также был проведен анализ работы алгоритма на основе А-ДКП в задаче удаления артефактов блочности из изображений, сжатых при помощи стандарта JPEG. Степень сжатия изображения оценивалась при помощи величины бит на пиксель – bpp (bits per pixel). Результаты восстановления тестового изображения «Лена» при различных значениях bpp приведены в табл. 1.

Таблица 1
ПОСШ восстановленного изображения «Лена», дБ

bpp	0,2	0,4	0,6	0,8	1
Сжатое изображение	28,89	33,57	35,49	36,85	37,96
Обработанное изображение	30,40	34,44	36,09	37,27	38,24

Применение алгоритма позволяет увеличить значение ПОСШ на величину до 1,5 дБ. Эффект размытия, вносимый алгоритмом в обрабатываемое изображение, визуально менее заметен, нежели блочная структура JPEG- изображения.

4. ЗАКЛЮЧЕНИЕ

По результатам проведенных исследований можно сделать вывод о том, что рассмотренный алгоритм на основе А-ДКП позволяет достичь хороших результатов в задаче восстановления изображений, что подтверждается как объективными метриками качества, так и субъективным визуальным анализом. Данный алгоритм может быть использован для удаления шума и подавления артефактов блочности на изображениях, как на этапе предварительной обработки, так и для постобработки.

5. ЛИТЕРАТУРА

1. Sikora T. Low complexity shape-adaptive DCT for coding of arbitrarily shaped image segments // Signal Process.: Image Comm., vol. 7, pp. 381-395, 1995.
2. Fan J., Gijbels I. Local polynomial modelling and its application // Chapman and Hall, London, 1996.
3. Donoho D.L., Johnstone I.M. Ideal spatial adaptation via wavelet shrinkage // Biometrika, n. 81, pp. 425-455, 1994.
4. Foi, A., Dabov K., Katkovnik V., Egiazarian K. Shape-adaptive DCT for denoising and image reconstruction // Proc. SPIE El. Imaging 2006, Image Process.: Algorithms and Systems V, 6064A-18, Jan. 2006.

ИНФОРМАЦИЯ ОБ АВТОРАХ

Гущина Ольга Николаевна – студентка физического факультета (направление Телекоммуникации) Ярославского государственного университета (ЯрГУ) им П.Г.Демидова. Контактный email – olyagushchina@rambler.ru.

Аминова Елена Альбертовна – студентка физического факультета (специальность Радиофизика и электроника) ЯрГУ им П.Г.Демидова. Контактный email – lena@piclab.ru.

Восстановление Информации о Смежности Вершин для Полигональных Сеток, Полученных с Помощью Метода Марширующих Кубов

Ростислав Хлебников
Institute for Computer Graphics and Vision
Graz University of Technology, Graz, Austria
khlebnikov@icg.tugraz.at

Аннотация

Topological information is crucial for many surface mesh processing algorithms. This information can be both desirable, e.g. for optimizing mesh rendering, and required, e.g. for surface curvature computation.

In this paper we propose a high-performance and robust algorithm for extracting mesh connectivity information for isosurfaces of 3D data that have been extracted using marching cubes algorithm.

Keywords: *Mesh topology, Marching Cubes, isosurface.*

1. ВВЕДЕНИЕ

Несмотря на существенный рост доступных вычислительных мощностей, извлечение изоповерхностей из пространственных данных по-прежнему требует значительных временных затрат. Кроме этого, увеличивается и объем данных, поступающих на вход алгоритмов. Так, современный компьютерный томографический сканер, способен создать объемное изображение размером 512x512x320 вокселей всего лишь за одно вращение [1]. Поэтому для извлечения изоповерхностей используются вычислительные возможности графических ускорителей, которые обеспечивают высокую степень параллелизации обработки данных. Так как метод марширующих кубов [2] является одним из наиболее распространенных методов построения изоповерхностей в силу его высокого быстродействия и простоты реализации, на данный момент уже описаны способы использования графических адаптеров для его ускорения [3].

Недостатком такого подхода является то, что полученная изоповерхность описывается лишь с помощью набора треугольников. Однако, для работы многих алгоритмов необходимо наличие топологической информации, как, например, для расчета кривизны [4], для сглаживания поверхности [5] или для автоматического расчета уровней детализации [6]. Таким образом, встает задача быстрого и надежного восстановления информации о топологии поверхностей, описанных с помощью набора треугольников.

Описанный в последующих секциях алгоритм использует небольшую модификацию алгоритма марширующих кубов для извлечения изоповерхности из пространственной регулярной решетки и сохраняет информацию о смежности вершин этой изоповерхности в компактной структуре данных. Каждая секция начинается с краткого обзора литературы в соответствующей области, за которым следует описание наших результатов.

2. СТРУКТУРА ДАННЫХ

На текущий момент предложено множество структур данных, позволяющих в компактном виде представить информацию о

топологии полигональной сетки. Многие из них являются модификациями представления с помощью «крылатых» ребер (winged edge data structure, [7]). Так, например, представление с помощью радиальных ребер [8] пригодно также для описания поверхностей, не являющихся многообразиями. Кроме того весьма распространено описание полигональных сеток с помощью полу-ребер [9], которое реализовано в таких библиотеках, как CGAL [10] и OpenMesh [11].

Так как нам необходима информация лишь о смежности вершин, мы предлагаем структуру данных, основанную на представлении с помощью радиальных ребер, но с исключением излишней информации (см. Рис. 1).

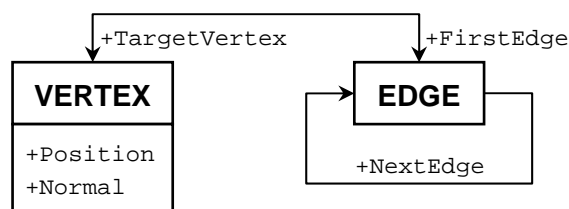


Рисунок 1: Диаграмма классов структуры данных для определения смежных вершин.

3. АЛГОРИТМ

При восстановлении топологической информации о полигональной сетке, описанной с помощью набора треугольников, одной из основных задач является определение того, какие из вершины, принадлежащие различным треугольникам, совпадают. Эта задача осложняется тем, что при вычислении координат вершины в алгоритме, создающем изначальное описание полигональной сетки, возможны погрешности вычислений, обусловленные использованием чисел с плавающей запятой. В работе [12], Рок и Вожный используют сбалансированное дерево двоичного поиска (а именно, AVL-дерево [13]) для нахождения и слияния пространственно близких вершин. В работе [14], МакМэйнс и др. используют хэш-таблицы для поиска одинаковых вершин. Это позволяет достигать более высокой производительности, однако, в силу особенностей хэш-функции, слияние вершин возможно только при одинаковом бинарном представлении координат этих вершин.

В качестве входных данных, оба вышеописанных метода используют файлы в формате STL. Эти файлы могут быть получены с помощью множества прикладных программ и поэтому, при создании алгоритмов восстановления топологической информации, невозможно полагаться на какие-либо дополнительные предположения о свойствах

входных данных. Однако, так как мы рассматриваем полигональные секты, созданные с помощью метода марширующих кубов, мы можем объединить положительные стороны алгоритмов, описанных выше.

Как и в [14], мы используем хэш таблицу для слияния вершин. Для того, чтобы избавиться от влияния численной погрешности расчетов при построении хэш-ключа, для каждой вершины, создаваемой при построении изоповерхности, мы записываем индекс ребра регулярной решетки, на котором создается эта вершина или индекс узла в случае если изоповерхность проходит непосредственно через узел решетки. Так как алгоритм марширующих кубов [2] создает не более одной вершины на каждом ребре решетки или проходит через ее узлы, этот индекс (а значит и значение хэш-функции) будет совпадать только у вершин, которые должны быть слиты. Точные координаты таких вершин, тем не менее, могут отличаться из-за погрешностей вычислений при обработке различных ячеек, имеющих общее ребро.

Для регулярных сеток размером до 1024^3 , 32-битный индекс может быть построен, например, следующим образом:

$$Index = i + n_x \cdot (j + n_y \cdot (k + dirIndex \cdot n_z)),$$

где (i, j, k) – индекс узла, откуда исходит ребро, (n_x, n_y, n_z) – количество узлов регулярной сетки по каждой из координатных осей, а $dirIndex$ равен 0, если изоповерхность проходит через этот узел, 1 для ребер, ориентированных по оси X, 2 – для оси Y и 3 – для оси Z.

В остальном, наш алгоритм следует схеме восстановления топологической информации в оперативной памяти, описанной в [14].

4. РЕЗУЛЬТАТЫ И ВЫВОДЫ

Мы реализовали предложенный алгоритм на языке C++. В качестве реализации алгоритма марширующих кубов мы использовали реализацию, распространяемую вместе с nVidia CUDA SDK [15]. В качестве хэш-функции мы использовали MurmurHash2A [16].

Мы сравнили результаты работы нашего алгоритма с алгоритмами, описанными в [12] и [14]. Результаты сравнения представлены в Таблице 1.

Количество вершин (тыс.)	Предложенный алгоритм	Алгоритм [14]	Алгоритм [12]
10	0.86	0.73	1.10
100	1.62	1.64	2.74
1000	4.05	4.02	6.52

Таблица 1: Сравнение времени работы алгоритмов восстановления топологической информации (в секундах). Каждое время работы получено усреднением за 50 запусков. Алгоритм [14] оставляет неслитыми в среднем от 0.5% до 1.6% от общего числа вершин. Предложенный нами алгоритм и [12] создают абсолютно идентичные полигональные сетки.

5. ЗАКЛЮЧЕНИЕ

В данной статье мы представили алгоритм, позволяющий надежное и быстрое восстановление топологической информации для полигональных сеток, полученных с помощью метода марширующих кубов, а также описали

компактную структуру данных, позволяющую извлекать информацию о смежных вершинах в полигональной сетке.

Благодарности

Данная работа была спонсирована Европейским Союзом под инициативой FP7 VPH, контракт номер 223877.

6. СПИСОК ЛИТЕРАТУРЫ

- [1] L.K. Anderson, "What's Next for CT," May. 2007.
- [2] W.E. Lorensen and H.E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *Proceedings of the 14th annual conference on Computer graphics and interactive techniques - SIGGRAPH '87*, 1987, pp. 163-169.
- [3] C. Dyken, G. Ziegler, C. Theobalt, and H. Seidel, "High-speed Marching Cubes using HistoPyramids," *Computer Graphics Forum*, vol. 27, 2008, pp. 2028-2039.
- [4] Zhong Li, B. Barsky, and Xiaogang Jin, "An effective third-order local fitting patch and its application," *2009 IEEE International Conference on Shape Modeling and Applications*, Beijing, China: 2009, pp. 7-14.
- [5] Z. Lv and X. Chen, "A Three Dimensional Mesh Improvement Algorithm Based on Curvature Flow," *2009 Second International Symposium on Information Science and Engineering*, Shanghai, China: 2009, pp. 36-40.
- [6] M. Hussain, Y. Okada, and K. Nijjima, "LOD modelling of polygonal models based on multiple choice optimisation," *10th International Multimedia Modelling Conference, 2004. Proceedings.*, Brisbane, Qld., Australia: , pp. 203-210.
- [7] B.G. Baumgart, "A polyhedron representation for computer vision," Anaheim, California: 1975, p. 589.
- [8] K. Weiler, "The Radial Edge Structure: A Topological Representation for Non-Manifold Geometric Boundary Modeling," *Geometric Modeling for CAD Applications*, Amsterdam: North-Holland, 1988, pp. 3-36.
- [9] M. Mäntylä, *An introduction to solid modeling*, Rockville Md.: Computer Science Press, 1988.
- [10] L. Kettner, "Halfedge Data Structures," *CGAL User and Reference Manual*, CGAL Editorial Board, 2010.
- [11] M. Botsch, S. Steinberg, S. Bischoff, and L. Kobbelt, "OpenMesh - a generic and efficient polygon mesh data structure," 2002.
- [12] S.J. Rock and M.J. Wozny, "Generating Topological Information from a "Bucket of Facets"," 1992, pp. 251–259.
- [13] Г.М. Адельсон-Вельский и Е.М. Ландис, "Один алгоритм организации информации," *Доклады АН СССР*, Т. 146, №2, 1962, стр. 263–266.
- [14] S. McMains, J.M. Hellerstein, and C.H. Séquin, "Out-of-core build of a topological data structure from polygon soup," Ann Arbor, Michigan, United States: 2001, pp. 171-182.
- [15] "NVIDIA CUDA SDK - Physically-Based Simulation," http://www.nvidia.com/content/cudazone/cuda_sdk/Physically-Based_Simulation.html, 2008.
- [16] A. Appleby, "Murmur hash," <http://sites.google.com/site/murmurhash/>, 2009.

Сегментация отсканированных документов

Алексей Вилькин, Марта Егорова
 НИЯУ «МИФИ», Москва, Россия
 { aleksey.vilkin, marta.egorova } @gmail.com

Аннотация

Рассматривается восходящий подход сегментации отсканированных документов на области фона, текста и фотографий. На первом этапе изображение разбивается на блоки. Для каждого блока считается ряд текстурных характеристик. На основе этих характеристик построены два AdaBoost комитетов классификаторов. Они объединяются в дерево решений, которое определяет тип блока. На втором этапе типы блоков итеративно корректируются на основе анализа соседних областей. На тестовой выборке достигнута точность сегментации более 80%.

Ключевые слова: сегментация отсканированных документов, текстурные признаки

1. ВВЕДЕНИЕ

Большая часть информации в современном мире хранится в электронном виде. Многие печатные документы также переводятся в электронный вид с помощью систем OCR (*optical character recognition*), которые требуют сегментации областей. Также анализ областей документа требуется при сохранении сканированных документов в PDF с MRC (*mixed raster content*) сжатием. При MRC сжатии для фона, текста, рисунков создаются собственные маски, и каждый тип информации сжимается с помощью собственного алгоритма сжатия.

Существует довольно много исследований по данной теме, в литературе можно выделить два основных подхода: восходящий (*bottom-up*) и нисходящий (*top-down*) подходы. В *bottom-up* [1,2,3] алгоритмах анализ начинается с объектов низкого уровня, таких как пиксели, зоны, соседние области; затем полученные объекты соединяются и классифицируются, как области документа. Такие алгоритмы хорошо обрабатывают области со сложной формой, но не учитывают при анализе объекты высокого уровня. Напротив, *top-down* алгоритмы [4] начинают со всего изображения в целом и пытаются разделить его на области конкретного типа. Такие алгоритмы не всегда способны обработать регионы сложной формы, например, прямоугольные блоки текста или заголовки на несколько колонок текста. Многие подходы хорошо справляются со своей задачей для ограниченного набора изображений, так как для анализа используют специфичные признаки или предположения.

Мы предлагаем восходящий подход для работы с полутонными 8 bpp отсканированными изображениями документов, который на первом этапе классифицирует блоки, используя текстурные признаки, а затем итеративно уточняет класс блока за счет учета класса соседних блоков.

2. ОПИСАНИЕ АЛГОРИТМА

2.1 Количественные характеристики текстур

Изображение I первоначально разделяется на блоки размера $N \times N$ пикселей, где значение N зависит от разрешения отсканированного документа. Были рассмотрены размеры окна в интервале 10-80 пикселей и два варианта пересекающихся блоков и расположение блоков «плиткой». Размер окна должен быть достаточно маленьким, чтобы отделять различные области (текст, рисунок, фон) друг от друга, но и достаточно большим, для того чтобы полученные характеристики были адекватными.

В процессе экспериментов исследовалось более 20 количественных признаков текстур [1,3,5] среди которых при помощи AdaBoost toolbox [6] были выделены наиболее информативные:

$$\text{- средняя яркость блока } B_i: \bar{B}_i = \frac{\sum_{r=1}^N \sum_{c=1}^N B_i(r, c)}{N^2} \quad (1)$$

$$\text{- средняя разница средних яркостей блоков } B_k \text{ в 4-связной окрестности блока } B_i: \overline{dB}_i = \frac{\sum_{k=1}^4 |\bar{B}_i - \bar{B}_k|}{4} \quad (2)$$

$$\text{- среднее вертикальной } dB_y^i \text{ и горизонтальной } dB_x^i \text{ производной блока: } d_{x,y} B_i = \frac{\sum_{r=1}^N \sum_{c=1}^{N-1} dB_x^i(r, c) + \sum_{r=1}^{N-1} \sum_{c=1}^N dB_y^i(r, c)}{2N(N-1)} \quad (3)$$

$$\text{- однородность блока } B_i: H = \sum_{i,j} \frac{N_d(i, j)}{1 + |i - j|}, \quad (4)$$

где N_d нормированная матрица совместной встречаемости (*co-occurrence matrix*), d задает пространственное отношение.

- процент пикселей с градиентом больше порога:

$$P_g = \sum_{\forall (r,c) \in B_i} \{1 | \nabla B_i(r, c) > T\} / N^2, \text{ где } \nabla B_i(r, c)$$

вычислется как квадратный корень из суммы квадратов горизонтальной и вертикальной производных. (5)

- процент изменений значений пикселей после морфологической операции открытия B_i^o над бинарным изображением B_i^b , полученным бинаризацией по порогу 128:

$$P_m = \sum_{\forall (r,c) \in B_i} \{1 | B_i^o(r, c) \neq B_i^b(r, c)\} / N^2. \quad (6)$$

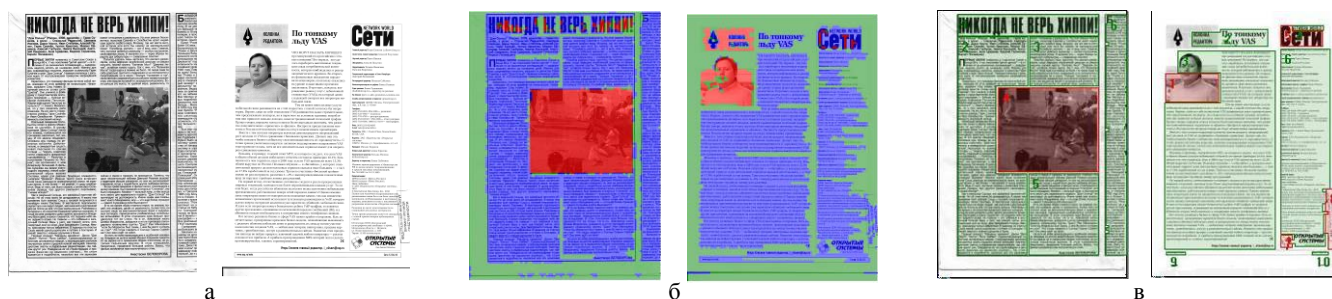


Рисунок 1 Результаты сегментации документов: а) исходные документы; б) сегментация предложенным методом, в) сегментация программой FineReader 9.0

2.2 Классификация блоков

Количественные характеристики текстов позволяют отнести блок к одному из трех типов: рисунок, фон, текст. Для этого построено два AdaBoost классификатора: один для определения фотографий C_i , второй - для текста T_i . На рисунке 2 показано дерево решений, объединяющее оба классификатора для разделения на 3 класса. Каждый классификатор обучен на выборке из 57 документов разных типов, из разных источников. Если оба классификатора C_i , T_i дают положительный результат, то выбирается тот, что по модулю дальше от соответствующего порога T_t , T_i . Если оба отрицательные, то блок классифицируется как фон.

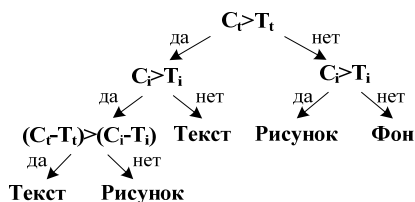


Рисунок 2 Дерево решений для классификации блока

Пороги полагаются равными сумме весов всех слабых классификаторов. Изменяя значение порога можно влиять на количество ошибок первого и второго рода.

2.3 Уточнение класса блоков

В результате классификации иногда возникают ошибки: отдельные блоки классифицируются неверно. Для получения однородных областей и подавления шумов для блока, класс которого отличается от класса соседних блоков мы используем простую, но эффективную процедуру: определяем количество соседей каждого класса, и если число соседей какого-то класса больше 6, то блок реклассифицируется. Данная итерация выполняется несколько раз. Чем меньше окно, тем более разнородным, зашумленным получается результат.

3. РЕЗУЛЬТАТЫ

Работа алгоритма проверена на тестовой выборке, состоящей из 174 отсканированных изображений газет, документов, журналов разных типов. Результат классификации пикселей изображений представлен в таблице 1 в виде матрицы ошибок (*confusion matrix*) [7] усредненной по всем изображениям. Истинные значения посчитаны по вручную размеченным маскам. Для данного критерия наилучшие результаты с учетом производительности при разрешении

сканирования 300 dpi были получены для размера окна в 30 пикселей и расположения блоков «плиткой».

Таблица 1 Относительная матрица ошибок, усредненная по всем изображениям тестовой выборки

Классифицируемые	Истинные		
	Фон	Текст	Рисунок
Фон	0.37	0.07	0.01
Текст	0.04	0.33	0.02
Фото	0.01	0.03	0.12

Предложенный алгоритм правильно классифицировал 82% пикселей. FineReader 9.0 правильно классифицировал 89% пикселей. Таким образом алгоритм даёт результаты, сравнимые с коммерческим ПО. Далее мы планируем улучшить алгоритм, в первую очередь, исследовать классификацию блоков с использованием SVM и, k-NN, а также применить более сложный анализ классифицированных блоков.

4. СПИСОК ЛИТЕРАТУРЫ

- [1] J.J.Sauvola, M.Pietikäinen, *Page segmentation and classification using fast feature extraction and connectivity analysis. Int. Conf. on Document Analysis and Recognition, 1995*
- [2] F. Wahl, K. Wong, R. Casey, *Block segmentation and text extraction in mixed text/image documents, Computer Graphics and Image Processing, 1982, Vol.20, pp. 375-390.*
- [3] H. S. Baird, M. A. Moll, Chang An, *Document Image Content Inventories. Proc. of SPIE/IS&T Document Recognition & Retrieval, 2007.*
- [4] F. Cesarini, S. Marinai, G. Soda, M. Gori, *Structured Document Segmentation and Representation by the Modified X-Y tree. International Conference on Document Analysis and Recognition, 1999, pp. 563.*
- [5] Шапиро Л., Стокман Дж., *Компьютерное зрение, Бином. Лаборатория знаний, 2009.*
- [6] <http://graphics.cs.msu.ru/science/research/machinelearning/adaboosttoolbox>
- [7] http://en.wikipedia.org/wiki/Confusion_matrix

Об авторах

Вилькин Алексей - студент 5 курса факультета Кибернетики НИЯУ «МИФИ». Его областью интереса являются распознавание образов, обработка изображений.

Марта Егорова – аспирантка НИЯУ «МИФИ». Ее областью интереса являются оценка и улучшение качества изображений, алгоритмы компьютерного зрения.

Удаление аддитивного белого гауссова шума из цифровых изображений на основе анализа главных компонент

Владимир Волохов, Евгений Сергеев
Лаборатория цифровые цепи и сигналы

Ярославский государственный университет им. П.Г. Демидова, Ярославль, Россия
volokhov@piclab.ru, dcslab@uniyar.ac.ru

Аннотация

В настоящей работе рассматривается метод восстановления цифровых изображений из зашумленных данных, основанный на анализе главных компонент. Приведены результаты моделирования, показывающие основные особенности данного метода преобразования. Проведено сравнение полученных результатов с результатами, синтезированными на основе современных алгоритмов фильтрации.

Ключевые слова: анализ главных компонент, область обучения, векторы обучения, сохранение границ, фильтрация цифровых изображений.

1. ВВЕДЕНИЕ

В настоящее время существует множество различных алгоритмов фильтрации цифровых сигналов, изображений и видеопоследовательностей. В зависимости от поставленной задачи, результаты работы современных алгоритмов фильтрации данных (далее цифровых изображений) могут отличаться. Предполагая, что данные являются искаженными аддитивным белым гауссовым шумом, выделим несколько стандартных подходов к решению вышеозначенной задачи. Во-первых, отметим методы основанные на «классической» локальной обработке в пространственной области, вейвлет-, курвлет-анализ [1] и их аналоги, которые, используя некоторые априорные знания о анализируемых данных, позволяют достаточно эффективно (с точки зрения вычислительной сложности и в определенных ситуациях с точки зрения качества восстановления) обрабатывать рассматриваемые данные. Во-вторых, отметим методы, основанные на анализе главных компонент (рассматриваемые в настоящей работе) [2], билатеральной фильтрации и ее аналогах [3], нелокальной обработке [4, 5], которые позволяют за счет адаптации к рассматриваемым данным производить их эффективную обработку с точки зрения качества восстановления, но обладают высокой вычислительной сложностью.

2. ОПИСАНИЕ ПРЕДЛАГАЕМОГО АЛГОРИТМА

Предполагая, что анализируемое цифровое изображение \mathbf{x} искажено аддитивным белым гауссовым шумом \mathbf{n} с нулевым математическим ожиданием и дисперсией σ^2 , перечислим основные шаги работы рассматриваемого алгоритма фильтрации.

1. Оцениваем дисперсию шума σ^2 на входном зашумленном изображении $\mathbf{y} = \mathbf{x} + \mathbf{n}$. Для этого можно воспользоваться достаточно распространенной формулой из теории М-оценок:

$$\hat{\sigma}^2 = \frac{\text{Медиана}(ДВК_1)}{0,6745}, \quad (1)$$

приведенной, например, в [2]. Здесь $ДВК_1$ – диагональные вейвлет-коэффициенты первого уровня вейвлет-разложения.

2. Разбиваем входное зашумленное изображение на совокупность перекрывающихся блоков, как приведено в [2]. Внутри каждого из таких блоков можно выделить: область обучения, область фильтрации и область наложения блоков. Размеры рассматриваемых областей могут варьироваться.

3. Внутри области обучения выбираем всевозможные блоки размера $N \times N$ (векторы обучения). Последние, будучи представленными в виде векторов-столбцов длиной N^2 каждый, позволяют сформировать некоторую матрицу S_y^1 размера $N^2 \times M$, столбцами которой и являются рассматриваемые векторы-столбцы. Здесь M – число векторов обучения, найденных в области обучения, а римские цифры I или II обозначают этап обработки изображения.

4. На основе матрицы S_y^1 составляем ковариационную матрицу Q_y^1 . Для матрицы Q_y^1 находим собственные числа и соответствующие им собственные векторы (главные компоненты данных, заключенных в матрице S_y^1 , которые образуют оптимальный, в смысле среднеквадратической ошибки, базис представления данных).

5. Для всех $l = 1, \dots, N^2$ и $i = 1, \dots, M$ находим проекции $y_{li}^1 = x_{li}^1 + n_{li}^1$ множества векторов, заключенных в матрице S_y^1 , на множество собственных векторов, найденных на предыдущем шаге [2]. Здесь y_{li}^1 (l -я проекция вектора i из матрицы S_y^1 на множество собственных векторов матрицы Q_y^1) представляет сумму l -ой проекция вектора i неискаженных данных и l -ой проекция вектора i шума.

6. Осуществляем обработку (фильтрацию) полученного множества проекций с использованием линейной среднеквадратической оценки:

$$\hat{x}_{li}^I = \frac{\sigma_l^2}{\sigma_l^2 + \sigma^2} \cdot y_{li}^I, \quad (2)$$

представленной в [2]. Здесь σ^2 – дисперсия шума, а σ_l^2 – дисперсия l -ой проекции векторов $i=1, \dots, M$ неискаженных данных, которую можно найти с использованием оценки максимального правдоподобия [2]:

$$\hat{\sigma}_l^2 = \max \left[0, \frac{1}{M} \sum_{i=1}^M (y_{li}^I)^2 - \sigma^2 \right]. \quad (3)$$

7. На основе множества обработанных данных \hat{x}_{li}^I восстанавливаем матрицу S_y^I , а на основе последней отдельную обработанную область на изображении. Повторяя аналогичную операцию для остальных областей фильтрации с учетом их наложения можно обработать зашумленное изображение целиком [2] и получить первую, «грубую» оценку неискаженного изображения.

8. Используя зашумленное изображение, повторяем шаги 2-5, рассмотренные выше. При этом устанавливаются другие размеры областей обучения, областей фильтрации и областей наложения, а так же размеры векторов обучения.

9. Выражение (2) заменяем следующим соотношением:

$$\hat{x}_{li}^{\Pi} = \frac{|z_{li}^{\Pi}|^2}{|z_{li}^{\Pi}|^2 + \sigma^2} \cdot y_{li}^{\Pi}, \quad (4)$$

где y_{li}^{Π} и z_{li}^{Π} – l -е проекции вектора i из матриц S_y^{Π} и S_z^{Π} на множество собственных векторов матриц Q_y^{Π} и Q_z^{Π} для зашумленных данных и данных, полученных на основе «грубой» оценки сформированной на шаге 7, соответственно.

10. Повторяя рассуждения, изложенные на шаге 7, получаем вторую, «точную» оценку исходного изображения.

3. РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ

Рассматриваемый в настоящей работе алгоритм был реализован с использованием пакета Matlab. Визуальные и численные (пиковое отношение сигнала к шуму – ПОСШ) результаты моделирования кратко приведены на рис. 1, на примере классического тестового изображения «Лена» размера 512×512 пикселей. Как видно из приведенных фрагментов восстановленных изображений, предложенный алгоритм достаточно неплохо справился с поставленной задачей, хорошо сохранив локальные особенности (криволинейные контура, границы и т.п.) анализируемого изображения.

4. ЗАКЛЮЧЕНИЕ

На основе проведенных исследований можно заключить, что рассматриваемый алгоритм позволяет достичь достаточно хороших результатов при решении задачи восстановления цифровых изображений из зашумленных данных.

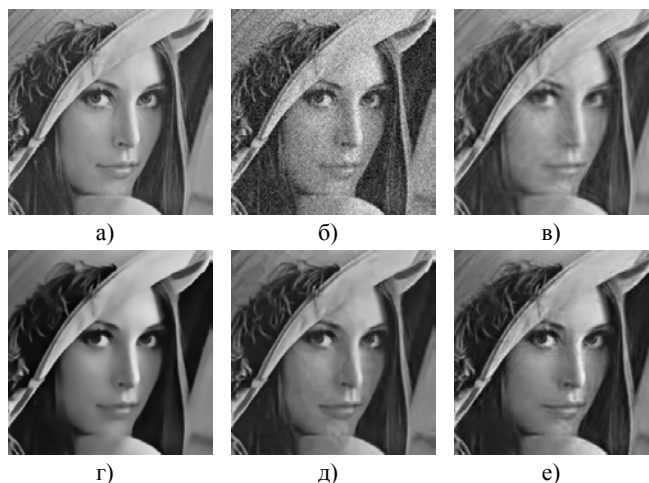


Рис. 1. Фрагменты изображения «Лена»: а) исходного; б) зашумленного, ПОСШ=20,15 дБ; в) метод, рассматриваемый в [1], ПОСШ=30,39 дБ; г) метод, рассматриваемый в [4], ПОСШ=32,55 дБ; д) метод, рассматриваемый в [5], ПОСШ=32,06 дБ; е) предложенный метод, ПОСШ=31,54 дБ

К достоинствам алгоритма можно отнести возможность сохранения локальных особенностей изображений, а так же адаптивность к анализируемым данным и использование двухэтапной схемы анализа цифрового изображения, которая позволяет более гибко анализировать рассматриваемые данные. К основным недостаткам алгоритма можно отнести высокую вычислительную сложность. Последнее связано с трудоемкостью вычисления базиса собственных векторов.

5. ЛИТЕРАТУРА

- [1] Starck J. L., Emmanuel J., Candes E., Donoho D. The curvelet transform for image denoising // IEEE Trans. Image Processing, 2002. V. 11, № 6. P. 670 – 684.
- [2] Muresan D. D., Parks T. W. Adaptive principal components and image denoising // IEEE Int. Conf. on Image Processing, 2003. V. 1. P. 101 – 104.
- [3] Comaniciu D., Meer P. Mean shift analysis and applications // IEEE Tran. on Pattern Analysis and Machine Intelligence, 1999. P. 1197 – 1203.
- [4] Buades A., Coll B., Morel J. M. A non-local algorithm for image denoising // IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition, 2005. V. 2. P. 60 – 65.
- [5] Dabov K., Foi A., Katkovnik V., Egiazarian K. Image denoising by sparse 3D transform-domain collaborative filtering // IEEE Tran. on Image Processing, 2007. V. 16, № 8. P. 2080 – 2095.

Об авторах

Волохов Владимир Андреевич – аспирант лаборатории цифровые цепи и сигналы Ярославского государственного университета им. П.Г. Демидова. E-mail: volokhov@piclab.ru

Сергеев Евгений Владимирович – аспирант лаборатории цифровые цепи и сигналы Ярославского государственного университета им. П.Г. Демидова. E-mail: dcslab@uniyar.ac.ru

Формирование и идентификация скрытых голографических водяных знаков в JPEG-изображениях

Владислав Гендин

Кафедра Компьютерной Фотоники и Видеоинформатики

Санкт-Петербургский Государственный Университет Информационных Технологий Механики и Оптики,

Санкт-Петербург, Россия

vlad.gendin@gmail.com

Аннотация

В работе рассматривается метод внедрения синтезированных на компьютере голограмм Фурье и Френеля в JPEG-изображения в виде водяного знака, совпадающего по размеру с изображением-приемником, в частотной области с предварительным уменьшением динамического диапазона голограммы и исследуется устойчивость восстановления водяного знака из JPEG-изображения с различным коэффициентом компрессии.

Keywords: Водяные знаки, Цифровая голография, JPEG

1. ВВЕДЕНИЕ

В последние годы активно развиваются методы защиты цифровых изображений путем внедрения в них скрытых водяных знаков. Скрытые водяные знаки могут предотвратить подмену фотографии на документах или выявить факт подмены на этапе проверки. Скрытые водяные знаки необходимы для защиты авторских прав на изображения, поскольку возможно идентифицировать авторство изображения по водяному знаку.

Некоторые свойства голограмм, например то, что каждая точка голограммы содержит информацию обо всем изображении, открывают широкие возможности для внедрения голограмм в качестве скрытых водяных знаков.

Поскольку в настоящее время изображения чаще всего распространяются в формате JPEG, то рассмотрим метод внедрения скрытых голографических знаков в JPEG-изображения.

2. СИНТЕЗ ГОЛОГРАММ ФРЕНЕЛЯ И ФУРЬЕ

Голограмма Френеля – пропускающая голограмма, при регистрации которой регистрирующая среда находится в области дифракции Френеля рассеянного объектом излучения. Голограмма образуется благодаря интерференции объектной волны в зоне дифракции Френеля с опорной плоской волной.

Распределение интенсивности в плоскости записи голограммы (x, y) представляет собой свертку исходного распределения амплитуд в объекте $u(\xi, \eta)$ и квадратической фазовой функции пространственного распространения волны

$$h(x, y) = u(\xi, \eta) \otimes \left[\frac{1}{i\lambda d} \exp\left(\frac{2\pi i}{\lambda}\right) \exp\left[i \frac{\pi}{\lambda d} (ax^2 + ay^2)\right] \right], \quad (1)$$

где a – размер пикселя ПЗС матрицы камеры, λ – длина волны источника, d – расстояние от объекта до плоскости регистрации голограммы, \otimes – свертка.

В выражении (1) не учитывается угол между опорной и объектной волной, т.е. синтезируется соосная голограмма. Результат синтеза голограммы Френеля представлен на рис. 1, б. Исходный объект, голограмма которого синтезировалась, представлен на рис. 1, а.

Голограмму Фурье можно определить как голограмму плоского объекта, записываемого с помощью опорного точечного источника, расположенного в плоскости объекта, параллельной плоскости голограммы.

Голограмма Фурье может быть рассчитана исходя из того, что преобразование Фурье объектов, имеющих сдвиг в частотной области относительно начала координат, соответствует интерференции волны, рассеянной объектом, с опорной волной [1]

$$h(x, y) = \mathcal{F}^{-1} \{ \gamma H(\xi - N, \eta - M) + \gamma H(-\xi - N, -\eta - M) \}, \quad (2)$$

где \mathcal{F}^{-1} – оператор обратного преобразования Фурье, γ – коэффициент усиления, N и M сдвиги относительно начала координат.

Для уменьшения возмущений в изображении-приемнике при синтезе голограмм Фурье и Френеля исходное изображение умножается на произвольный фазовый множитель [1]

$$\mu(x, y) = \exp(i\xi(x, y)), \quad (3)$$

где $\xi(x, y)$ – случайная фаза (нормальное распределение).

Результат синтеза голограммы Фурье объекта, представленного на рис. 1, а, представлен на рис. 1, с.

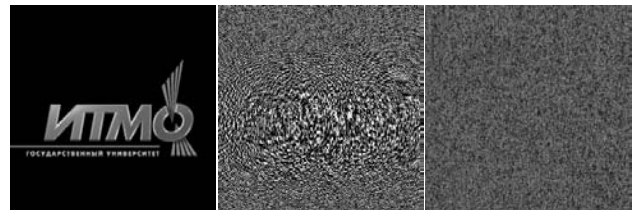


Рис 1: Исходный объект (а), результат синтеза голограммы Френеля (б) и Результат синтеза голограммы Фурье (с).

3. ВСТРАИВАНИЕ ВОДЯНЫХ НАКОВ В JPEG-ИЗОБРАЖЕНИЯ

Внедрение водяного знака в изображение-приемник $g(x, y)$ осуществляется в частотной области. Для этого выполняются следующие действия:

1. Изображения конвертируется в формат BMP.

2. Выполняется преобразование Фурье исходного изображения и голограммы водяного знака.
3. Для уменьшения возмущений, вызываемых внедрением водяного знака, выполняется преобразование динамического диапазона спектра голограммы в интервал от 0 до 16 для 8-битного изображения. Этого диапазона достаточно для точного восстановления голограммы.
4. Спектр исходного объекта складывается с Фурье-образом голограммы с уменьшенным динамическим диапазоном.
5. Выполняется обратное преобразование Фурье.
6. Полученное изображение конвертируется в JPEG формат.

Процесс, описанный в пунктах 2–5 аналитически можно представить следующим образом:

$$s(x, y) = \mathfrak{I}^{-1}\{\mathfrak{I}\{g(x, y)\} + \mathfrak{R}\{\mathfrak{I}\{h(x, y)\}\}\}, \quad (5)$$

где $s(x, y)$ – результирующее изображение с внедренным водяным знаком, \mathfrak{R} – оператор изменения динамического диапазона, $g(x, y)$ – распределение амплитуд в исходном объекте.

На рис. 2, b, c представлены результаты внедрения водяного знака на основе голограмм Френеля и Фурье в изображение, представленное на рис. 2, a.



Рис 2: Исходное изображение-контейнер (a), изображение с внедренным водяным знаком на основе голограммы Френеля (b) и на основе голограммы Фурье (c).

Для идентификации водяного знака достаточно применить один из методов для восстановления соответствующей голограммы. Для голограммы Фурье – преобразование Фурье, для голограммы Френеля – метод Френеля [3].

Результаты восстановления голограмм Френеля и Фурье представлены на рис. 3.

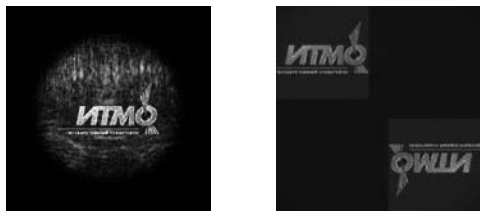


Рис 3: Результат восстановления голограммы Френеля (a) и голограммы Фурье (b).

Поскольку изображение с внедренным водяным знаком может подвергаться JPEG-сжатию с различной степенью качества, то необходимо провести исследование устойчивости восстановления водяного знака из JPEG-изображения с различной степенью качества.

4. ИССЛЕДОВАНИЕ КАЧЕСТВА ВОССТАНОВЛЕНИЯ ВОДЯНОГО ЗНАКА ИЗ ИЗОБРАЖЕНИЯ С РАЗЛИЧНОЙ СТЕПЕНЬЮ КАЧЕСТВА JPEG-СЖАТИЯ

Для количественного сравнения результатов восстановления введем критерий качества восстановленного изображения. Таким критерием может служить коэффициент корреляции исходного водяного знака и результата восстановления водяного знака из изображений, подвергнутых JPEG-сжатию с различным коэффициентом компрессии.

На графике, приведенном на рис. 4, представлены зависимости коэффициента корреляции от коэффициента компрессии JPEG-сжатия.

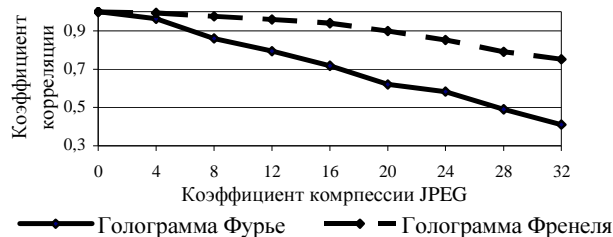


Рис. 4. Сравнение устойчивости методов к JPEG-сжатию.

Из графика видно, что к воздействию JPEG-сжатия голограмма Френеля более устойчива, чем голограмма Фурье.

5. ЗАКЛЮЧЕНИЕ

Представленные в работе методы компьютерного внедрения голографических водяных знаков показали достаточную эффективность и устойчивость восстановления водяного знака. Внедрение в качестве водяного знака голограммы Френеля показало большую устойчивость JPEG-сжатию, чем внедрение голограммы Фурье. Поэтому можно рекомендовать использование в качестве водяных знаков голограмм Френеля в тех случаях, когда изображение будет подвергаться JPEG-сжатию, например при распространении в сети интернет.

6. ЛИТЕРАТУРА

- [1] Михаил Смирнов, “Голографический подход к встраиванию голографических водяных знаков в фотоизображение,” *Оптический журнал*, том 72, № 6, с. 51-56, 2005.
- [2] Сергей Балтийский, Игорь Гуров и др., “Современные методы цифровой голографии” /В кн.: *Проблемы когерентной и нелинейной оптики* /Под ред. И.П. Гурова и С.А.Козлова. С. 91-117, 2004.
- [3] Владислав Гендин, “Формирование и идентификация изображений со встроенными водяными знаками методом цифровой голографии” *Труды научно-технического центра Фотоники и оптоинформатики* /Под ред. И.П. Гурова и С.А. Козлова. С. 406-415, 2009.

7. ОБ АВТОРЕ

Владислав Гендин, аспирант СПбГУ ИТМО, кафедра Компьютерной Фотоники и Видеоинформатики. Адрес электронной почты: vlad.gendin@gmail.com.

Алгоритм модификации типового трехмерного портрета по заданным фотоизображениям

Александра Соловьева

Кафедра «Автоматизированные системы обработки информации и управления»

Ижевский государственный технический университет, Ижевск, Россия

alexandra-solovyova@yandex.ru

Abstract

In the paper a method for typical 3D portrait wireframe adjustment using given full-face and half-face photographs is described. Special attention is given to the processing of the given images with a view to extract information on the peculiar features of the depicted face. An example of sculptural portrait construction is presented.

Keywords: *Sculptural portrait, Photograph, Space curve design, Typical 3D wireframe parameterization, Geometric modeling*

1. ВВЕДЕНИЕ

Моделирование трехмерных портретов находит применение в научных исследованиях в направлении компьютерного зрения, эргономики, психологии, пластической медицины, в дизайне и рекламе. Все чаще построение трехмерного портрета упоминается в контексте биометрической идентификации как альтернатива распознаванию по отпечаткам пальцев.

В то же время воспроизведение трехмерного объекта, простое для сознания человека, оказывается неожиданно сложным для компьютера. Воссоздание сложной геометрической формы человеческой головы связано с необходимостью выполнения большого количества трудоемких построений, часто требующих определенных художественных навыков, что является причиной повышенного внимания к разработке методов автоматизации проектирования трехмерных портретов [1, 2 и др.]. Разнообразие существующих подходов подтверждает перспективность разработок в данной области.

В данной статье рассмотрен алгоритм модификации типового трехмерного каркаса по заданным фотографиям в фас и профиль.

2. ПОСТАНОВКА ЗАДАЧИ

Целью нашего исследования является максимальная автоматизация проектирования фотореалистичного трехмерного портрета [3]. Параметризация типовой трехмерной модели позволяет одновременно сохранить информацию о структуре изображаемого объекта и сократить участие пользователя в моделировании до задания набора определяющих параметров. В качестве источника информации для индивидуализации типового портрета мы используем заданные пользователем фотографии в фас и профиль, полученные при равномерном освещении со стороны фотоаппарата. Используемая нами типовая трехмерная модель формируется кинематическим методом. В состав трехмерного каркаса входят формообразующий контур-сечение, лежащий в горизонтальной плоскости, и контуры-профили, лежащие на поверхности модели, определяющие его деформацию в процессе движения вдоль

вертикальной оси координат, в том числе пара профилей, определяющих проекции объекта на координатные плоскости (соответствующие фотографиям в фас и профиль).

Предлагаемый метод построения трехмерных портретов основан на векторизации изображений, по результатам которой выполняется модификация типового трехмерного каркаса.

3. ОБРАБОТКА ИЗОБРАЖЕНИЯ

Обработка изображения с целью извлечения информации об индивидуальных особенностях лица из заданных фотографий включает в себя четыре этапа.

Этап 1. Обработка изображений. Включает в себя выделение контуров и разбиение изображения на цветовые области.

Выделение контуров, аппроксимирующих перепады яркости изображения, производится с помощью рекурсивного оператора-анализатора локальных областей [4]. Начиная с точек перепада яркости, локальный анализатор рекурсивно отслеживает цепочки контурных пикселей. Разбиение на цветовые области выполняется путем двухэтапной цветовой кластеризации. С помощью кластеризации по параметрам цветности (в пространствах *CIELAB* и *HSI*) производится выделение на изображении основных областей, однородных по цвету. Полученная при этом область лица разбивается на градиации по яркости, отражающие изменения трехмерной формы лица.

Полученные контуры перепадов яркости и границы цветовых кластеров подвергаются векторизации (Рисунок 1).



Рисунок 1: Результат векторизации контуров и границ цветовых областей

Этап 2. Анализ цепочек контуров и границ цветовых областей. Заключается в расчете значений функций, характеризующих расположение и геометрические особенности цепочек примитивов. В числе используемых функций анализа: $Length(c)$ – длина цепочки c , $Count(c)$ – количество примитивов (отрезков прямых и дуг окружностей, составляющих цепочку c), $Quantity(x, y)$ – среднее количество контуров на единицу площади в окрестности точки, $Rect(c)$ – описывающий прямоугольник для цепочки и т.д.

Этап 3. Отбраковка ложной информации. Выделенный набор цепочек геометрических примитивов подвергается спе-

циальной обработке для устранения множества сильно изломанных участков и мелких (либо сопоставимых по длине с размерами изображения) фрагментов, не значимых для распознавания, но значительно снижающих скорость обработки и приводящих к возникновению погрешностей. Причиной их появления может являться чрезмерная зашумленность изображения, наличие теней, контрастные текстуры, недостаточное высокое разрешение предъявленного изображения.

В основу удаления цепочек положена проверка условия

$$(Length(c) < \varepsilon) \vee ((Length(c) > \min(W, H)) \wedge (Count(c) \leq m)),$$

где c – цепочка примитивов; W и H – размеры исходного фотоизображения; ε и m – константы, значения которых выбираются по результатам проведенного анализа). Далее выполняется аппроксимация длинных цепочек контуров и границ цветовых областей с помощью сплайнов. Сплайновая фильтрация позволяет одновременно упростить описание векторизованных цепочек и избавиться от артефактов.

Этап 4. Выявление значимых фрагментов и точек. Осуществляется с помощью экспертной системы, в основу правил которой положены классические пропорции головы и закономерности в расположении контуров и границ цветовых областей изображения. Среди цепочек примитивов выполняется распознавание характерных линейных элементов лица, таких как контур лица, линии губ, век и др. Линейные элементы связываются с характерными точками лица, которые далее используются для первоначального масштабирования трехмерного каркаса. Цепочки геометрических примитивов, формирующие характерный линейный элемент, считаются более значимыми по сравнению с остальными. Сплайнам, описывающим контуры и границы цветовых зон изображения, сопоставляется определенное значение весового коэффициента, определяющее степень существенности их вклада в формирование трехмерной модели.

4. ФОРМИРОВАНИЕ ТРЕХМЕРНОЙ МОДЕЛИ

Первое приближение типового каркаса к результирующей трехмерной модели осуществляется путем его «натягивания» на найденные проекции характерных точек. Далее выполняется синтез сплайновых контуров путем подстановки фрагментов контуров и границ цветовых областей лица. Выполняется обход профилей масштабированного каркаса, сопровождающийся анализом их окрестностей. Для попавших в окрестность текущего профиля цепочек контуров и границ цветовых областей оцениваются значения весовых коэффициентов. Кроме того, для определенных фрагментов профилей, составляющих каркас, более существенными являются векторные цепочки контуров (например, в области глаз, крыльев носа), для других – границы цветовых областей (например, градиции яркости изображения в области щек). Точки с достаточно высокими значениями веса вместе с опорными точками исходного типового профиля образуют сетку, по которой формируется сглаживающий кубический сплайн [5].

Результирующие профили одновременно сохраняют особенности формы исходных элементов трехмерного каркаса и учитывают расположение близлежащих цепочек контуров и границ цветовых областей. Трехмерный портрет приобретает узнаваемые черты (Рисунок 2). Степень сходства полученной модели с исходными данными может быть установлена экспертом.

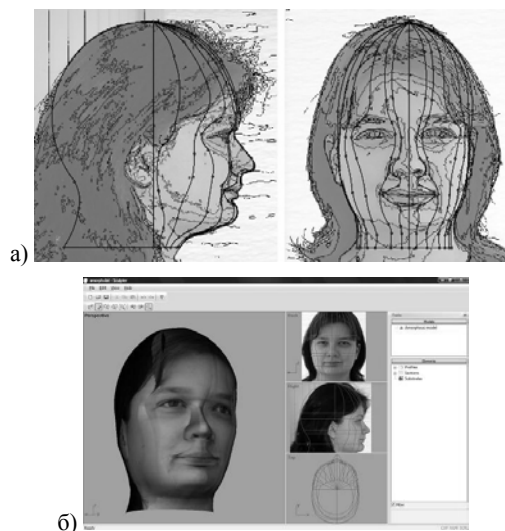


Рисунок 2: Результат модификации типового каркаса:

- а) уточненные профили трехмерного портрета;
б) текстурированный трехмерный портрет

5. ЗАКЛЮЧЕНИЕ

Применение разработанного алгоритма модификации типового трехмерного каркаса по заданным фотоизображениям позволяет автоматизировать трудоемкий процесс ручной корректировки типового трехмерного портрета. Предложенный метод разработан применительно к построению скульптурных портретов по фотографиям в фас и в профиль, однако обладает универсальностью и может быть обобщен для параметризации других типов моделей со сложной геометрией, а также для большего набора заданных изображений.

6. ЛИТЕРАТУРА

- [1] Blanz V., Vetter T. *A Morphable Model for the Synthesis of 3D Faces* // SIGGRAPH'99 Conference Proceedings. P.187-194.
- [2] Вежнев В.П. *Использование контурных моделей для выделения черт лица на фронтальном изображении* // Труды конференции “Математические методы распознавания образов” (ММРО-10), 2001. – с.179-181.
- [3] Solovyova A.N., Kuchuganov A.V. *Using Contours and Colour Region Boundaries of Photographs in Sculptural Portrait Design* // GraphiCon'2009: the 19th International Conference on Computer Graphics and Vision: Moscow, October 5-9, 2009.: Conference Proceedings. – М., 2009. P.315-316.
- [4] Kuchuganov A.V. *Recursions in tasks of images analysis* // Pattern Recognition and Image Analysis: New Information Technologies (PRIA-9-2008). V. 1. P.365-367.
- [5] Шикин Е.В., Плис Л.И. *Кривые и поверхности на экране компьютера. Руководство по сплайнам для пользователей.* – М.: ДИАЛОГ-МИФИ, 1996. – 240 с.

Об авторе

Александра Соловьева – аспирант кафедры АСОИУ Ижевского государственного технического университета.
E-mail: alexandra-solovyova@yandex.ru.

Построение трехмерной модели порового пространства горных пород

Александр Чигорин, Антон Конушин

Лаборатория Компьютерной Графики и Мультимедиа, МГУ им. М.В. Ломоносова, Москва, Россия

aachigorin@gmail.com, ktoh@graphics.cs.msu.ru

Аннотация

В данной работе рассматривается задача построения сетевой модели порового пространства горной породы по серии послойных томографических снимков. Предложен новый алгоритм, состоящий из трех этапов. На первом этапе происходит объединение послойных снимков в трехмерное изображение. На втором - сегментация на материалы с использованием идей гистерезиса и сдвига среднего. На третьем этапе происходит извлечение скелета порового пространства и строится сеть из пор и горлышек, гомотопная скелету.

Ключевые слова: сегментация изображений, скелетонизация, анализ горных пород.

1. ВВЕДЕНИЕ

Современное развитие томографических сканеров позволяет получать послойные снимки образцов горных пород высокого разрешения (рис. 1(a), интенсивностью показана плотность материала). По этим снимкам можно восстановить трехмерную структуру образца и определить его свойства. Такой подход, в отличие от экспериментальной оценки в лаборатории, не разрушает образец и не требует дорогостоящего оборудования. Особый интерес представляет моделирование порового пространства горной породы. В последнее время удалось добиться значительных успехов в определении морфологических, петрофизических и гидродинамических свойств порового пространства, аппроксимированного моделью, состоящей из сети пор и горлышек. Большинство алгоритмов построения сетевой модели состоит из трех этапов. На первом этапе происходит объединение послойных снимков в трехмерное изображение. На втором - сегментация на материалы. На третьем строится сеть из пор и горлышек (с информацией о прилегающих материалах). Недостатком существующих алгоритмов является то, что они работают на образцах простой структуры, в которых можно выделить четкие границы между материалами. В данной статье мы предлагаем новый алгоритм построения сетевой модели порового пространства, призванный устранить указанные недостатки.

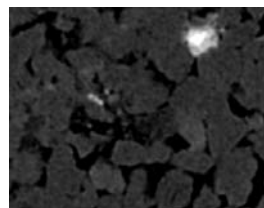
2. ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ

Рассмотрим существующие методы по каждому из этапов.

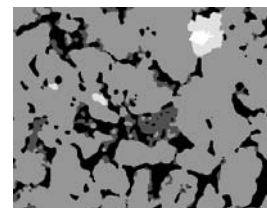
Построение трехмерного изображения. В работе [1] приведен обзор методов построения трехмерного изображения. Данные методы можно разделить на три категории. В первой моделируется процесс образования горных пород, контролируемый данными, извлеченными из снимка одного из слоев. Во второй для получения трехмерного изображения предлагается использовать статистические свойства слоя. В третьей категории трехмерное изображение восстанавливается по послойным снимкам образца. Эта категория методов гарантирует реалистичное представление всех структур образца, размер которых превышает разрешающую способность томографа.

Сегментация. В одном из методов используется анизотропная диффузия для сглаживания и нерезкое маскирование для усиления переходов. В случае простых изображений сегментация осуществляется при помощи пороговой фильтрации, а в случае сложных - при помощи активных контуров. Другой подход к сегментации - это использование индикатора кригинга.

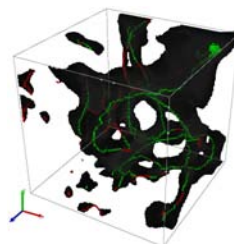
Построение сетевой модели. Методы данного этапа можно разбить на две категории [1] - использующие и не использующие срединную линию (скелет). Если известен скелет, то происходит классификация его вокселей на участки пор и горлышек. Затем происходит



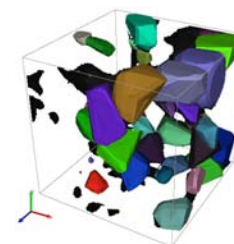
(a) Снимок слоя породы



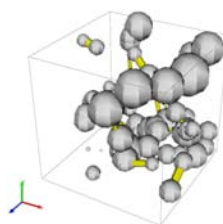
(b) Результат сегментации слоя



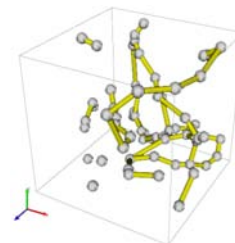
(c) Поровое пространство и его скелет (красные вершины - горлышки, зеленые - поры)



(d) Аппроксимация порового пространства набором многогранников



(e) Визуализация сетевой модели



(f) Связанность сфер в сетевой модели

Рисунок 1: Шаги алгоритма построения сетевой модели

построение сети, гомотопной скелету. Если скелет неизвестен, то используется понятие максимального вписанного шара. С каждым вокселем ассоциируется шар с центром в данном вокселе, касающийся стенок порового пространства. Далее происходит объединение шаров в группы и классификация групп на лежащие внутри пор или внутри горлышек.

3. ОПИСАНИЕ ПРЕДЛОЖЕННОГО МЕТОДА

Мы реконструируем трехмерное изображение по набору послойных снимков образца.

Сегментация. Вследствие особенностей работы томографа изображения получаются зашумленными, а границы между материалами размытыми. Это приводит к нежелательному «затиранию» маленьких горлышек соседними материалами. С другой стороны, замечено, что в области пор и глины практически нет шума - поры всегда черные, а глина немного светлее. Поэтому поры и глину можно выделить отдельно при помощи простого метода сегментации, а остальные материалы отсегментировать более устойчивым методом. Для выделения пор и глины воспользуемся идеей гистерезиса. Пусть $V \in \mathbb{R}^{N \times M \times K}$ - трехмерное изображение образца. Далее описан алгоритм выделения пор и глины:

1. Отсегментируем V с маленьким порогом на поры. Получим

матрицу $C_1 \in \mathbb{R}^{N \times M \times K}$,

$$[C_1]_{ijk} = \begin{cases} 1, & T_{\text{pore}_1} \leq [V]_{ijk} < T_{\text{clay}} \\ 2, & [V]_{ijk} < T_{\text{pore}_1} \\ 0, & \text{иначе} \end{cases},$$

где T_{pore_1} - маленький порог на поры, T_{clay} - порог на глину. 1 соответствует *порам*, 2 - *глине*, 0 - *другим материалам*.

- Выделим в C_1 связанные компоненты, среди точек, отличных от нуля. Пусть матрица $M_1 \in \mathbb{R}^{N \times M \times K}$ задает маску, элементы которой равны 1 в точках, принадлежащих связанным компонентам, в которых есть хотя бы одна точка класса *пора*.
- Необходимо выделить «затертые» маленькие горлышки отдельным способом. Ясно, что интенсивность точек горлышка меньше средней интенсивности по окрестности. Вычислим свертку $S = V \star f(a, b)$, где $f(a, b) = a \exp \frac{-x^2}{2b^2}$ - функция Гаусса. Маска $M_2 \in \mathbb{R}^{N \times M \times K}$,

$$[M_2]_{ijk} = \begin{cases} 1, & [S_1]_{ijk} - [S_2]_{ijk} > T_{\text{throat}}, \text{ где } T_{\text{throat}} - \text{порог на горлышко, указывает на принадлежность точки } (i, j, k) \text{ классу } \textit{пора}. \\ 0, & \text{иначе} \end{cases}$$

- Теперь проведем повторную сегментацию с более высоким порогом на поры, с учетом маски пор M_2 . Получим матрицу $C_2 \in \mathbb{R}^{N \times M \times K}$,

$$[C_2]_{ijk} = \begin{cases} 1, & T_{\text{pore}_2} \leq [V]_{ijk} < T_{\text{clay}} \text{ или } [M_2]_{ijk} = 1 \\ 2, & [V]_{ijk} < T_{\text{pore}_2} \text{ и } [M_2]_{ijk} \neq 1 \\ 0, & \text{иначе} \end{cases}$$

- Из-за нечеткости границ между материалами возникают плавные переходы интенсивности шириной в 2-3 вокселя. Они классифицируются как глина, что неверно. Составим маску глины $M_3 \in \mathbb{R}^{N \times M \times K}$

$$[M_3]_{ijk} = \begin{cases} 1, & [C_2]_{ijk} = 2, \\ 0, & \text{иначе} \end{cases},$$

и осуществим над ней морфологическую операцию открытия $M_4 = M_3 \circ S$, $S \in \mathbb{R}^{3 \times 3 \times 3}$, $[S]_{ijk} = 1$, где \circ - поэлементное произведение матриц (произведение Адамара)

- Учтем полученные маски для получения итоговой сегментации. $C = C_2 \circ (M_1 \circ M_4)$.

Далее необходимо детализировать классификацию точек с меткой *другие материалы*. Из-за наличия шума необходимо учитывать окрестность каждой точки, причем требуется, чтобы граница окрестности совпала с границей между материалами. Для этого пересегментируем изображение при помощи сдвига среднего [2] с маленькой шириной окна по интенсивности. Далее классифицируем полученные сегменты при помощи метода ближайшего соседа на основе квадрата-деревьев. В качестве признаков для классификации выступают математическое ожидание и дисперсия интенсивности по сегменту. Оказалось, что для обучения классификатора, пользователю нужно указать два-три сегмента для каждого класса. Результаты сегментации приведены на рис. 1(b).

Построение сетевой модели. Следующим шагом алгоритма является построение графа P порового пространства, аппроксимирующего его структуру. Вершины графа соответствуют порам, а ребра - горлышкам (рис. 1(f)). Каждой вершине и ребру приписываются атрибуты - общий объем, радиус, длина, объем глины, shape factor [4]. На первом этапе мы извлекаем скелет порового пространства при помощи метода, описанного в [3]. Для полученного скелета строим граф S , вершинам которого соответствуют воксели скелета. Две вершины графа S соединены ребром, если соответствующие им воксели 26-связаны. Затем классифицируем воксели скелета на

воксели, лежащие внутри пор или внутри горлышек. Каждой вершине графа S сопоставляется радиус поперечного сечения порового пространства, вокруг соответствующего данной вершине вокселя. Если радиус больше определенного порога, то эта вершина соответствует поре, иначе - горлышку. Пример порового пространства с классифицированным скелетом приведен на рис. 1(c). Далее, следуя по скелету, разбиваем поровое пространство на набор многогранников, как показано на рис. 1(d). Каждый многогранник является единицей декомпозиции порового пространства и в графе порового пространства P ему соответствует вершина. Связанность многогранников (и вершин графа P) задается скелетом. При этом мы уже можем вычислить все атрибуты вершин и ребер P . Если заменить многогранники сферами, а горлышки цилиндрами, то сетевую модель можно визуализировать (рис. 1(e)). Чтобы показать связанность, удобно изображать сферы меньшего радиуса (рис. 1(f)).

4. ЭКСПЕРИМЕНТЫ

Для проверки корректности работы алгоритма сегментации вручную были размечены снимки слоев нескольких образцов. Вероятности ошибок первого и второго рода при классификации пор получились равными 6% и 7% соответственно. В случае глины - 7% и 30%. Ошибки классификации возникают на краях, где даже человеку сложно выделить четкую границу.

Сетевая модели, полученная для песчаника Berea, сравнивалась с моделью из работы [4]. В качестве характеристик модели выступали *пористость, степень вершины, радиус горлышка и радиус поры*. Пористость нашего образца составила 21%, а пористость образца из работы [4] - 23%. Поэтому можно было предположить, что образцы имеют схожую структуру. Для остальных характеристик были построены гистограммы распределения принимаемых ими значений. Средняя степень вершины в случае нашей модели и модели из [4] составила 2 и 3.9 соответственно, средний радиус горлышка - 7.7μ и 12.2μ , а средний радиус поры - 38μ и 25μ . Полученные различия легко объяснить различными подходами к построению модели. Среднее значение радиуса горлышка в [4] получается больше, потому что в нашем случае радиус горлышка ограничен пороговым значением, а в [4] радиус горлышка может быть произвольным и часто получается больше порогового значения в нашей модели. Средний радиус поры уменьшается при увеличении среднего радиуса горлышка, так как большие поры чаще делятся на части. А средняя степень вершины зависит от количества горлышек, которых больше в модели, предложенной в [4], так как их максимальный радиус не ограничен.

5. ЗАКЛЮЧЕНИЕ

В данной работе предложен метод построения сетевой модели образца горной породы. Проведено сравнение с одним из существующих методов на примере вычисления модели для песчаника Berea. В дальнейшем планируется подготовить набор тестовых образцов с известными характеристиками и провести сравнение реальных и получаемых при помощи моделирования характеристик.

6. ЛИТЕРАТУРА

- [1] A. S. Al-Kharusi and M. J. Blunt, "Network extraction from sandstone and carbonate pore space images", *Journal of Petroleum Science and Engineering*, vol. 56, pp. 219–231, 2007.
- [2] P. Meer D. Comaniciu, "Mean shift, mode seeking, and clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
- [3] R. Kashyap T. Lee and C. Chu, "Building skeleton models via 3-d medial surface/axis thinning algorithms", *Computer Vision, Graphics, and Image Processing*, vol. 56, no. 6, pp. 562–478, 1994.
- [4] S. Bakke P. Oren, "Reconstruction of berea sandstone and pore-scale modelling of wettability effects", *Journal of Petroleum Science*, vol. 39, pp. 177–199, 2003.

Стереовизуализация научных и медицинских объемных данных трассировкой лучей в реальном времени

Николай И. Гаврилов, Александра А. Белокаменская
факультет Вычислительной Математики и Кибернетики
Нижегородский государственный университет, Н.Новгород, Россия
gavrilov86@gmail.com; aab04@yandex.ru

Аннотация

Предложен метод ускорения прохода пустых вокселей, заключающийся в использовании двухуровневой иерархии текстур: помимо основного массива, 512^3 , в GPU загружается двухканальная текстура $16 \times 16 \times 16$. Каждому элементу этой текстуры соответствует свой блок $32 \times 32 \times 32$ исходных данных. Текстура хранит минимальное и максимальное значения данных соответствующего блока. Предлагаемый метод ускорения алгоритма позволяет добиться качественной стереовизуализации мультиобъемов в реальном времени. Построена программа Stereo Multi-Volume Viewer (SMV), реализующая стерео рендеринг мультиобъемных данных в реальном времени на лучших бытовых моделях GPU. Выполнено сравнение производительности реализации на GLSL и на OpenCL.

Ключевые слова: GPGPU, Multi-Volume Rendering, прямой объёмный рендеринг, медицинская визуализация, иерархия 3D текстур

ВВЕДЕНИЕ

В научной визуализации часто приходится иметь дело со скалярными полями данных, заданными в трехмерном пространстве. Данные могут быть получены самыми различными способами: численный эксперимент, сканирование с помощью магнитного резонанса, компьютерная томография (КТ), сканирование с помощью ультразвука. Например, результатом КТ является множество слоёв, т.е. двумерных массивов данных, которые вместе образуют трехмерное скалярное поле. Объем данных эксперимента значителен. Например, объем одного исследования в медицинской томографии, обычно равен 0.1-1ГБ. Обеспечение реального времени визуализации означает вывод на экран порядка 25 и более кадров в секунду.

Метод прямого объёмного рендеринга методом обратной трассировки лучей с 90-х годов прошлого века позиционирует себя как эффективный инструмент для визуального анализа объёмных данных. Разработаны и доведены до широкой научной общественности [1] подходы, позволяющие решать задачи объёмного рендеринга в реальном времени на основе параллельных вычислений и высокопроизводительной вычислительной техники. Эффективно решаются в интерактивном режиме задачи сегментации объема [2]. Развитие GPU как вычислительных устройств в последние годы дало этому направлению новый значительный импульс и позволило говорить о многообъемном рендеринге в реальном времени [3-5]. Стереовизуализация научных данных достаточно давно культивируется в бостонском университете (<http://scv.bu.edu>), но не вошла пока в широкую практику в медицине.

МЕТОДЫ ВИЗУАЛИЗАЦИИ

В медицине томограммы хранятся в специальном формате данных DICOM, а для визуализации томограмм используют так называемые DICOM-визуализаторы. Довольно большая часть их общедоступна, в том числе: eFilm, MicroDicom, OsiriX, Onis-Viewer, 3DimViewer, RadiAnt и отечественный (www.multivox.ru) MultiVox DICOM Viewer. Все эти Viewer-ы обеспечивают визуализацию томограммы в виде классических полутоновых изображений 2D-проекции и сечений объёмного скалярного поля. Большинство Viewer-ов обеспечивают также возможность визуализации 3D-моделей томограммы, используя чаще всего простые модификации прямого объёмного рендеринга (Рис.1).



Рис.1. Пример 3D-визуализации в eFilm

Наилучшего качества и информативности объёмного рендеринга позволяет достичь метод обратной трассировки лучей. С помощью этого метода можно визуализировать данные как набор изоповерхностей или в виде тумана, или и то и другое вместе. У изоповерхностей и тумана можно варьировать прозрачность и цвет. В научном эксперименте и дорогих профессиональных системах можно встретить примеры мультиобъемного рендеринга, к которому пока не предъявляют требования реального времени (Рис.2).

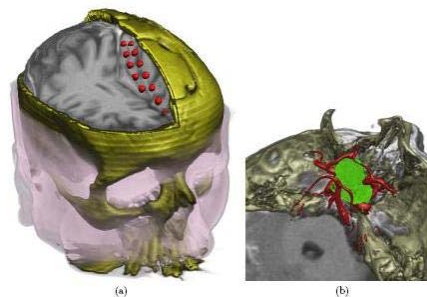


Рис.2. Пример мультиобъемного рендеринга [5]

Высокое качество изображения формирует интерес к использованию стерео-изображений, в которых, также как в прямом рендеринге, используется цвет и прозрачность вокселей.

РЕАЛИЗАЦИЯ ТРАССИРОВКИ ЛУЧЕЙ ДЛЯ СТЕРЕОВИЗУАЛИЗАЦИИ ОБЪЁМОВ НА GPU

Наиболее удобной структурой хранения пространственных массивов данных на GPU является 3D-текстура, аналогичная 3D-массиву действительных чисел. Для CPU удобнее может оказаться иерархическая структура, например, октодереве. Для доступа к элементам дерева процессору необходимо пройти множество ветвлений, что приемлемо для CPU, но убивает производительность для GPU, к тому же, в программах для GPU нет рекурсии. Программа, реализующая трассировку, выполнена в среде MS Visual Studio 2008, как шейдерная программа для GPU, в среде OpenGL и GLSL (шейдерный язык для OpenGL).

Предложен метод ускорения вычислений, заключающийся в использовании иерархии текстур: помимо основного массива, $512 \times 512 \times 512$, в GPU загружается меньший массив – двухканальная текстура $16 \times 16 \times 16$. В пространстве каждому элементу этой текстуры соответствует свой блок $32 \times 32 \times 32$ исходных данных. Текстура хранит минимальное и максимальное значения данных соответствующего блока.

Обращаясь к маленькой текстуре, луч может пропускать «неинтересные» области пространства, двигаясь с большим шагом. Такое решение повышает FPS в среднем в три раза.

3. РЕЗУЛЬТАТЫ

Тестовые данные предоставлены ИПФ РАН, а также получены из Интернет-ресурса [1]. Это как результаты численного моделирования в физике, так и результаты КТ.

Ниже представлена производительность в зависимости от размера данных. Эксперимент проводился на видеокарте GeForce 8600 GT.

Объем	128^3	256^3	512^3	$512 \times 512 \times 690$
FPS	50-60	30-40	8-15	5-10

Благодаря использованию ускоряющей структуры, увеличение размера объёмных данных меньше сказывается на производительности. Качественная визуализация больших массивов данных в реальном времени выделяет созданную программную систему среди многочисленных аналогов. Для сравнения производительности OpenCL и GLSL-шейдеров написаны две программы, реализующие алгоритм без ускоряющей структуры, рисующий одну изоповерхность для плотности 65 (плотность кости) и освещенный туман для данных всего диапазона плотностей. Протестировано на данных размером 256^3 . Для проведения эксперимента использовался графический ускоритель NVIDIA GeForce 8600GT и

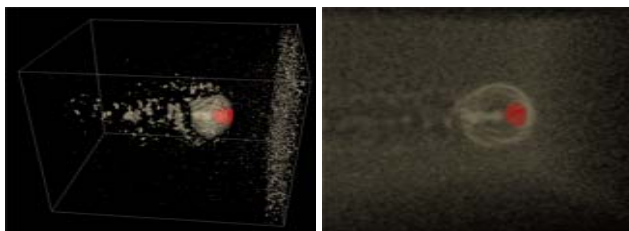


Рис.2. Визуализация взаимодействия двух сред в SMV

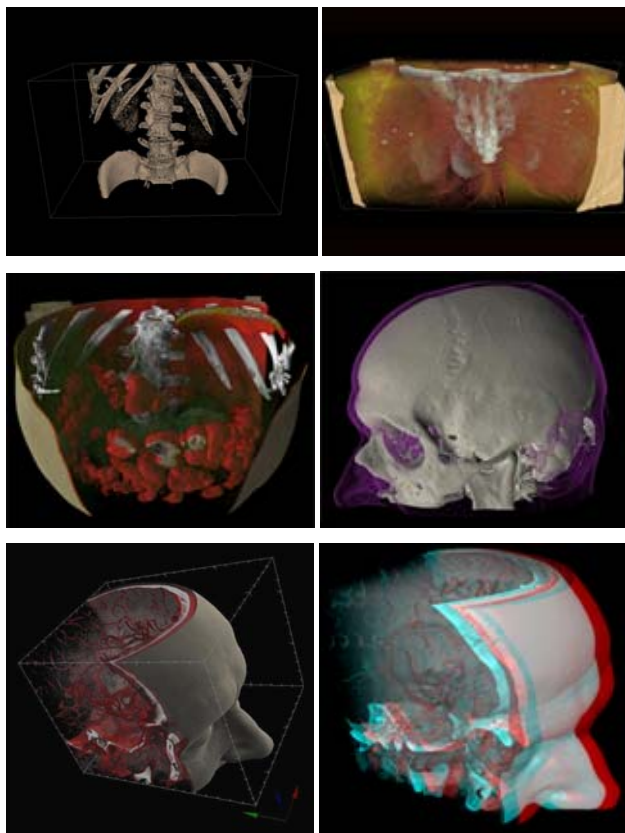


Рис.2. 3D и стерео визуализация в программе SMV

центральный процессор Intel Core2Duo 2,33 GHz. Размер окна визуализации составлял 512×512 точек. Производительность на OpenCL составила 5,6 fps, на GLSL – 10,2 fps. Переход на GPU GeForce GTX 285 | Radeon HD 5870 дает не менее, чем четырехкратный прирост производительности.

4. ЛИТЕРАТУРА

- [1] Klaus Engel, Markus Hadwiger, Joe M. Kniss, Aaron E. Lefohn, Christof Rezk Salama, Daniel Weiskopf; *Real-Time Volume Graphics, SIGGRAPH-2004*
<http://old.vrvis.at/via/resources/course-volgraphics-2004/>
- [2] Ятченко А.М., Крылов А.С., Гаврилов А.В., Архипов И.В. Построение 3D модели кровеносных сосудов по серии КТ изображений печени / Conf. Proc. of the 19th International Conference on Computer Graphics and Vision “GraphiCon’2009”, Moscow, 2009
- [3] Friedemann Roßler, Eduardo Tejada, Thomas Fangmeier, Thomas Ertl, Markus Knauff, *GPU-based Multi-Volume Rendering for the Visualization of Functional Brain Images*
<http://www.vis.uni-stuttgart.de/ger/research/pub/pub2006/simvis06-roessler.pdf>
- [4] Ralph Brecheisen, Anna Vilanova i Bartoli, Bram Platel, Bart ter Haar Romeny. *Flexible GPU-Based Multi-Volume Ray-Casting* // Technical University of Eindhoven
<http://www.yip.wtb.tue.nl/pdfs/9881.pdf>
- [5] Beyer Johanna. GPU-based Multi-Volume Rendering of Complex Data in Neuroscience and Neurosurgery. - Dissertation/ Vienna University of Technology, okt.2009, 131p

- [6] The scientific computing and imagine institute at the university of Utah
<http://www.sci.utah.edu/download/IV3DData.html>

Abstract

Stereo-visualization of scientific and medical volume data in real time using the ray tracing

This work presents a method of volume rendering acceleration for GPU based on an empty voxel skipping. The skipping is constructed on the use of two-level hierarchy of textures: the main 3D texture-array (source data, 512^3) and a small two-channel texture $16 \times 16 \times 16$. Every $32 \times 32 \times 32$ block of the main texture corresponds to an element of the small texture. The small texture contains the minimal and maximal values of the block. The proposed method helps to achieve high-quality stereo visualization of multi-volume in real time. The program Stereo Multi-Volume Viewer (SMV) implements stereo rendering multi-volume data in real time on the best popular models of GPUs. The performances of GLSL and OpenCL versions have been compared.

Keywords: GPGPU, Multi-Volume Rendering, direct volume rendering, medical visualization, 3D texture hierarchy.

Интерактивная обучающая система «Освещение в 3ds Max»

Татьяна Сергеевна Бобышева
Технолого-экономический факультет

Нижегородский государственный педагогический университет, Нижний Новгород, Россия
tanya.bobysheva@gmail.com

Аннотация

Интерактивная обучающая система «Освещение в 3ds Max» представляет собой мультимедийный комплекс, реализованный в среде Adobe Flash. В нем рассматриваются приемы освещения трехмерных сцен, используемые в фотографии, изобразительном искусстве, кинематографе, а также теоретические основы компьютерной графики. Реализованы технологии разноуровневого обучения и обучения в игре, построенные на эффективных и эффектных способах подачи информации, которые способствуют легкому усвоению сложного материала.

Ключевые слова: 3ds Max, освещение, учебное пособие, интерактивный курс, обучение в игре, мультимедиа.

1. Актуальность

Одним из наиболее сложных и ответственных этапов разработки трехмерной сцены является ее освещение, от которого во многом зависит качество итогового изображения. Однако в многочисленных учебных изданиях и статьях, посвященных работе в 3ds Max, этой проблеме уделяется недостаточное внимание.

Обеспечить наиболее полное и эффективное представление вопросов освещения трехмерной сцены можно с помощью применения в учебном процессе информационных технологий. Благодаря широким дидактическим и воспитательным возможностям электронных учебных изданий, их разработка и применение позволят решить проблему усвоения сложного теоретического материала.

2. Инструменты создания мультимедийной обучающей системы

Электронное учебное пособие «Освещение в 3ds Max» создано с помощью программы Adobe Flash CS4 и языка программирования ActionScript 3.0.

Подготовка статических иллюстраций велась с использованием программ Adobe Photoshop и Adobe Illustrator. Анимационные ролики разрабатывались средствами Adobe Flash.

3. Структура электронного учебного пособия

Электронное учебное пособие «Освещение в 3ds Max» представляет собой систему взаимосвязанных компонентов: блока теории, комплекса практических заданий, блока самоконтроля и библиотеки вспомогательной информации.

Блок теории состоит из трех разделов, в содержании которых выделены общие вопросы (базовый уровень сложности) и

дополнительный материал, позволяющий реализовать технологию разноуровневого обучения.

Блок контроля и самоконтроля включает предварительное и итоговое тестирования.

Практическая часть состоит из комплекса упражнений и заданий различных уровней сложности. Выполнение упражнений направлено на формирование умений и навыков, а работа над заданиями обеспечивает их комплексное закрепление.

В качестве вспомогательного материала представлен глоссарий, список литературы, практические советы, интересные факты применения программ моделирования в кинематографе.

В пособии функционирует разветвленная система ссылок (на отдельные темы, глоссарий и список литературы), всплывающие подсказки (короткие определения терминов) и вставки материала повышенного уровня сложности.

4. Содержание мультимедийного учебного курса

Мультимедийный учебный курс «Освещение в 3ds Max» состоит из трех разделов:

- I. Базовые сведения о локальных моделях освещения, используемых в 3ds Max.
- II. Приемы освещения сцен. Общие принципы постановки освещения. Имитация различных условий освещения (на примере интерьера и экстерьера).
- III. Сложное освещение сцен. Природа света. Математические и алгоритмические основы освещения в компьютерной графике. Настройка глобального освещения (Scanline Render, Mental Ray, VRay).

В качестве примера реализации предложенных подходов к освещению, демонстрируется дизайн-проект художественной мастерской. В нем представлены визуализации сцены с различными условиями освещения (общее естественное освещение, общее искусственное освещение, местное освещение модели и рабочих мест). Приведено подробное описание постановки света для данной сцены, а также результаты сравнения (по качеству и времени визуализации) различных инструментов освещения.

5. Формы представления учебного материала

Интерактивность учебного пособия позволяет подавать теоретический материал в наиболее эффективных формах:

- простой повествовательный материал представлен в виде текста с большим количеством статических иллюстраций и анимационных роликов;

- более сложный материал, раскрывающий основополагающие понятия, требует иной подачи, например, в форме аудио-лекции (по ходу воспроизведения постепенно формируются схемы, что обеспечивает максимальную наглядность; после прослушивания есть возможность прочитать тезисы для закрепления полученных знаний);
- для обеспечения неразрывности процессов получения знаний и формирования умений и навыков, в пособии активно используются интерактивные тренажеры.

В каждом разделе представлены интерактивные схемы-обобщения, позволяющие в игровой форме повторить ключевые моменты теории.

Для подведения итогов изучения теоретических вопросов предлагается пройти предварительное тестирование. После ответа на вопрос пользователю сообщается о правильности или неправильности выбора, и дается верный ответ. Таким образом, обеспечивается повторение и закрепление материала, коррекция знаний и рефлексия обучаемого.

Итоговое тестирование выполняет функцию оценки знаний, здесь правильные ответы уже не приводятся. Во время выполнения теста все кнопки становятся неактивными. После его прохождения дается словесная оценка результатов, процентное соотношение верных и неверных ответов и рекомендации по повторению отдельных тем.

Для формирования умений и навыков предлагается несколько упражнений. Четко обозначаются цели и перечень опорных знаний. Используются три формы организации упражнений:

1. Словесное пошаговое описание выполнения упражнения.
2. Видео-урок, в котором каждое действие подробно комментируется, есть возможность приостановить воспроизведение для выполнения части упражнения в окне 3ds Max.
3. Эмуляция работы в программе 3ds Max.

Первые две формы организации упражнений доставляют пользователю неудобства, так как описание действий находится в одном окне, а непосредственное выполнение задания осуществляется в другом. Решением проблемы переключения между окнами стала эмуляция работы в программе 3ds Max средствами Adobe Flash. Таким образом, происходит соединение обучения и игры, а также обеспечение максимально комфортных условий обучения.

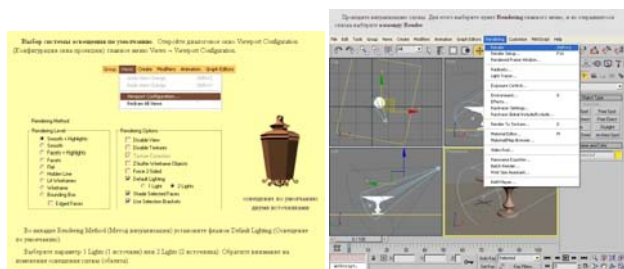


Рис. 1: Интерактивный тренажер (слева) и имитация работы в 3ds Max средствами Adobe Flash (справа)

Задания представляют собой предложения по модификации сцен, созданных в ходе упражнений. В зависимости от

объема начальных данных и степени самостоятельности при выборе решения, задания делятся на 3 уровня сложности:

- задание с подробным объяснением последовательности достижения результата;
- задание с описанием конечного результата и несколькими вариантами его достижения;
- задание с описанием конечного результата без предложений по его достижению.

6. Дизайн и функциональные возможности электронного учебного пособия

При разработке дизайна интерфейса учитывался комплекс эргономических и психолого-педагогических требований.

Главная навигационная панель оформлена в виде списка, что позволяет в любой момент перейти к нужному разделу. При этом доступ к подуровням содержания осуществляется через слайд с перечнем тем раздела. Форма и расположение всех элементов функционально оправданы. Для максимально комфортной работы отсутствует полоса прокрутки текста.

Применяются эффектные приемы оформления, призванные привлечь внимание пользователя. В то же время, эффективность способствует эффективности интерактивного обучения, благодаря реализации подхода «обучение в игре».

Основные функциональные возможности обучающей системы: печать страниц (с иллюстрациями и без), поиск по тексту и по глоссарию, просмотр ключевых иллюстраций в полноэкранном режиме, добавление страниц в закладки.

7. Применение интерактивной обучающей системы «Освещение в 3ds Max»

Электронное учебное пособие «Освещение в 3ds Max» разрабатывалось с учетом особенностей детей старшего школьного возраста, но благодаря осуществлению технологии разноуровневого обучения, оно может использоваться студентами учреждений высшего и среднего профессионального образования.

Кроме того, благодаря максимальной интерактивности обучающей системы и насыщенности иллюстративным материалом, разработанное электронное учебное пособие будет интересно любому желающему изучить процесс работы в 3ds Max.

Сведения об авторе

Бобышева Татьяна – выпускница Нижегородского государственного педагогического университета, технологический факультет.

Abstract

Interactive training course Lighting in 3ds Max

Lighting in the 3ds Max is an interactive training system, which represents a multimedia complex realized in Adobe Flash. The system teaches a short course of the three-dimensional scene lighting about techniques which are used in photography, art, and movie industry. The course includes also theoretical foundations of computer graphics. One of the interesting features of the system is a training-in-game didactic technique.

Разработка графического движка для компьютерной игры на GLSL

Александр Эдуардович Веселов
факультет вычислительной математики и кибернетики
Казанский государственный университет, Казань, Россия
try9998@gmail.com

Аннотация

В работе обсуждается создание современного движка для игр, широко использующего возможности шейдерного программирования и параллельных вычислений на графических процессорах (GPU). Решаются традиционные для движка задачи: симуляция окружающей обстановки и персонажей. В том числе задача динамической симуляции растительности (травы, кустов, хвойных и лиственных деревьев) и облаков, как наиболее динамичных объектов. Разнообразие персонажей и визуальных эффектов также решено на основе шейдеров оптимальным по ресурсам и производительности образом.

Ключевые слова: разработка игр, шейдеры, OpenGL, GLSL,

1. Постановка задачи и актуальность

С развитием возможностей графических процессоров происходит и развитие игр, использующих эти новые возможности. Поэтому остается актуальным создание все новых движков для игр, модифицирующих традиционные алгоритмы к новым возможностям параллельных вычислений на графических процессорах (GPU). Традиционные задачи движка: симуляция окружающей обстановки и персонажей.

В части симуляции окружающей обстановки вменим себе задачу динамической симуляции растительности (травы, кустов, хвойных и лиственных деревьев) и динамической симуляции неба, прежде всего облаков, как наиболее динамичных объектов.

В части симуляции персонажей поставим задачу разнообразить персонажи и обеспечить оптимальные по ресурсам интересные визуальные эффекты. Пример построения трехмерных сцен игры, на рис.1, дает представление о необходимом аппарате.



Рис.1. Пример построения трехмерной сцены игры

2.Динамическое окружение

Моделирование облаков

Общепризнанным методом моделирования облаков являются шумы Перлина [1]. Однако реалистично визуализировать облака только шумом Перлина не удастся, что наглядно показано на рис.2.

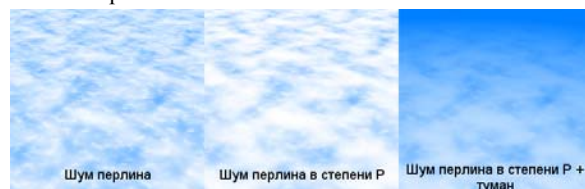


Рис.2.Шум Перлина и «туман» в симуляции облаков

Еще один усложняющий фактор – движение облаков. Для реалистичной анимации, облака представлены слоями, каждый слой движется со своей скоростью.

Хвойные деревья

С ними обычно все просто, однако было замечено, что добавление отражающего компонента освещения значительно улучшает внешний вид (рис.3).



Рис.3.Влияние specular составляющей

Широколиственные деревья

Для этого типа деревьев наиболее удобным является представление в виде: ствол + система частиц. Традиционная реализация [2] показана на рис.4.

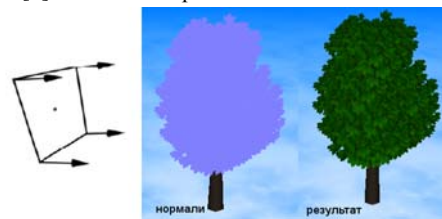


Рис.4.Внешний вид при традиционной визуализации

Однако, учитывая классические алгоритмы [3], современные шейдерные реализации алгоритмов освещения по Борескову А.В. [4] и последние рекомендации конференции разработчиков игр в Сан-Франциско [5] можно построить более интересные визуальные модели приблизительно с теми же временными затратами (рис.5).



Рис.5. Использование карты нормалей для увеличения эффекта трехмерности

Трава

Для моделирования травы используем только диффузное освещение. Для индивидуализации каждого куста травы внесем цветовой шум и случайный поворот куста вокруг своей оси. Для естественности движения травы под действием ветра реализуем волновое движение травы. Анимация деревьев и кустов выполнена аналогично.



Рис.6. Травинки склоняются пробегающей волной с разбросом направлений наклона

Кристаллы

По замыслу кристаллы должны быть светящимися, поэтому для них можно применить аддитивный закон смешивания и тем самым обеспечить прозрачность независимую от порядка. Для засветки краев применена модель освещения Rim [4].



Для повышения визуального качества, можно построить на основе вершин кристаллов систему частиц.

Рис.7. Пример симуляции кристаллов с системой частиц

3. МОДЕЛИРОВАНИЕ ПЕРСОНАЖЕЙ

Выбранная тематика игры такова, что каждый персонаж, зачастую, имеет нестандартный алгоритм визуализации. Рассмотрим наиболее интересные из персонажей и вопросов их моделирования.

Моделирование огня

Для моделирования огня используем, согласно [5], аддитивное смешивание цветов и вершинную анимацию (рис.7).

Огненный элемент (на рис.8 - слева) выглядит убедительно, однако, голем создан без участия художника и оставляет желать лучшего. Для повышения качества огня можно увеличить количество частиц, но более эффектно использовать сдвиг частиц относительно голема (рис.9).



Рис.8. Конструирование огненного голема из 2 частей



Рис.9. Сдвиг -0.5; сдвиг 0.0; сдвиг 0.5.

Достоинства метода очевидны: значительное повышение качества, почти без изменения нагрузки на GPU.

Визуализация призрака

В результате сложившейся практики качественный призрак не должен содержать диффузную составляющую освещения (не должен иметь собственного цвета) - только отражающую. Однако, полностью отказаться от диффузного освещения мы не можем, так как образ не будет хорошо читаемым. Поэтому мы не будем полностью избавляться от диффузного освещения, а только ослабим его (рис.10).



Рис.10. Пример симуляции призрака

4. ЗАКЛЮЧЕНИЕ

Оборудование, на котором тестировалась производительность движка: ATI Mobility Radeon HD 2400, 128 Mb. Результаты:

Небо	1000 fps
Анимированных деревьев: 100 по 34 quads	516 fps
Анимированных кустов травы: 394/400, издалека	120 fps
Анимированных кустов травы: 84/400, ср. расст-е	25 fps
Огненных големов: 100 по 11000 треуг, 33000 вер.	35 fps

Тестирование производительности показало приемлемое время для эксплуатации движка на ноутбуке с современным GPU.

5. ЛИТЕРАТУРА

- [1] Ken Perlin's course notes from the GDCHardCore gamers workshop / San Francisco, Dec 9, 1999. http://freespace.virgin.net/hugo.elias/models/m_clouds.htm
- [2] Ворсин С.В. Визуализация лесных массивов и рельефа местности в реальном времени / GameDev.ru - Разработка игр, 2004 (обн.2010) (<http://www.gamedev.ru/code/articles/?id=4211>)
- [3] Holly Rushmeier, David Banks, Peter Shirley. "A Basic Guide to Global Illumination"/ SIGGRAPH 98 Course 5 <http://www.cs.princeton.edu/courses/archive/fall02/cs526/papers/rushmeier98.pdf>
- [4] Боресков А.В. Модели освещения (с примерами на GLSL) / Сайт Steps 3D. 2003-2010 <http://www.steps3d.narod.ru/tutorials/lighting-tutorial.html>
- [5] Game Developers Conference/ March, 9-13. 2010, San Francisco (http://drewskillman.com/GDC2010_VFX.pdf)

Сведения об авторе

Александр Эдуардович Веселов - студент 2 курса, факультет вычислительной математики и кибернетики,

Казанский государственный университет, Казань, Россия
try9998@gmail.com.

Об одной реализации метода фотонных карт для визуализации каустик

Ульянов Данила Ярославович
Нижегородский Государственный Технический Университет им. Р. Е. Алексева
Нижний Новгород, Российская Федерация
danila-ulyanov@narod.ru

Аннотация

В данной работе описывается метод визуализации каустик на основе техники трассировки лучей. Генерация карты фотонов и последующая визуализация выполняются на графическом процессоре. Для повышения эффективности используется пространственная хэш-функция и битоническая сортировка. Реализация алгоритма базируется на языке C++ и OpenGL / GLSL. Даны оценки производительности.

Keywords: Ray tracing, Caustics, Photon maps, GPU sorting.

1. ПОДХОДЫ К ПОСТРОЕНИЮ КАУСТИК

В основе расчётов каустик лежит, как правило, прямая трассировка лучей от источников света. Результат трассировки обычно сохраняется в специальную базу данных – *фотонную карту* – и используется на этапе визуализации. Данный подход моделирования глобального освещения, и каустик в частности, называется методом фотонных карт. Существует, по крайней мере, ещё один способ визуализации каустик, называемый *caustics mapping* [2]. Выбор между этими двумя методами, прежде всего, зависит от способа визуализации сцены. Это либо стандартный для графического конвейера метод растеризации примитивов, либо метод обратной трассировки лучей. Техники глобального освещения (GI), обеспечивающие попутно и построение каустик, пока не доступны для реального времени. Рассмотрим далее другие варианты получения нужных эффектов.

1.1 Caustics Mapping

Метод разработан для наиболее применимого, в настоящее время, способа визуализации сцены, называемого растеризацией. Примитивы, из которых состоит сцена, проецируются на плоскость экрана и сортируются по глубине. Реализация данного метода рассмотрена подробно в [2] (рис.1). Метод дан для сравнения с основным методом, реализованным в статье.

При проективном построении сцены авторы предлагают использовать отдельный вершинный буфер в качестве набора фотонов. В вершинном шейдере данные фотоны трассируются через сцену по законам отражения и преломления. Весь алгоритм выполняется в пространстве изображения (*image space*), поэтому информация о геометрии сцены передаётся через заранее построенные карты нормалей и глубины. Результатом работы алгоритма является карта каустик, которую можно отфильтровать и проективно наложить на геометрию сцены. Основные ограничения подхода схожи с таковыми в технике теневых карт (*shadow maps*). При реализации данного подхода могут возникать сложности с фильтрацией карты каустик и ее нежелательным зашумлением (см. рис. 2). Существенное ограничение на

применение метода налагает также ограниченный размер полученной карты.

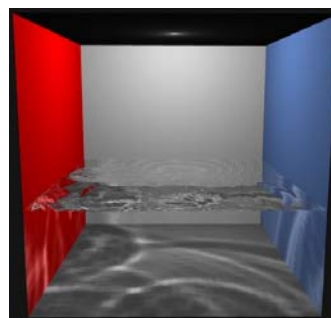


Рис. 1. Пример подводных каустик, построенных методом Caustics Mapping [2]

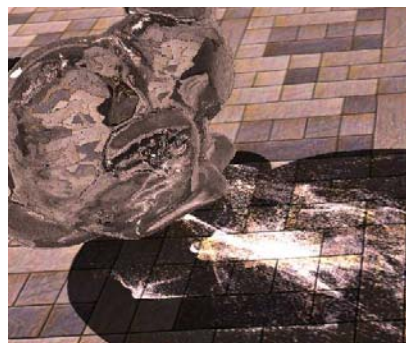


Рис. 2. Пример зашумления карты каустик [5]

Тем не менее, метод вполне пригоден для использования в графических приложениях реального времени.

1.2 Photon Mapping

Для метода визуализации сцены с помощью обратной трассировки лучей используется метод фотонных карт. Механизм трассировки лучей от наблюдателя (для визуализации) и от источника света (для построения фотонной карты) практически одинаков, что позволяет использовать для трассировки фотонов уже имеющийся в реализации расчета изображения код трассировки. Об этом методе и пойдёт речь в далее.

2. ПОСТРОЕНИЕ ФОТОННОЙ КАРТЫ

Для реализации задачи построения каустик выбраны язык программирования C++ и OpenGL API совместно с шейдерным языком GLSL. Так как практически все расчёты производятся на GPU, важно отметить, что все тесты производились на компьютере, оснащённом видеокартой NVIDIA GeForce 250 GTS. Визуализация сцены производится

методом обратной трассировки лучей, описанном в работе [1].

На каждом кадре генерируется набор лучей, как показано на рис. 3. Для получения фотонной карты данные лучи – “фотоны” света – трассируются от источника света через геометрию сцены. В данном случае, поскольку сцена задаётся аналитически и достаточно проста, ускоряющие структуры для нахождения пересечений не востребованы и были опущены.

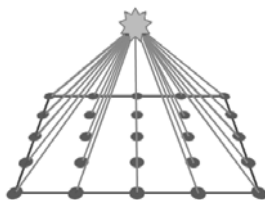


Рис. 3. Схема испускания фотонов

После трассировки конечное положение каждого фотона сохраняется в фотонной карте и может быть учтено при расчете цвета фрагмента на результирующем изображении.

Есть возможность значительно ускорить визуализацию сцены с каустиками по сравнению с полным перебором фотонной карты. Для этого нужно отсортировать фотонную карту так, чтобы уменьшить время доступа к ближайшим фотонам в окрестности точки соударения. Это можно сделать, используя пространственную хэш-функцию и битоническую сортировку (*bitonic sort*) [3]. Фотоны, непосредственно влияющие на фрагмент, при этом находятся бинарным поиском.

2.1 Производительность

Результат работы программы показан на рис 4. Все тесты для построенной программы проводились с разрешением фотонной карты 128x128, соответственно трассировалось 16384 фотона. По результатам тестов (см. рис. 5) видно, что частота кадров имеет примерно логарифмическую зависимость от общего числа обрабатываемых фрагментов и соответственно заметно убывает с увеличением выходного разрешения. При увеличении числа отражающих/преломляющих объектов частота кадров меняется не столь существенно, это объясняется тем, что наиболее затратными (по времени выполнения) этапами визуализации, являются этапы сортировки фотонной карты и работы с ней для конкретного фрагмента.



Рис. 4. Пример визуализации каустики для отражающего кольца методом Photon Mapping

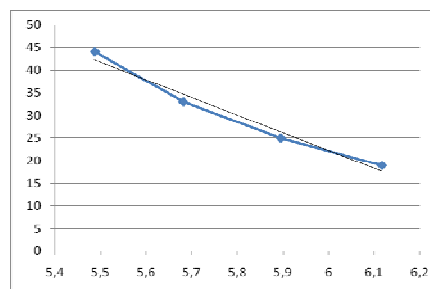


Рис. 5. График зависимости частоты кадров от $\log(N)$ где N - число фрагментов. Узловые точки соответствуют выходным разрешениям: 640x480, 800x600, 1024x768, 1280x1024

Доступная реализация техники Caustics Mapping [5] в разрешении 1024x768 выдаёт в среднем 70 кадров в секунду. В то время как наша программа дает 25 кадров в секунду, также обеспечивая реальное время. Другими словами Caustics Mapping (в данной реализации) не имеет однозначного преимущества.

3. ЗАКЛЮЧЕНИЕ

На данный момент обе технологии являются слишком ресурсоёмкими для использования в графических приложениях связанных с интерактивными развлечениями. Однако в скором времени ситуация может измениться. К тому же метод фотонных карт, как и трассировка лучей в целом, обеспечивает большую точность визуализации, и потому может быть предпочтён в ряде случаев, когда это имеет решающее значение.

4. ССЫЛКИ

- [1] Timothy J. Purcell , Ian Buck , William R. Mark , Pat Hanrahan. *Ray tracing on programmable graphics hardware* /SIGGRAPH'2002 (<http://graphics.stanford.edu/papers/rtongfx/>)
- [2] Musawir A. Shah, Jaakko Kontinen, S. N. Pattanaik, "Caustics Mapping: An Image-space Technique for Real-Time Caustics", IEEE Transactions on Visualization and Computer Graphics, March/April 2007, Vol 13(2), pp 272-280 (<http://graphics.cs.ucf.edu/caustics/final.pdf>)
- [3] Peter Kipfer. Chapter 46: Improved GPU Sorting/ GPU Gems 2. Apr 2005 (http://http.developer.nvidia.com/GPUGems2/gpugems2_chapter46.html)
- [4] Steven Parker, William Martin, Peter-Pike J. Sloan, Peter Shirley, Brian Smits, Charles Hansen. *Interactive Ray Tracing*. 1999 Symposium on Interactive 3D Graphics , April, 1999 <http://www.ppsloan.org/publications/rt99.pdf>
- [5] Резник С. Моделирование стеклянных поверхностей / Портал уральских разработчиков игр, июнь 2010. <http://www.uraldev.ru/articles/id/39>

Об авторе

Ульянов Данила Ярославович - студент Нижегородского Государственного Технического Университета им.П.Е.Алексеева, danila-ulyanov@narod.ru

Редактор для создания реалистичных физически корректных моделей эффектов для ландшафта

Александр Сергеевич Иванюк*, Алексей Игоревич Потапов, Алексей Александрович Щекин

*Белорусский государственный университет, Минск, Беларусь

Волгоградский государственный технический университет, Волгоград, Россия

{tride.a, waaarh, izgoj}@gmail.com

Аннотация

В работе обсуждается создание программной среды для моделирования физических эффектов виртуальной реальности на ландшафте с широким использованием графических процессоров (GPU) для обеспечения реального времени моделирования. Показан редактор для расчета динамики атмосферы в приповерхностном слое на основе уравнений Навье-Стокса.

Ключевые слова: GPGPU, виртуальная реальность, проектирование ландшафтов, моделирование газодинамики

1. ВВЕДЕНИЕ

Для компьютерных игр создаются целые миры, в которых виртуальные герои побеждают виртуальных драконов. Необходимость рассмотрения сцен проекта с различных точек наблюдения, оценки проекта в динамике, при перемещении наблюдателя в пространстве разрабатываемой архитектурной среды, возникает и на этапе согласования, и сдачи проекта заказчику, когда наличие возможности «пройтись» по будущему парку или скверу, оценить, каким он будет в то или иное время года или суток, как он будет выглядеть через несколько лет, позволяет более взвешенно подойти к оценке качества проекта[1]. Необходимо моделировать физику явлений подобную реальной (гравитация, свойства воды, столкновение с предметами и т. п.). Для создания убедительного комплекса ощущений реальности компьютерный синтез свойств и реакций виртуальной реальности должен производиться в реальном времени. Часто физика явлений очень сложна и рассчитывать её в реальном времени не получается, поэтому есть смысл использовать предварительные расчеты.

2. СУЩЕСТВУЮЩИЕ РЕШЕНИЯ

При большом разнообразии 3D-пакетов, лишь очень немногие из них имеют эффективные средства для работы с ландшафтами. Можно выделить пять основных генераторов ландшафта Bryce 3D, Terragen, MojoWorld, Vue d'Esprit, Vista Pro. Редакторы с разной степенью успешности справляются с генерацией фотореалистичного ландшафта. Слабым местом практически всех редакторов является анимация природных эффектов, таких как рябь на воде, колыбание травы и т.д. Например, в Bryce 3D есть анимация воды, дыма, погодных эффектов, но они недостаточно реалистичны. В Terragen [5] даже нельзя создать растения. В MojoWorld есть возможность

воздействовать на воду и деревья, но качество графики низкое.

В Vue d'Esprit [4] есть волны и анимация растительности, но очень примитивные. Vista Pro [3] анимацию растительности и воды не поддерживает. Таким образом, можно видеть, что есть потребность сделать редактор для физически корректных моделей явлений природы, таких как волны, ветер, дымка и тому подобное

3. ОПИСАНИЕ ТРЕБОВАНИЙ К НОВОМУ РЕДАКТОРУ

Итак, идея этой работы состоит в том, чтобы сделать редактор, конструирующий физически-корректные модели различных атмосферных явлений. Редактор должен позволять следующее:

1. редактировать ландшафт: детали рельефа, растительность, задавать физические условия для моделирования
2. в реальном времени просматривать результаты такового моделирования
3. позволять сохранять результаты в некотором виде, который сможет быть использован для воспроизведения в дальнейшем, не тратя вычислительные мощности на моделирование.

Таким образом, будут получены визуальные и физические эффекты, согласованные друг с другом.

4. ДЕТАЛИ РЕАЛИЗАЦИИ

Представление ландшафта

Ландшафт представляется картой высот. В геометрическом смысле рельеф представлен трехсторонними полигонами, соединяющими точки, являющиеся уровнями высот карты рельефа. Для оптимизации рельефа, производится разбиение карты на отдельные блоки. Число клеток блока должно быть равно $n=2m$, где $m \in \mathbb{O}[1, \dots]$, число вершин блока $k=2m+1$. Данные блоки растеризуются с отдельной текстурой и осуществляются при рендеринге отбрасывание, не входящих в видовой объем с использованием техники Quadrees.

Моделирование текущих сред

Одной и важных частей редактора является моделирование и визуальное представление текущих сред — воздуха (как движение ветра) и воды. Моделирование данных сред в редакторе производится на основе уравнения Навье-Стокса,

$$\frac{\partial u}{\partial t} = -(u \cdot \nabla)u + \nu \nabla^2 u + f \quad \text{представленного в следующей форме:}$$

$$\frac{\partial \rho}{\partial t} = -(u \cdot \nabla)\rho + k \nabla^2 \rho + S \quad (1)$$

где введены следующие обозначения: u – поле скоростей среды, ν - кинематическая вязкость, f – поле внешних сил, ρ – плотность среды. Следуя [8] уравнения (1), в частных производных, заменяются на системы разностных уравнений. При этом каждый раз для моделирования изменения поля при переходе от момента времени t_0 к моменту времени $t_0 + dt$ мы проходим через следующую последовательность шагов в соответствии со смыслом компонентов уравнения Навье-Стокса:

1. Изменяем поле скоростей текучей среды как результат диффузии, решая системы линейных уравнений вида

$$\bar{u}_{i,j} = u_{i,j} - a * (u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4 * u_{i,j}) \quad (2)$$
 где u – значение поля скоростей в момент времени t_0 , а \bar{u} – в момент времени t_0+dt
2. подвергаем поле скоростей изменению под действием компоненты, отвечающей за конвекцию среды;
3. производим специальную корректировку для того, чтобы в моделировании соблюдался закон сохранения массы.

Данный метод был выбран потому, что

1. он ориентирован на создание именно визуальных эффектов, а значит характеризуется устойчивостью;
2. при любом разумном шаге интегрирования и размере сетки результаты моделирования будут похожи на физически достоверные.

Именно поэтому на визуализируемой сцене может быть смоделировано огромное число эффектов, включая как те, которые были заранее просчитаны, так и те, которые требуют взаимодействия с подвижными объектами и потому должны моделироваться в реальном времени. Все эффекты будут выглядеть корректно, а для регулирования уровня загрузки CPU размер сетки для моделирования может динамически изменяться в зависимости от расстояния до камеры.

Была выполнена реализация алгоритма на CPU и OpenCL с распараллеливанием вычислений. Учитывая, что в процессе вычислений применяется неявный метод, распараллеливанию были подвергнуты лишь некоторые части численной модели. Результаты сравнения на компьютере с AMD Athlon x64 QL-65, видеокартой Radeon 4570, 2 GB Ram, Windows 7 x64 приведены в таблице.

Размер сетки \ тип реализации модели	CPU, fps	OpenCL, fps
10x10	290	285
30x30	220	200
70x70	60-63	75-80

Ниже приведены изображение, полученные в результате моделирования воздействия ветра на дым.

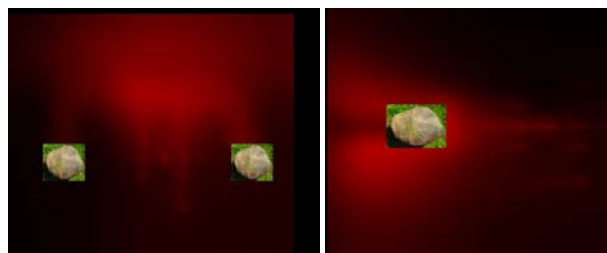


Рис.1. Огибание препятствий облаком дыма (красный). Ветер - снизу вверх (слева) и справа налево (справа).

Растительность и вода

В редактор планируется ввести возможность моделирования воздействия ветра на травянистую поверхность и модель движения воды.

5. ЗАКЛЮЧЕНИЕ

Для редактора произведен поиск, оценка и реализация метода моделирования текучих сред, выполнена начальная реализация ландшафта.

Планируется добавить возможность редактирования ландшафта, водных поверхностей и растительности, а также проводить моделирование воздействия воздушных потоков на них.

ЛИТЕРАТУРА

[1] Черников С.С. Среда для создания виртуальной реальности при разработке ландшафтного дизайна и малых архитектурных форм /Материалы конференции «Технологии Microsoft в теории и практике программирования», Нижний Новгород, 19–20 марта 2008 г.

[2] Animatek World Builder 3.0 Pro Review, - (electronic resource: <http://www.creative-3d.net/productFeatureDisplay.cfm>)

[3] Vistapro 4.1.1 Review, - (electronic resource: <http://www.creative-3d.net/vistapro.cfm>)

[4] Генераторы ландшафтов в графическом дизайне: обзор и области применения, - (electronic resource: <http://www.tugoff.ru/index.php>)

[5] “Краткий обзор программы Terragen” ; “Галерея компьютерной графики”, - [electronic resource] : <http://a4u.tomsk.ru/texts/files/pasha.0012/>

[6] Визуализация лесных массивов и рельефа местности в реальном времени, - [electronic resource] : <http://www.gamedev.ru/code/articles/?id=4211>

[7] Willem H. de Boer; Fast Terrain Rendering Using Geometrical MipMapping; E-mersion Project; October 2000

[8] Jos Stam. Real-time fluid dynamics for games; GDC 2003 proceedings.

Авторский указатель

Apalkov I.	122	Rukhlinskiy A.	45
Araújo B.	30	Safonov I.	79
Arutyunyan A.	324	Semeikina E.	133
Bilgili A.	57	Senyukova O.	92
Bokov M.	308	Shevtsov M.	45
Bucha V.	79	Shmaglit L.	122
Chirkov I.	111	Sorokin D.	137
Denisov A.	129	Stanković S.	116
Emelyanov A.	38	Tai, Wen-Kai	14
Figueiredo M.	30	Tan, Charlie Irawan	14
Frolov V.	49	Trukhina N.	111
Gudkov V.	87, 308	Tu, Shih-Chun	14
Huang, Yan-Zheng	72	Ushmaev O.	324
Ignatenko A.	49	Vaccarim A.	64
Kharlamov A.	49	Vashkelis V.	111
Khryashchev V.	122	Vetrov D.	92
Konushin A.	97, 100	Wu, Jyun-Ming	72
Konushin V.	97	Yakubenko A.	100
Krylov A.	129, 137, 320	Yurin D.	133
Krylov V.	107	Yusov E.	22
Kurt M.	57	Žarić N.	116
Lee, Chung-Nan	72	Zerubia J.	107
Letavin M.	45	Zhang-Jian, Da-Jing	72
Lukin A.	92		
Martino M.	64	Аминова Е.А.	335
Medvedeva V.	129	Архипов В.И.	163
Mizin I.	100	Бабаян П.	329
Oliveira J.	30	Барладян Б.Х.	148
Orović I.	116	Баша Н.	300
Öztürk A.	57	Белокаменская А.	350
Pavelyeva E.	320	Бобков А.	285
Pereira J.	30	Бобышева Т.	353
Ratti C.	64	Боголепов Д.	253
Rojas F.	64	Будков В.Ю.	207

Валиев И.В.	269	Матвеев З.А.	246
Веселов А.	355	Матвеев Ю.Н.	315
Визильтер Ю.	175	Местецкий Л.	222
Вилькин А.	339	Мекедов И.	230
Вознюк Т.	218	Поздняков С.Г.	155
Волобой А.Г.	155, 269	Потапов А.	360
Волохов В.	342	Потемин И.С.	144, 148
Востряков К.	142, 148	Потехин А.Л.	273
Гаврилов Н.	350	Прищепа М.В.	207
Галактионов В.А.	148	Птицын Н.В.	200
Гендин В.	344	Пустовойтов Д.	280
Голубев М.Н.	331	Разумков А.	333
Горбачевич В.	175	Ронжин А.Л.	207
Гущина О.Н.	335	Ростков Д.А.	292
Дебелов В.А.	238	Салимзибаров Р.	333
Долговесов Б.С.	257	Сергеев Е.	342
Егорова М.	339	Симакин И.С.	192
Ершов С.В.	155	Симончик К.К.	315
Жданов Д.	144, 148	Синдеев М.	214
Задонский Д.	230	Смирнов А.Г.	292
Иванюк А.	360	Смирнов С.	329
Казанский И.	285	Снежко Э.В.	163
Карпов А.А.	207	Соловьева А.	346
Кетков Ю.Л.	246	Сопин Д.	253
Кирьянов К.А.	188	Стержанов М.	169
Клименко С.	285, 292	Терещенко В.	218, 280
Клышинский Э.С.	155	Турлапов В.	253, 263
Ковалёв В.А.	163	Тухтасинов М.	304
Козлов Д.	263	Ульянов Д.	358
Козлов Д.С.	238	Хлебников Р.	337
Конущин А.	214, 348	Чигорин А.	348
Коробков А.	333	Шапиро Л.З.	148
Коростелев Е.И.	257	Шмаглит Л.А.	331
Кугаевских А.В.	181	Шульга Л.	300
Мазурок Б.С.	257	Щекин А.	360
Макарова Е.	230	Янчик Д.	280

