

The 22<sup>nd</sup> International Conference  
on Computer Graphics and Vision

**GraphiCon'2012**

---

October 01–05, 2012  
Moscow, Russia

## **Conference Proceedings**

22-я Международная Конференция  
по Компьютерной Графике и Зрению

**ГрафиКон'2012**

01–05 октября, 2012  
Москва, Россия

## **Труды Конференции**

Московский государственный университет имени М.В. Ломоносова



МОСКВА – 2012

## **Organizing Committee:**

Co-Chairs: Yevgeny Moiseyev (Lomonosov Moscow State University)  
Yuri Bayakovsky (Lomonosov Moscow State University)  
Andrey Krylov (Lomonosov Moscow State University)  
Vladimir Galaktionov (KIAM RAS)

## **Conference Organizing Committee**

Committee Coordinator	Stepan Vyaltsev (Higher School of Economics)
Financial Manager	Maxim Fedyukov (Lomonosov Moscow State University)
Web site	Pavel Batanov (Lomonosov Moscow State University)
Publishing/Printed Materials	Maxim Mizotin (Lomonosov Moscow State University) Andrey Nasonov (Lomonosov Moscow State University)

## **Young Scientist School Organizing Committee**

Co-Chairs: Yevgeny Moiseyev (Lomonosov Moscow State University)  
Alexey Voloboy (KIAM RAS)  
Andrey Krylov (Lomonosov Moscow State University)

## **Advisory Board**

Yuri Bayakovski (Lomonosov Moscow State University)	Stanislav Klimenko (Institute of Computing for Physics and Technology)
Valeriy Bobkov (Institute of Automation and Control Processes FEB RAS)	Anton Konushin (Lomonosov Moscow State University)
Victor Debelov (Novosibirsk State University)	Andrey Krylov (Lomonosov Moscow State University)
Vladimir Galaktionov (Keldysh Institute of Applied Mathematics RAS)	Timour Paltashev (National Research University ITMO)
Alexey Ignatenko (Lomonosov Moscow State University)	Vadim Turlapov (Lobachevsky State University of Nizhni Novgorod)
Denis Ivanov (Corporation "Russian Systems")	Dmitriy Vatolin (Lomonosov Moscow State University)

## **International Program Committee:**

Chair: Karol Myszkowski (Max-Planck-Institut für Informatik)  
Anton Konushin (Lomonosov Moscow State University)

Members:	Valery Adzhiev (UK) Boris Alpatov (Russia) Mark Bannatyne (USA) Olga Barinova (Russia) Boris Barladyan (Russia) Yuri Bayakovsky (Russia) Sergey Berezin (Russia) Valery Bobkov (Russia) Alexander Bondarev (Russia) Alex Bovyrin (Russia) Aleksey Chernavskiy (Russia) Vladimir Chernov (Russia) Dmitry Chetverikov (Hungary) Sabine Coquillart (France) Marc Daniel (France) Victor Debelov (Russia) Roman Durikovic (Slovakia) Viktor Eruhimov (Russia)	Victor Gaganov (Russia) Dan Gelb (USA) Amy Gooch (Canada) Vladimir Grishin (Russia) Mikhail Iakobovski (Russia) Andres Iglesias (Spain) Denis Ivanov (Russia) Joaquim Jorge (Portugal) Vladimir Kaleyda (Russia) Nikolay Kim (Russia) Anton Konushin (Russia) Andrey V. Kopylov (Russia) Vassili Kovalev (Belarus) Andrey Krylov (Russia) Victor Lempitsky (UK) Alexey Lukin (Russia) Tom Malzbender (USA) Maxim Mizotin (Russia) Timour Paltashev (Russia)	Alexander Pasko (UK) Victor Pilyugin (Russia) Dimitri Plemenos (France) Michael Rychagov (Russia) Ilya Safonov (Russia) Alexey Skvortsov (Russia) Philipp Slusallek (Germany) Sergey Sokolov (Russia) Vadim Turlapov (Russia) Oleg Ushmaev (Russia) Natalia Vassilieva (Russia) Dmitriy Vatolin (Russia) Dmitriy Vetrov (Russia) Yuri Vizilter (Russia) Aleksey Voloboy (Russia) Konstantin Vostryakov (Russia) Dmitriy Yurin (Russia) Evgeniy Zhilyakov (Russia)
----------	--	---	--

**Address: GraphiCon-2012, Graphics&Media Lab, 7<sup>th</sup> floor, 701 room, Graphics&Media Lab VMK (2-Gum corpus), Lomonosov Moscow State University, Leninskie Gory 1-52, Moscow, 119991, Russia**  
**Tel: +7-495-939-01-90 E-mail: [gc2012@graphicon.ru](mailto:gc2012@graphicon.ru) Web Site: <http://gc2012.graphicon.ru/>**



**The conference is organized by**  
**Lomonosov Moscow State University,**  
**Dept. of Computational Mathematics and Cybernetics,**



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



**Sponsored by Russian Foundation for Basic Research**



**Supported by Russian Academy of Science**

# Contents

Preface . . . . .	9
-------------------	---

## Technical section (English)

---

### EN1: Vision

Interactive camera distortion correction . . . . .	12
<i>B. Barladyan, L. Shapiro, I. Valiev, A. Voloboy</i>	
Pose refinement of transparent rigid objects with a stereo camera . . . . .	17
<i>I. Lysenkov, V. Eruhimov</i>	
Detection of swapped views in stereo image . . . . .	23
<i>A. Shestov, A. Voronov, D. Vatolin</i>	
Problem of auto-calibration in image mosaicing . . . . .	27
<i>A. Spizhevoy, V. Eruhimov</i>	
Monocular object localization by superquadrics curvature reprojec-tion and matching . . . . .	33
<i>E. Zappia, I. Afanasyev, N. Biasi, M. Tavernini, A. Fornaser, A. Selmo, M. De Cecco</i>	

### EN2: Graphics

Irradiance cache for a GPU ray tracer . . . . .	39
<i>V. Frolov, K. Vostryakov, A. Kharlamov, V. Galaktionov</i>	
Audio-aware on-the-fly animation from single photo . . . . .	45
<i>K. Kryzhanovsky, A. Vil'kin, I. Safonov, Z. Pushchina</i>	
Virtual reality technology for the visual perception study . . . . .	51
<i>G. Menshikova, Yu. Bayakovski, E. Luniakova, M. Pestun, D. Zakharkin</i>	
Adaptive 3D color anaglyph generation for printing . . . . .	55
<i>E. Patana, I. Safonov, M. Rychagov</i>	
GPU ray tracing — comparative study of ray-triangle intersection algorithms . . . . .	61
<i>V. Shumskiy, A. Parshin</i>	

### EN3: Medical image processing

Robotic aided surgery in full knee replacement . . . . .	67
<i>Liu Yuncai</i>	
Two approaches for noise filtering in 3D medical CT-images . . . . .	68
<i>M. Storozhilova, A. Lukin, D. Yurin, V. Sinitsyn</i>	
Border extraction of epidermises, derma and subcutaneous fat in high-frequency ultrasonography . . . . .	73
<i>A. Semashko, A. Yatchenko, A. Krylov, A. Bezugly, N. Makhneva, N. Potekaev</i>	

Deblurring in fundus images . . . . .	76
<i>A. Chernomorets, A. Nasonov</i>	

## EN4: Image processing

A hash based approach for secure keyless steganography in lossless RGB images . . . . .	80
<i>A. Chaudhary, J. Vasavada, J. Raheja, S. Kumar, M. Sharma</i>	
Cost-effective multiframe demosaicking based on bilateral filtering . . . . .	84
<i>K. Gorokhovskiy</i>	
Detection and correction of unnatural purple colors in images of refractive objects taken by digital still camera . . . . .	89
<i>M. Matrosov, A. Ignatenko, S. Sivovolenko</i>	
Learning graph Laplacian for image segmentation . . . . .	95
<i>S. Milyaev, O. Barinova</i>	
A new measure for distance-field based shape matching . . . . .	101
<i>P. Voronin, A. Adinetz, D. Vetrov</i>	

## Technical section (Russian)

---

### RU1: Biometry

Simile classifiers for face classification . . . . .	108
<i>V. Konushin, T. Lukina, A. Kuharenko, A. Konushin</i>	
Метод параллельных цепей для распознавания дактилоскопических изображений	113
<i>B. Гудков</i>	
Тополого-геометрическая модель изображения отпечатка пальца . . . . .	119
<i>B. Гудков, O. Ушмаев</i>	
Получение устойчивого криптографического ключа из биометрической характеристики изображения отпечатков пальцев . . . . .	125
<i>O. Ушмаев, B. Кузнецов</i>	
Метод проекционной фазовой корреляции в ключевых точках радужной оболочки глаза . . . . .	128
<i>E. Павельева</i>	
Модификация алгоритма классификации возраста людей по изображению лица на основе метода ранжирования . . . . .	133
<i>A. Рыбинцев, Т. Лукина, В. Конушин, А. Конушин</i>	

### RU2: Vision

Распознавание растровых текстурных изображений на основе параметрических ациклических марковских моделей . . . . .	138
<i>C. Двоенко, Динъ Шанг</i>	
Быстрый алгоритм обнаружения пешеходов по видеоданным . . . . .	144
<i>A. Казаков, А. Бовырин</i>	
Расширенный набор характеристик Хаара . . . . .	149
<i>Г. Юэсаков</i>	

Частичное обучение с учителем на небольших исходных выборках . . . . . 152  
*С. Завалишин, И. Сафонов*

О методах подсчета посетителей при зенитном расположении видеокамеры . . . 158  
*П. Харебов, Р. Салимзибаров*

## **RU3: Visualization and modeling**

Визуализация сейсмических данных на виртуальном глобусе . . . . . 163  
*А. Бобков, А. Леонов, В. Чебров*

Высококачественная объемная визуализации в реальном времени . . . . . 169  
*Д. Боголепов, И. Бугаев, Д. Сопин, Д. Ульянов, В. Турлапов*

Качественная объёмная визуализация гигавоксельных массивов в блочном представлении на примерах данных из медицины . . . . . 174  
*Н. Гаврилов, В. Турлапов*

Симуляция динамики жидкости и её взаимодействия с твёрдой поверхностью средствами графических ускорителей . . . . . 180  
*Ю. Хохлова, Д. Гладкий, С. Кузиковский, И. Белаго*

## **RU4: Scientific Visualization**

Analysis of unsteady space-time structures using the optimization problem solution and visualization methods . . . . . 184  
*A. Bondarev*

Scientific visualization and FRep geometric modeling: a survey . . . . . 189  
*A. Pasko, V. Adzhiev, E. Malikova, V. Pilyugin*

Разработка адаптивного мультиплатформенного визуализатора результатов научных расчётов для высокопроизводительных вычислительных систем . . . . . 193  
*К. Рябинин*

Визуализация потоков и обработка изображений: сближение и взаимодействие экспериментальных и численных данных . . . . . 199  
*И. Знаменская*

## **RU5: Images and vision**

Модификация воксельного метода для построения единой триангуляционной оболочки объекта по множеству видов . . . . . 200  
*В. Бобков, А. Кудряшов*

Меры искажения на основе энтропийных характеристик для анализа цифровых изображений . . . . . 204  
*В. Горбачев, И. Метелев, Е. Кайнарова, Е. Яковлева*

Устойчивая сегментация цифрового изображения . . . . . 208  
*М. Харинов*

Интегральные спиновые изображения в системах поиска трёхмерных моделей . . . . . 214  
*А. Крыловецкий, И. Черников*

Регуляризованный по Тихонову биспектральный вариационный метод обработки оптических изображений . . . . . 218  
*А. Потяголова, А. Ларичев, А. Разгулин, Н. Ирошников*

## RU6: Graphics

Система интерактивного расчета глобального освещения для гибридных сцен . . . . .	227
<i>Д. Боголепов, Д. Сопин, Д. Ульянов, В. Турлапов</i>	
Оптимизация определения видимых фрагментов текстуры для алгоритма виртуализации памяти графических ускорителей . . . . .	233
<i>А. Гордеев, Д. Гладкий, И. Белаго, С. Кузиковский</i>	
Оптимальное совмещение конечного множества точек с непрерывной по- верхностью в трёхмерном пространстве . . . . .	236
<i>А. Самойлов, Е. Попов</i>	
Интерактивное моделирование автомобильных красок . . . . .	242
<i>А. Волобой, С. Ершов, С. Поздняков</i>	
Спектральная трассировка лучей в задачах моделирования цветового сдви- га, возникающего в флюоресцентных средах . . . . .	248
<i>Д. Жданов</i>	

## Young Scientists School

---

An evaluation of finger alphabet intelligibility using quality assessment of video with masked content . . . . .	256
<i>P. Heribanova, Ja. Polec, D. Tarcsiova</i>	
Выделение объектов в видеопотоке на основе разрезов графов . . . . .	262
<i>А. Конушин, Н. Четвериков</i>	
Критерий перезахвата на ложный объект при корреляционном методе со- провождения . . . . .	266
<i>А. Ефимов, И. Козлитин</i>	
Ускорение расчёта вторичного освещения с помощью фильтрации в про- странстве экрана и уточнения на основе информации о близлежащей геометрии . . . . .	269
<i>А. Груздев, В. Фролов, А. Игнатенко</i>	
Применение нейронной сети для распознавания частных признаков дакти- лоскопических изображений . . . . .	273
<i>В. Гудков, К. Дорофеев</i>	
Параллельное программирование алгоритма идентификации дактилоско- пических изображений . . . . .	275
<i>В. Гудков, Д. Лепихова</i>	
Реализация методики улучшения и сегментации изображения темплета непре- рывнолитой заготовки . . . . .	278
<i>И. Мацко, О. Логунова, И. Порохов</i>	
Устойчивый метод выделения границ радужки в видимом диапазоне . . . . .	282
<i>И. Малин</i>	

Классификация автодорожных знаков на основе свёрточной нейросети, обу- ченной на синтетических данных . . . . .	284
<i>Б. Мусеев, А. Чигорин</i>	
Многокадровый оптический поток на основе траекторий . . . . .	288
<i>М. Синдеев, А. Конушин, К. Ротер</i>	
Реконструкция траектории движения и формы объекта с использованием инфракрасных маркеров и стереопары . . . . .	292
<i>Р. Зейналов, А. Конушин</i>	

# Preface

Dear Participant,

We would like to welcome you at GraphiCon'2012, a leading international conference on computer graphics, computer vision, image and video processing in Russia. The 22nd event is hosted by Lomonosov Moscow State University on October 1-5, 2012. This year we have a great program consisting of keynote presentations and scientific papers, carefully selected by International Program Committee, and a special poster section of of young scientist school. The International Program Committee was formed of 53 members representing 10 countries from all over the world. Being top experts in the respective areas, all of them have done a tremendous job reviewing on average 3 papers out of 80 submitted. We express thanks to the committee members, who served at a considerable personal sacrifice and with impressive collective wisdom. The final decision was based on at least two reviews of each manuscript and ended up with 48 works selected for oral presentation. Also 11 papers will be presented as posters at young scientist school. Keynote presentations are made by Marina Gavrilova (University of Calgary, Canada) and Dmitry Vatolin (Lomonosov Moscow State University, Russia). Keeping traditions, GraphiCon'2012 has not only scientific, but also extensive educational program, which is organized in cooperation with NVidia and devoted to computer graphics programming on modern hardware. We would like to thank our sponsors - Russian Foundation for Basic Research and Russian Academy of Sciences, volunteers organized by Graphics and Media Lab, Faculty of computational mathematics and cybernetics, Lomonosov Moscow State University, and everyone who made this event happen.

We do hope you will enjoy the conference,  
Anton Konushin, on behalf of GraphiCon'2012 Organizing Committee





## Technical section (English)

---

GraphiCon'2012

October 01–05, 2012  
Moscow, Russia

## Interactive Camera Distortion Correction

B. Kh. Barladyan, L.Z. Shapiro, I.V. Valiev, A.G. Voloboy  
Keldysh Institute of Applied Mathematics RAS, Moscow

### Abstract

Here interactive software and algorithm of camera distortion correction is considered. Elaboration of the camera distortion correction is needed during car parking system design. Specific algorithm representation as the set of six scale coefficients tables is introduced. This representation takes into account specific of cameras used in automobile industry and possibility of hardware implementation of given interactively created algorithms. The goal of the interactive software is to provide to car parking system designer a tool for elaboration of camera with desirable (reasonable) distortion. The scale coefficient tables are transferred to camera manufacturer for designed camera creation.

**Keywords:** parking camera design, camera distortion, distortion correction, lens design

### 1. INTRODUCTION

Using different cameras especially reversing cameras in the car parking system becomes almost standard in modern cars. There are a large number of manufacturers offering such system on the market [1, 2 and 3]. These cameras have wide-angle objective about 180 degree or even more to provide maximal visible area. Typically only horizontal extra wide angle is needed in parking systems. Moreover requirements to the visibility of the lower and upper hemispheres are different. Typically the visibility of the lower part of hemisphere is the most essential in parking systems from the driver point of view. So camera makers design cameras with asymmetrical view field. In common case view field may be asymmetrical in horizontal direction also. It is reasonable approach for cameras placed on the left and right bumper sides or on side rearview mirror. The cameras with such extra wide angles and asymmetrical view field unavoidable have large distortions, in common case asymmetrical ones, which should be corrected to better environment understanding by car driver. There are a number of algorithms about distortion correction [4-9]. Some of them are implemented in commercial software [10-12].

During design of parking system developers tune the camera position and orientation, taking into account the given car specific, select cameras with appropriate optical and electric specification. To provide maximally useful and effective overview of critical areas around the car the parking system designers would like to have custom distortion correction for used cameras. In general cases this correction can have own specific for each camera depending on the camera position. Some areas in the image should be magnified and other ones should be reduced.

The parking systems become currently a mass product, so the most reasonable and effective solution becomes embedding of distortion correction algorithm directly in the camera electronics. Camera makers can now implement almost any distortion correction algorithm in camera image processing but the algorithm for given specific camera should be elaborated by parking system designer and passed to the camera maker company in acceptable form. It should be pointed that in some practical cases camera distortions cannot be completely corrected in principle. Typical example is the camera with the view angle more than 180 degree. So the task of parking system designer is

elaboration of camera image in form optimized to control the car vicinity by car driver but not distortion correction itself. It is the main purpose of utility where suggested algorithm was implemented.

### 2. SCALE ALGORITHM REPRESENTATION

Taking into account possible asymmetric of camera distortions, the default image produced by camera is split from correction point of view on four parts by vertical and horizontal axes. The axes intersection point is the fixed point of distortion correction – the position of this pixel is not changed during correction. For rest pixels two type of scaling coefficients are defined. For vertical distortion the scale in given image point is defined as the function of X coordinate – ScaleY(x). These functions are defined separately for upper and lower sides (relatively to horizontal splitting axis). Addition scale function ScaleY(y) defines the scale coefficients as the function of coordinate y independently of x coordinate. Applying the scaling of the all these three functions does not move the pixels along horizontal splitting axis. The same three functions are defined for horizontal image scaling – ScaleX(y) for left and right image sides and ScaleX(x). Example of User Interface implemented in our software where these three functions in the table form can be set for image with resolution 800x600 is shown on Fig.1.

Vertical distortion		Horizontal distortion		ScaleX(x)	
<input checked="" type="checkbox"/> Scale(y)	<input type="checkbox"/> Smooth	<input checked="" type="checkbox"/> Scale(y)	<input type="checkbox"/> Smooth	<input checked="" type="checkbox"/> Scale(x)	<input type="checkbox"/> Smooth
Upper side		Lower side		Left side	
Pixel	Scale value	Pixel	Scale value	Pixel	Scale value
1	0.913001	1	1	1	1
2	0.577038	2	2.54046	2	1
3	0.671846	3	1.61382	3	
4	0.930835	4	1	4	
<input type="checkbox"/> Single table for both sides		<input type="checkbox"/> Single table for both sides		<input type="checkbox"/> Single table for both sides	
<input type="checkbox"/> Smooth		<input type="checkbox"/> Smooth		<input type="checkbox"/> Smooth	
Pixel		Pixel		Pixel	
Scale value		Scale value		Scale value	
1		1		1	
2		300		400	
3		600		800	
4					

Fig.1. Example of six scale tables.

Scale coefficients are defined for some set of pixels and are interpolated between them. By default linear interpolation is used. Apart linear interpolation the spline interpolation can be applied as well. Scale functions are displayed in the graphical form as it is showed on Fig. 2. Also it is possible to use the same scale function for both image sides (horizontal or vertical).

Interface, presented on Fig. 1, provides edition of all six scale functions. So, practically arbitrary distortion correction can be created in this way.

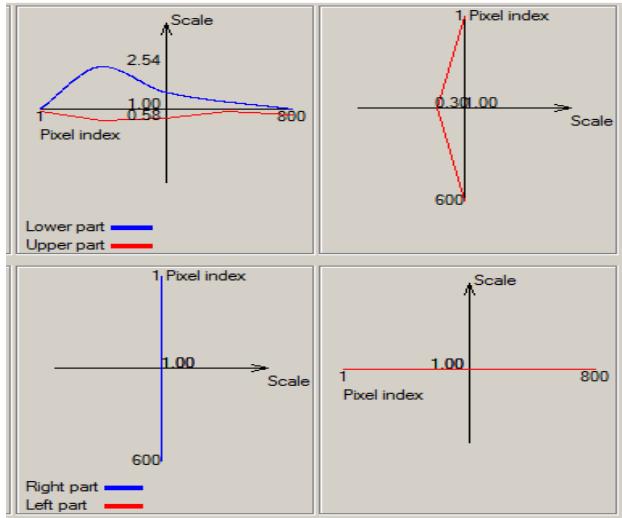


Fig.2. Scale function graphs.

### 3. DIRECT IMAGE CORRECTION

Specification of distortion correction in numerical form is not very convenient for parking camera system designer. It is more desirable to provide interface directly on image, so that user can move the image point from one position to another one, while the scale functions will be created for this correction automatically. This feature was provided in our software (Fig. 3).

In direct image correction mode the auxiliary grid provides convenient visual control how current distortion functions affect on original image. A designer can select color and step of this grid. Green contour provides presentation of original image size. Also it is possible to directly move selected pixel to the new position by mouse drag and drop. Application provides appropriate changes of scale tables and real time image correction. Additional slider provides convenient control of scale along X or Y axis according selected line in appropriate ScaleY(y) or ScaleX(x) table.

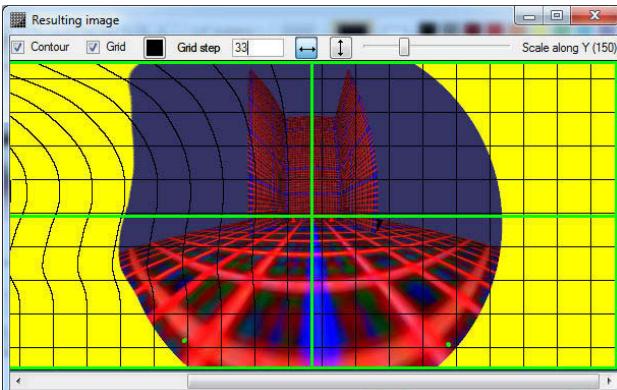


Fig.3. Direct image correction.

But this feature elaboration was rather complex because automatic creation of scale functions has various reasonable solutions. The image scale in horizontal and vertical directions can be considered independently due to selected correction algorithm representation, but scale in each direction in general case depends on two functions. In horizontal direction, for example it is ScaleX(y) and

ScaleX(x). It is hard to find reasonable and transparent for a user proportion between these two functions of given image pixel moving. Moreover the new X position of pixel is determined by integral of ScaleX(x) from zero (Y axis) till its initial X coordinate (see details of algorithm below). So the movement of pixel can be achieved using different ScaleX(x) function. Only corresponding integral is essential. Due to this reason the direct image correction in our system affects on ScaleX(y) and ScaleY(x) functions only. Even with this restriction the task is not trivial as it is described in algorithms details below.

### 4. CORRECTION ALGORITHM DETAILS

#### 4.1 Smoothing algorithm

The third order polynomial was used for smooth interpolation of scale coefficients between points of definition. To provide really smooth interpolation it is desirable that extreme points (maximal/minimal scales) in linear interpolation are to be extreme points of spline interpolation. In another words the interpolation should be monotony between node points. This requirement is provided in described below algorithm by special definition of derivations in knot points.

The segment of the original polyline between all the pairs of "knots"  $(x_i, y_i)$  and  $(x_{i+1}, y_{i+1})$  is smoothed by the third order polynomial  $q_i(x)$ , where

$$q_i(x_{i+1}) = q_{i+1}(x_{i+1}) = y_{i+1}$$

$$q_i'(x_{i+1}) = q_{i+1}'(x_{i+1})$$

The third order polynomial  $q(x)$  for which

$$q(x_1) = y_1$$

$$q(x_2) = y_2$$

$$q'(x_1) = k_1$$

$$q'(x_2) = k_2$$

can be written in symmetrical form

$$q = (1-t) \cdot y_1 + t \cdot y_2 + t \cdot (1-t) \cdot (a(1-t) + b t)$$

where

$$t = (x - x_1) / (x_2 - x_1)$$

$$a = k_1 \cdot (x_2 - x_1) - (y_2 - y_1)$$

$$b = -k_2 \cdot (x_2 - x_1) + (y_2 - y_1)$$

The derivatives in the knots are defined as followed:

1. For the first knot:

$$k_1 = (y_2 - y_1) / (x_2 - x_1) \quad (1)$$

2. For the last knot:

$$k_n = (y_n - y_{n-1}) / (x_n - x_{n-1})$$

For intermediate knots:

$$k_i = 0 \text{ if } y_i \leq y_{i-1} \text{ and } y_i \leq y_{i+1} \quad (2.1)$$

$$k_i = 0 \text{ if } y_i \geq y_{i-1} \text{ and } y_i \geq y_{i+1} \quad (2.2)$$

$$k_i = (y_{i+1} - y_{i-1}) / (x_{i+1} - x_{i-1}) \text{ for all other cases.} \quad (2.3)$$

The derivations definition 2.1 and 2.2 provides that extreme points of linear interpolation remain extreme one for smooth interpolation also.

#### 4.2 Output image smoothing

The scale tables described above describe transformation of any point from original image to the corrected one, but both images are discretized ones, i.e. consist of pixels. So we have some freedom how pixels of output image will be constructed from original ones. The described below algorithm was constructed to provide relatively smooth image and minimize moiré and aliasing effects.

In general case the original image pixel will change its form in output image. It will be scaled depending on its X and Y coordinates. Scale of any pixel which don't have its own value in the table is calculated as a linear or smooth (see p. 4.1) interpolation between the nearest lower and higher values.

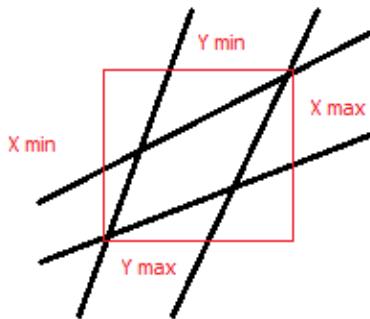


Fig.4. Scaling a pixel

Calculated coordinates of each pixel vertex after scaling and its bounding box drawn by red color are shown on Fig.4. We consider this bounding box as scaled original pixel which color should be put to the all pixels of new (output) image, which it covers as it is shown on Fig. 5.

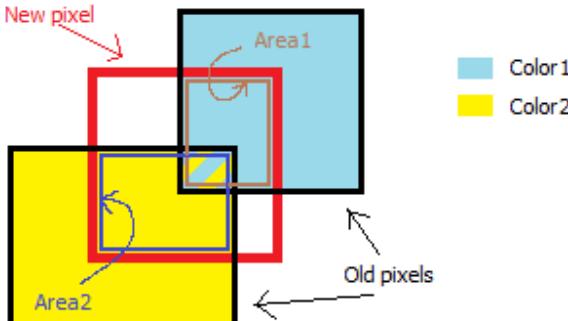


Fig.5. Color setting for a new image pixel.

It should be taken into account that scaled pixels (initial, "old" pixels on the Fig.5) can have overlapping due to extension by bounding box described above. Finally, the color of output is calculated as the weighted sum of initial pixels colors with weight of intersection area of old pixel with new one.

The result of image scaling for some test example is shown on Fig.6.

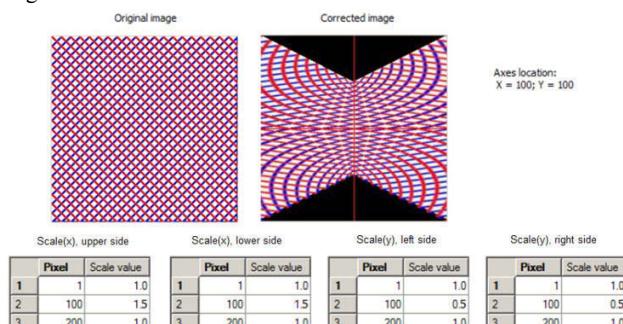


Fig. 6. Result of image scaling.

## 5. DIRECT IMAGE CORRECTION ALGORITHM

The procedure assumes that the user will move any pixels from one position (start position) to another one (end position). And the algorithm should calculate such scale transformations so that original pixel (correspondent pixel of non-scaled picture) will be moved to the selected end position. This task includes two sub-tasks:

- determination of initial pixel position by its start position on the current scaled image;
- calculation of new scale table described in p.2. (or correction of the current one) which moves pixel from initial position to the end one.

### 5.1 Calculation of initial pixel

The new (transformed) pixel coordinates ( $x_n, y_n$ ) are calculated from the original ( $x_o, y_o$ ) ones by the following formulae:

$$x_n(x_o, y_o) = x_a + (x_o - x_a) \cdot S_{xy}(y_o) \cdot \int_{x_a}^{x_o} S_{xx}(x) dx \quad (3)$$

$$y_n(x_o, y_o) = y_a + (y_o - y_a) \cdot S_{yx}(x_o) \cdot \int_{y_a}^{y_o} S_{yy}(y) dy \quad (4)$$

Here:

- $x_o, y_o$  are original pixel coordinates;
- $x_n, y_n$  are transformed (scaled) pixel coordinates;
- $S_{xy}, S_{xx}, S_{yx}, S_{yy}$  are scale functions defined in p.2;
- $x_a$  – Y axis position
- $y_a$  – X axis position

#### 5.1.1 Area subdivision and scales description.

The scale functions definition in table form described in p.2 subdivides the initial pixel image into rectangular cells. The horizontal bounds of these cells will be y-lines passed through the pixels where scale functions  $S_{xy}(y)$  (left and right) and  $S_{yy}(y)$  are defined. The vertical bounds of these cells will be x-lines passed through the pixels where scale functions  $S_{yx}(x)$  (upper and lower) and  $S_{xx}(x)$  are defined.

Let the bounds of i-th rectangle are  $x_{mini}, x_{maxi}, y_{mini}$  and  $y_{maxi}$ . Inside this rectangle we can represent integrals in (3) and (4) as:

$$\int_{x_a}^{x_o} S_{xx}(x) dx = \int_{x_a}^{x_{mini}} S_{xx}(x) dx + \int_{x_{mini}}^{x_o} S_{xx}(x) dx \quad (5)$$

$$\int_{y_a}^{y_o} S_{yy}(y) dy = \int_{y_a}^{y_{mini}} S_{yy}(y) dy + \int_{y_{mini}}^{y_o} S_{yy}(y) dy \quad (6)$$

Taking into account that the  $S_{xy}(y_o)$  is constant for given  $y_o$  and the  $S_{yx}(x_0)$  is constant for given  $x_0$  the scale transformation of initial pixel  $(x_o, y_o)$  to the scaled one  $(x_n, y_n)$  has the two following properties:

#### Property 1.

If  $y_o = \text{const}$  and  $x_{o1} < x_{o2}$  then  $x_{n1} < x_{n2}$ . If  $x_o = \text{const}$  and  $x_{o1} > x_{o2}$  then  $x_{n1} > x_{n2}$ . So in this case  $x_n(x_o)$  is monotone increasing function. It follows from (3) and (5).

#### Property 2.

If  $x_o = \text{const}$  and  $y_{o1} < y_{o2}$  then  $y_{n1} < y_{n2}$ . If  $y_o = \text{const}$  and  $y_{o1} > y_{o2}$  then  $y_{n1} > y_{n2}$ . So in this case  $y_n(y_o)$  is monotone increasing function. It follows from (4) and (6).

From properties 1 and 2 follow that the maximal and minimal values of scaled coordinates  $x_n$  and  $y_n$  will be achieved on the scaled boundaries of original rectangular boundaries.

Moreover the maximum and minimum of  $x_n$  will be achieved on the rectangle vertices. If we set  $x_o = x_{mini}$  then from (3) and (5) we have:

$$x_n(x_{\min}, y_0) = x_a + K_1 \cdot S_{xy}(y_0) \quad (7)$$

where

$$K_1 = (x_{\min} - x_a) \cdot \int_{x_a}^{x_{\min}} S_{xx}(x) dx,$$

and the  $S_{xy}(y_0)$  is monotone function of  $y_0$  inside the given cell (see p.4.1). So the minimum and maximum of (7) will be achieved on the segment ends. The same statement is true for  $x_0 = x_{\max}$  and finally for  $y_n$ .

Taking into account all these properties of scale transformation we implement the following algorithm for calculation initial  $(x_0, y_0)$  pixel from  $(x_n, y_n)$  scaled one:

### 5.1.2 Initial pixel finding algorithm.

1. Create the list of bounding boxes of scaled rectangular cells described above. Transformed original rectangular cells will be curved one in general case (see Fig. 3, for example) and so bounding boxes of scaled rectangular cells will be overlapped. Due to this reason the given scaled pixel can belong to the several bounding boxes simultaneously.
2. For each cell from list try to determine initial pixel by the following way:
3. If the given bounding box does not include  $(x_n, y_n)$  then go to the next one.
4. Calculate scaled  $(x_{n1}, y_{n1})$  pixel for center of original cell. If the distance from  $(x_{n1}, y_{n1})$  pixel to the  $(x_n, y_n)$  one is lesser than 1 then original pixel is found,  $(x_n, y_n)$  is center given original cell.
5. If width and height of original box is lesser than 1 then original pixel can not be found in the given cell. Go to the next bounding box.
6. Divide given original box on two ones by division of width or height in half. For each half calculate box of scaled cell. For each half cells execute pp. 3-6.

In the result we have calculated the initial pixel position.

## 5.2 Scale table correction

As it was pointed above the correction will not touch the  $S_{xx}(x)$  and  $S_{yy}(y)$  functions and so the correction will be done for  $S_{xy}(y)$  and  $S_{yx}(x)$  only. Corrections for  $S_{xy}(y)$  and  $S_{yx}(x)$  can be applied independently. So consider the  $S_{yx}(x)$  only. Firstly consider only the linear interpolation between node points. One scale segment for  $S_{yx}(x)$  is represented on the Fig. 7.

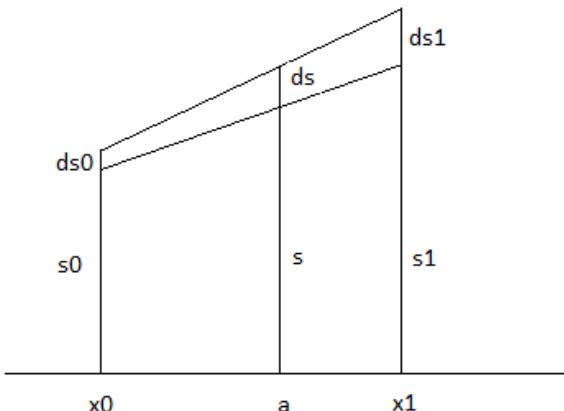


Fig. 7. Segment of scale function.

Here the movement pixel along  $x = a$  coordinate is considered. The point  $a$  belongs to the  $(x_0, x_1)$  segment. Let us denote the scale values and its variation at the ends of the segment as  $s_0, s_1$ ,

$ds_0$  and  $ds_1$  correspondingly. The scale and scale variation in the point  $a$  denote as  $s$  and  $ds$ . We want to determine correspondent changes  $ds_0$  and  $ds_1$  which provide  $ds$  changing in the point  $a$  in the linear interpolation case.

For simplicity, we will use the dimensionless coordinates.

Let  $x_0 = 0$  and  $x_1 = 1$ ,  $a_1 = (a - x_0) / (x_1 - x_0)$ .

Due to the linear interpolation we have:

$$\begin{aligned} s_0 \cdot (1-a) + s_1 \cdot a &= s \\ (s_0 + ds_0) \cdot (1-a) + (s_1 + ds_1) \cdot a &= s + ds \end{aligned}$$

So

$$ds_0 \cdot (1-a) + ds_1 \cdot a = ds \quad (8)$$

We should determine  $ds_0$  and  $ds_1$  via  $ds$  and  $a$ . Let us will find solution in form:

$$ds_0 = ds \cdot f_0(a); \quad ds_1 = ds \cdot f_1(a). \quad (9)$$

From (8) and (9) we have

$$f_0(a) \cdot (1-a) + f_1(a) \cdot a = 1; \quad (10)$$

From (9) and (10) we have the following boundary conditions:

$$f_0(0) = 1; \quad f_1(1) = 1.$$

$$f_0(1) = 0; \quad f_1(0) = 0.$$

We also naturally suppose functions symmetry:

$$f_0(a) = f_1(1-a) \quad (11)$$

From (10) and (11) also followed that

$$f_0(0.5) = f_1(0.5) = 1$$

So finally we have the following conditions

$$\begin{cases} f_0(0) = 1; \quad f_0(0.5) = 1; \quad f_0(1) = 0; \\ f_1(0) = 0; \quad f_1(0.5) = 1; \quad f_1(1) = 1; \end{cases} \quad (12)$$

There are many functions which satisfy to these conditions. For example we can determine  $f_0(a)$  on  $[0.5, 1.0]$  as any function decreasing from 1.0 to 0.0. In this case  $f_1(a)$  will be determined on  $[0, 0.5]$  by (11). Then we can determine  $f_0(a)$  on  $[0, 0.5]$  by substituting  $f_1(a)$  in equation (10) and solving it for  $f_0(a)$ . We consider the following two variants of solution:

### 5.2.1 1st variant of solution.

Let

$$f_0(a) = 2 \cdot (1-a) \text{ on } [0.5, 1] \quad (13)$$

and so from (11)

$$f_1(a) = 2 \cdot a \text{ on } [0, 0.5] \quad (14)$$

From (10) and (13) we have

$$f_1(a) = (1 - 2 \cdot (1-a)^2) / a \text{ on } [0.5, 1]$$

and from (10) and (14) we have

$$f_0(a) = (1 - 2 \cdot a^2) / (1-a) \text{ on } [0, 0.5]$$

And finally:

$$f_0(a) = \begin{cases} (1 - 2 \cdot a^2) / (1-a), & \text{on } [0, 0.5] \\ 2 \cdot (1-a), & \text{on } [0.5, 1] \end{cases}$$

$$f_1(a) = \begin{cases} 2 \cdot a, & \text{on } [0, 0.5] \\ (1 - 2 \cdot (1-a)^2) / a, & \text{on } [0.5, 1] \end{cases}$$

Both functions has the same maximum  $\sim 1.172$  in points  $1 - \sqrt{0.5}$  and  $\sqrt{0.5}$  appropriately.

### 5.2.2 2nd variant of solution.

Let us will find the solution in the parabola form  $f_0(a) = k_2 \cdot a^2 + k_1 \cdot a + k_0$  on  $[0, 1]$ . According (12) this parabola should pass through the points  $(0, 1)$ ,  $(0.5, 1)$  and  $(1, 0)$ . These restrictions completely define parabola coefficients:

$$f_0(a) = -2 \cdot a^2 + a + 1 \text{ on } [0, 1]$$

From (11)

$$f_1(a) = -2 \cdot a^2 + 3 \cdot a \text{ on } [0, 1]$$

$f_0(a)$  decreases from 1.0 to 0.0 on [0.5, 1.0]

It is easy to check that  $f_0$  and  $f_1$  satisfy the (10) equation.

Both functions has the same maximum 1.125 in the points 0.25 and 0.75 appropriately.

The less the function maximum the less will be scale values variations in the nodes and so the resulted curves will be more smooth. So we selected parabola for  $f_0()$  and  $f_1()$  functions.

### 5.2.3 Spline case.

Described above in p.5.2.2 solution works in linear interpolation case only. In spline case the task become nonlinear one and can not be solved analitically. In this case the solution from p.5.2.2 is used as initial approximation. Let us linear interpolation for given  $ds$  gives  $(ds_0, ds_1)$  solution. Then the non linear solution  $(ds_0^n, ds_1^n)$  we will find by using the following equation:

$$ds_1^n = K \cdot ds_0^n, \text{ where } K = ds_1 / ds_0; \quad (15)$$

Let us denote the function which calculate the scale variation in point  $a$  by using spline interpolation between scale table nodes as  $F_{spl}(x)$ , where  $x$  is scale variation in left segment point and the scale variation in the right segment point is defined by (15). Then  $ds_0^n$  can be found by solving the following non linear equation:

$$F_{spl}(ds_0^n) = ds$$

We solve this equation by founding the solution inside segment. The one boundary is defined by linear interpolation approximation and for the second one is the maximal acceptable scale if  $F_{spl}(ds_0) > ds$  and minimal one in oposite case.

## 6. RESULTS

Described interactive software was implemented as additional application (plugin) for CATIA [13, 14] CAD/PDM system. It can process both with images produced by real cameras and with ones simulated in CATIA by our products [14]. The plugin provides real time design of distortion correction algorithm in form of both scale tables and interactive resulting image modification. Corrected image is re-drawn during fraction of a second after parameters changing (Intel Core 2 Q9550 2.83Ghz).

The scale tables described above contains distortion correction information in a form acceptable for camera creation by the camera manufacturer. And implemented algorithms take into account parking system cameras manufacturing specific. In the result the implemented software provides effective and convenient tool for car park system designers to develop reasonable correction of camera images.

## 7. AKNOLEDGMENTS

This work was supported by the RFBR, grant10-01-00302, and by INTEGRA Inc. (Japan).

## 8. REFERENCES

- [1] <http://www.parkingcameras.com/store/home.php>
- [2] <http://www.espow.com/wholesale-car-electronics-car-review-systems-rear-view-cameras.html>
- [3] <http://www.thecarkitcompany.com.au/index.php/products/parking-sensors-a-reverse-cameras>
- [4] F. Devernay and O. Faugeras. Automatic calibration and removal of distortion from scenes of structured environments. SPIE Conference on investigative and trial image processing San Diego, CA, 1995.
- [5] H. Farid and A.C. Popescu. Blind removal of Lens Distortion. Journal of the Optical Society of America, 2001.
- [6] J. Jedlička, M. Potůčková. Correction of Radial Distortion in Digital Images. Charles University in Prague Faculty of Science, [http://dsp.vscht.cz/konference\\_matlab/MATLAB07/prispevky/jedlicka\\_potuckova/jedlicka\\_potuckova.pdf](http://dsp.vscht.cz/konference_matlab/MATLAB07/prispevky/jedlicka_potuckova/jedlicka_potuckova.pdf)
- [7] Janez Perš, Stanislav Kovačič. Model-Based Radial Lens Distortion Correction Using Tilted Camera Assumption. Faculty of Electrical Engineering University of Ljubljana Nonparametric, <http://vision.fe.uni-lj.si/docs/janezp/pers-wwk2002.pdf>
- [8] R. Swaminatha and S.K. Nayer. Non-metric calibration of wide angle lenses and poly-cameras. IEEE Conference on computer Vision and pattern recognition, pp 413, 1999.
- [9] G. Taubin. Camera model for triangulation. Lecture notes EE-148, 3D Photography, Caltech, 2001.
- [10] PTLens, <http://epaperpress.com/ptlens/>
- [11] Photoshop, Correcting image distortion, <http://helpx.adobe.com/photoshop/using/correcting-image-distortion-noise.html>
- [12] IRIS TUTORIAL, DSLR images distortion correction, [http://www.astrosurf.com/buil/iris/tutorial119/doc42\\_us.htm](http://www.astrosurf.com/buil/iris/tutorial119/doc42_us.htm)
- [13] CATIA - Virtual Design for Product Excellence, <http://www.3ds.com/products/catia/welcome/>
- [14] Inspirer, Specter optical simulation system <http://www.integra.jp/en/index.html>

## About the authors

Boris Kh. Barladyan, PhD, senior researcher, Keldysh Institute for Applied Mathematics RAS.

E-mail: [obb@gin.keldysh.ru](mailto:obb@gin.keldysh.ru)

Lev Z. Shapiro, PhD, senior researcher, Keldysh Institute for Applied Mathematics RAS.

E-mail: [pls@gin.keldysh.ru](mailto:pls@gin.keldysh.ru)

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](mailto:voloboy@gin.keldysh.ru)

# Pose Refinement of Transparent Rigid Objects With a Stereo Camera

Ilya Lysenkov

Itseez

ilya.lysenkov@itseez.com

Victor Eruhimov

Itseez

victor.eruhimov@itseez.com

## Abstract

We propose a new method for refining 6-DOF pose of rigid transparent objects. The algorithm is based on minimizing the distance between edges in a test image and a set of edges produced by the training model with a specific pose. The model is scanned with a monocular camera and a 3D sensor such as a Kinect device. The pose is estimated from a monocular image or a stereo pair. The method does not require a CAD model of the object. We demonstrate experimental results on a set of kitchen items essential for any home and office environment.

**Keywords:** *pose estimation, localization, transparent objects.*

## 1. INTRODUCTION

Perception for personal robotics is a wide and important application of computer vision. A personal robot is expected to efficiently interact with the environment. In particular, it has to be able to detect a specific object in a scene and find its pose for grasping and manipulation. Recent advances in object recognition and pose estimation [1] demonstrate good results with a monocular camera for textured objects. SIFT features are used to find similarities between training and test textured image patches and then geometric validation is used to filter out false matches. Since the training set contains 3D coordinates of all features, pose estimation in this approach is done by solving a PnP problem on SIFT matches. However if an object has few textured features, local descriptors will produce few matches and detection will fail. Moreover, if only a small part of the object is textured, it will be detected but there may be a substantial error in the pose estimation. Also, this method does not work with transparent objects.

Both textureless and transparent objects such as cups, dishes, staplers etc. are an essential part of home and office environment. The problem of estimating the pose of such objects is important for personal robotics. While recent developments in structured light sensors such as Kinect shows promising results in finding the pose of textureless objects, this type of technology does not work with specular and transparent surfaces. Our work in this paper is largely influenced by the methods for textureless objects coming from industrial robotics [2] that use a CAD model of an object to estimate its pose from a monocular camera by projecting the model to a test image and comparing object features with image edges. While CAD models of manipulated objects in industrial settings are usually available anyway, CAD models of all objects in the personal space are hard to capture.

We present an algorithm for refining 6-DOF pose of a transparent object using edge features. The method does not require a CAD-model, it needs a 3D scan of an object including a point cloud and images registered to each other. We show that the method can be used for accurate pose estimation of transparent rigid objects.

## 2. RELATED WORK

Transparent objects are very challenging objects in computer vision because their appearance in an image largely depends on a background. Also it is hard to capture a 3D model or a point cloud for transparent objects due to limitations in technologies of existing 3D sensors and because reconstruction of transparent objects is still a very hard problem [3].

The algorithm for detection and reconstruction of unknown trans-

parent objects was proposed in [4]. The algorithm uses two views of a test scene captured by a ToF camera. The algorithm is insensitive to changes in illumination and it was applied for grasping of isolated transparent objects by a robot. Grasping was successful in 41% of reconstructed objects and failed attempts are explained by errors in objects reconstruction and pose estimation.

The algorithm for pose estimation of transparent objects from two views of a test scene was proposed in [5]. Accurate pose estimation was achieved but the objects are required to stay on a table plane and they should be separated from each other. So the algorithm is not able to estimate 6-DOF pose.

Kinect sensor is used for pose estimation and recognition of transparent objects in [6]. However, results are reported only in case when objects are assumed to stay on a table plane. So accuracy in case of 6-DOF pose estimation is unclear.

Specularities are important features when working with transparent objects and there are very promising approaches to pose estimation [7, 8, 9] using this cue. However, these algorithms of pose estimation require a triangulated mesh or a CAD model of an object and they were evaluated with textureless objects only.

Texture features like SIFT are not suitable when working with textureless and transparent objects because such objects don't have their own texture. Computer vision research [10] and psychological studies [11] show that edges and contours of objects are important features and they can be used successfully for the object recognition problem. For example, humans can recognize objects from rough pencil sketches although texture is missing. This cue is available both for transparent and textureless objects and it makes the problem of pose estimation of transparent objects related to pose estimation of textureless objects.

The problem of untextured pose estimation has a long history in computer vision. See [12] for a detailed overview of the 2D-3D pose estimation problem. [13] shows that it is possible to estimate a pose of a textureless object by using single-view object detection algorithms. However the 2D object representation used in this method is viewpoint-dependent, so a set of detectors has to be trained for different viewpoints. Running all detectors is infeasible in the general case so pose clustering is used [14, 13, 15], first to make a rough estimation of the pose and then refine it by running a smaller set of detectors. The pose corresponding to the most confident detector is returned as an estimation of the object pose. But the accuracy of this estimation is bounded by the number of detectors that also defines the computational cost.

General multi-view approaches and a 3D model of an object are required to balance between the computational cost and the pose estimation accuracy. The idea to use a 3D representation of an object for recognition is going back to early computer vision of 70's and 80's, see, for example, [16]. Approaches [17, 2, 18] utilize this idea and they can estimate a pose of a textureless object quite accurately. Algorithms [17, 18] find the closest training pose and run a local optimization of it using a CAD model of an object. High-quality CAD-models are hard to obtain and although there are some CAD-models of typical household objects (like a cup or a bottle), models are not available for all specific objects that robots need to grasp in a household environment.

Our approach to pose refinement step is similar to [17, 18] and also based on edges cue. However, it does not require a CAD model and it is able to estimate 6-DOF pose of transparent objects.

### 3. PROPOSED APPROACH

To solve the considered problem we divide it to following tasks:

1. Create a 3D model which allows to generate object edgels (points on edges) for different poses. Our model contains a 3D object model and a 3D edge model. The 3D object model is a point cloud of the whole object and it is used to generate silhouette edges. The 3D edge model is a point cloud with points on surface edges, that is edges created by depth discontinuities or texture.
2. Determine a cost function which estimates dissimilarity between generated edgels and the observed test data and then minimize the cost function by varying parameters that determine pose of the object.

We will address all of these steps in the following subsections.

#### 3.1 Creation of the 3D model

There are no stable ways to estimate depth or produce point clouds for transparent objects [3]. So we take a copy of the object, paint it with a color and use the painted object in the model creation pipeline.

The 3D object model is created automatically from the train data. We scan each object on a planar surface with a Kinect device. Two fiducial markers consisting of grids of circles are placed in the field of view to provide accurate registration of frames. Depth map from Kinect allows us to segment the plane and calculate the object mask in each image.

We illustrate the algorithm of the surface edge model creation using a textureless object that has many surface edges (Fig. 1). First, we extract 3D points that correspond to surface edges in each frame, then we register point clouds from different frames, and, finally, we build a surface edge model.

##### Detecting edges in each frame

1. Find edges on each image of the object using Canny edge detector. Then find edges of the object by intersecting the detected edges with the object mask.
2. Select the points from the 3D cloud that correspond to image edges. Our point cloud is interpolated to the size of the train image and so there is a bijection between 3D points and image pixels. As a result we get a 3D edge model for each training image.

##### Registering point clouds

1. Transform all models to the same coordinate system associated with the first frame, using the poses from the fiducial markers. The corresponding points from different frames would coincide with each other in the ideal case but there are always some deviations in practice due to noise (see the Fig. 2A).
2. Register transformed point clouds. There is a classic and widely used algorithm Iterative Closest Point (ICP) for registration of two point clouds [19, 20]. Global approaches like [21] are used for registration of multiple point clouds because they can distribute registration error between all point clouds evenly. We have a good initial alignment of point clouds using the poses from the fiducial markers so we have used more simple global algorithm [22] with LM-ICP [23] to register pairs of point clouds.

##### Creating a surface edge model

1. Partition all transformed points into groups where each group corresponds to the same point of the object. This

allows to get more accurate coordinates of the object point by its noised observations. Partitioning is done by solving the problem of k-partite matching which is a generalization of the bipartite matching for the case of k-partite graphs. It is known to be an NP-hard problem [24] so we used a heuristic algorithm based on [25].

2. For each group compute accurate coordinates of the model point using robust estimation of location [26] e.g. the minimum covariance determinant estimator (MCD) [27]. The constructed model is given in the Fig. 2B and it represents edges of the object much better than transformed point clouds in Fig. 2A.
3. Downsample the constructed 3D edge model. The 3D edge model of the whole object is given in the Fig. 2C. It contains many close points that don't give additional information. So we keep 10% of points to lower computational costs of further processing. It is done by a trivial adaptation of the Douglas-Peucker algorithm [28] for this task. The downsampled model is given in the Fig. 2D.

It is important to note that as a result of the k-partite matching the silhouette edges, which presence depends on a point of view, will be automatically filtered out as they will not have correspondences in different frames.

We group all surface edge points into contours by proximity. 3D orientation of a contour at a point can be estimated as direction of the tangent vector to the 3D contour at this point. We do this by generalizing [29] to the 3D case by means of multi-dimensional robust statistics [26]. When the model is transformed in 3D space, orientations of points are transformed as usual 3D points. We use the contours to calculate the orientation in each projected edgel that can be used in the cost function.

The algorithm for constructing a silhouette model is similar. We register dense point clouds that we obtain from a Kinect by using the same algorithm (ICP registration with the initial pose from the fiducial markers). The coordinate system origin is placed into the mass center of the joint point cloud.

In order to guarantee that poses that are close to each other are produced by close rotation and translation vectors, we place the coordinate system origin into the center of mass for each of the objects.

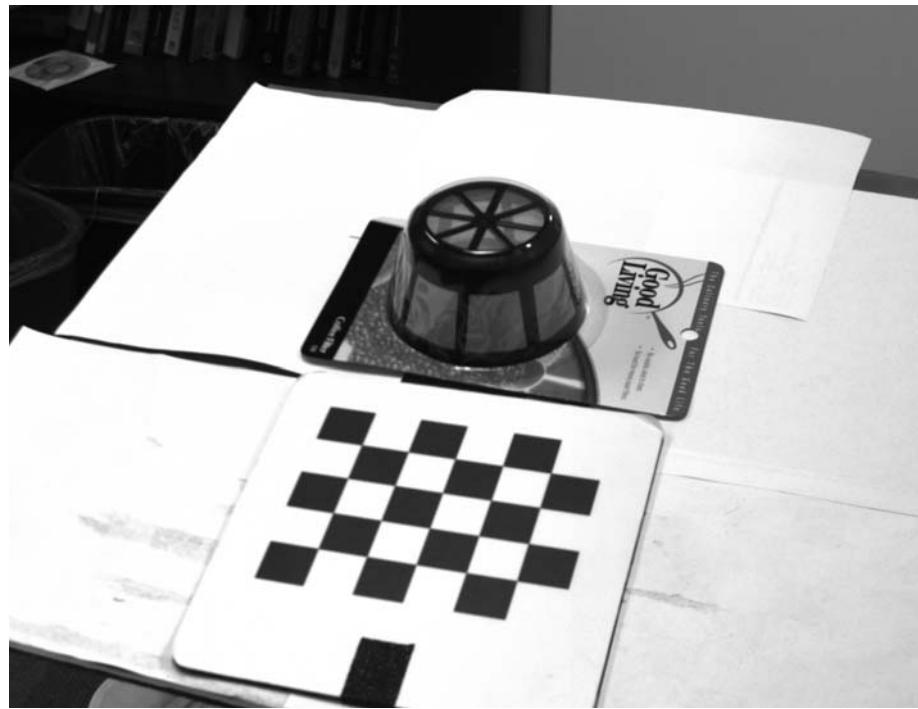
#### 3.2 The cost function

The cost function is defined by comparing detected test image edges with projections of 3D surface and silhouette edges that depend on the object pose. Given a rotation and translation of the object, we transform the point cloud into the test camera reference frame. Surface edges are projected into the image and they give us 2D surface edges because transparent objects don't have self-occlusions. In order to get silhouette edges, we project a dense point cloud into a test image, apply several closing operations to the resulting set of pixels and find the borders of the connected components. These borders constitute silhouette edges.

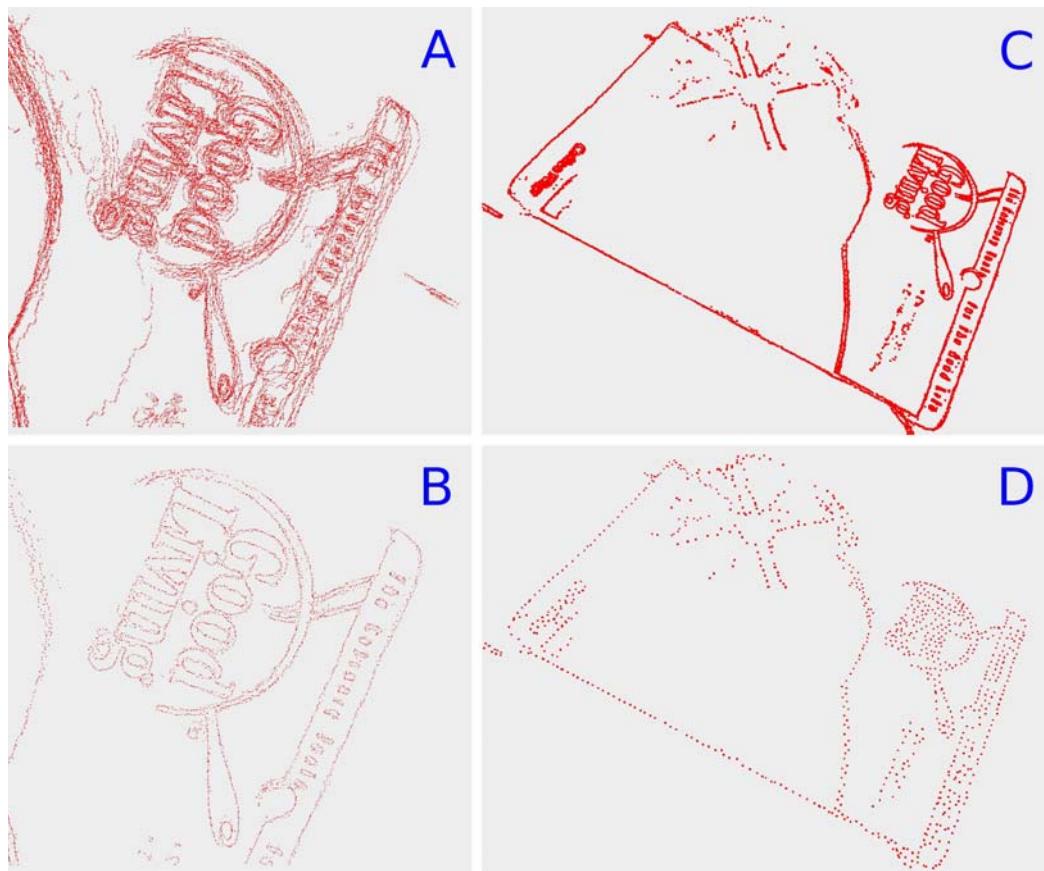
Now we want to construct a cost function that compares two sets of edges in an image. Let  $E = \{e_j\}$  be a set of pixels that belong to edges of a test image,  $T = \{t_i\}$  is a set of the model points projected into the image plane. One of the most popular cost functions is Chamfer Matching (CM):

$$d_{CM}(E, T) = \frac{1}{|T|} \sum_{t_i \in T} \min_{e_j \in E} \|t_i - e_j\|, \quad (1)$$

where  $\|\cdot\|$  is Euclidian norm. However, mean is not a robust statistic because a single outlier can affect the final value severely. Edge detection is an unstable operation that produces a lot of variation, especially in edge endpoints. In order to overcome this



**Figure 1:** Example of an object to be modeled.



**Figure 2:** Creation of the surface edge model. (A) All train point clouds transformed to the same coordinate system. (B, C) Refined and denoised point cloud with k-partite matching and robust statistics. (D) Downsampled point cloud which approximates the full model well.

issue we use the Partial Directed Hausdorff (PDH) distance [30]:

$$d_H(E, T) = K_{t_i \in T}^{th} \min_{e_j \in E} \|t_i - e_j\|. \quad (2)$$

Here  $K_{t_i \in T}^{th}(\mathbf{X})$  is the  $K$ th ranked value in the sorted set  $\mathbf{X}$ . Throughout the paper we use  $K = 0.8|T|$ , where  $|T|$  is the number of elements in  $T$ . However, this distance can be set to zero by placing the object infinitely far away from the camera. It means the global minimum will be achieved in the incorrect pose for this distance. So we introduce a Normalized PDH (NPDH) distance:

$$d_H(E, T) = \frac{1}{\sqrt{\det C}} K_{t_i \in T}^{th} \min_{e_j \in E} \|t_i - e_j\|, \quad (3)$$

where  $C$  is the covariance matrix of the projections of the point cloud into the image.

Both CM and PDH distances are known to behave incorrectly in clutter. Oriented Chamfer Matching (OCM) [10] is known to handle clutter better. However, it is more computationally expensive, so we use NPDH throughout the paper.

The PDH cost function is computed separately for surface and silhouette edgels. The resulting distances are added with different weights: 2/3 for surface and 1/3 for silhouette edges. The weight for surface edges is higher because surface edges are more stable: they are constructed by fusing edgels from many training frames, so we know that each surface edge is found robustly by the edge detector.

The cost function 3 is minimized by the global optimization algorithm DIRECT [31] from the NLOpt library [32] by varying the 6 parameters defining pose of the object: a translation and rotation vectors.

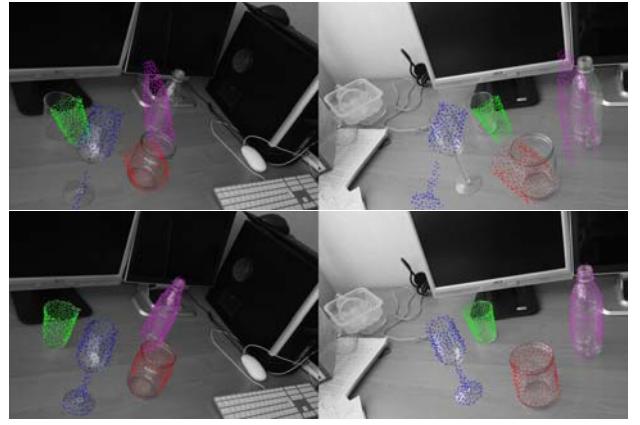
## 4. EXPERIMENTAL RESULTS

The algorithm was tested on the base of 5 transparent objects. We take 5 pairs of kitchen items and paint one object in each pair in white color to make it opaque because there are no reliable way to scan a transparent object [3]. We use the painted object to scan it with Kinect to create object models. Each training sequence contains 12 frames with different poses of the table relative to Kinect. The asymmetric circles pattern from the OpenCV library was used as the fiducial marker to estimate poses between frames. To create test data we used the corresponding transparent objects captured by a calibrated stereo pair of Canon EOS 40D cameras from a distance about 1 meter. Images were resized to resolution of about one megapixel (1166x778). Each test object is placed exactly as the corresponding training object relatively to the fiducial marker, so we know the ground truth.

The objective of these experiments is to investigate how accurate the initial guess about the object pose should be for the algorithm to produce a stable correct result. We ran the algorithm with many different initial guesses generated randomly. In particular, the correct pose was translated in random direction on the specified distance  $d$  and rotated in random direction on the specified angle  $\alpha$ . Each experiment with specific values of  $d$  and  $\alpha$  was repeated 50 times and we take 27 different combinations of these values. All objects in the test base have rotation symmetry and this was taken into account when evaluating pose returned by the algorithm but this knowledge was not used by the algorithm itself.

The example of the results is given in Fig. 3. Points of objects' models are colored. They are projected into the image plane using initial hypothesis of objects' poses and poses refined by the algorithm. Initial poses are quite far away from correct poses. However, final poses are accurate enough for grasping.

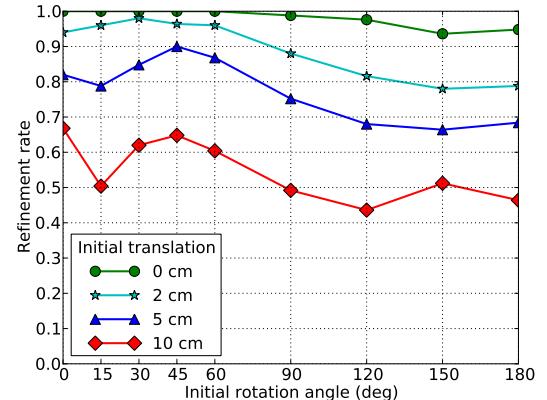
We run the algorithm on all 5 objects to see how often the algorithm returns a correct pose. We consider the pose estimation successful if the difference between the returned and correct poses is less than 2 cm in translation and 10 degrees in rotation.



**Figure 3:** Images from a stereo pair with the projected poses found with the algorithm, initial (upper row) and refined (bottom row).

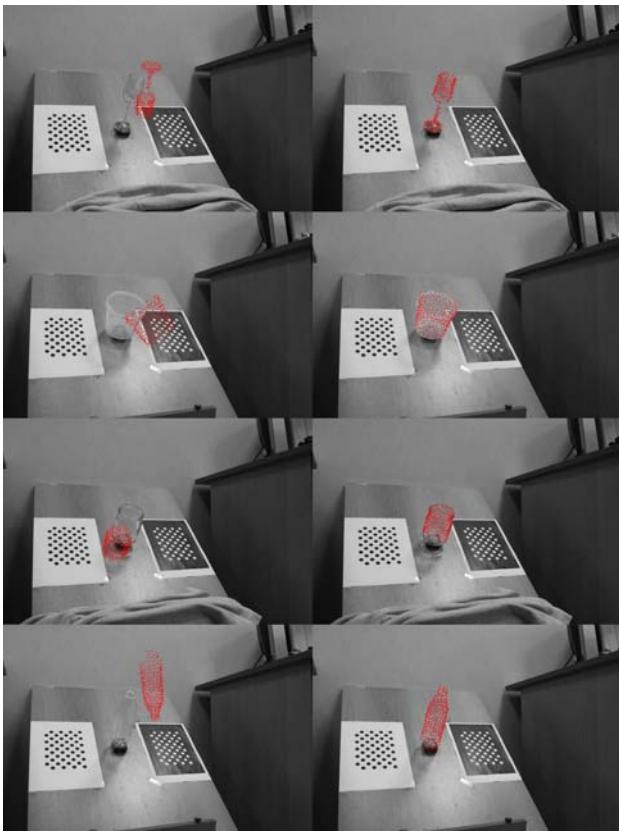
Fig. 4 shows the statistics for all 5 objects. The percent of runs when the algorithm succeeded is plotted on the y-axis. The chart shows that if the initial translation error is less than 2 cm, we can successfully reconstruct the pose in more than 80% of the cases. Black area in Kinect depth map that corresponds to specular and transparent surfaces can give us a hypothesis about the object location. This information can be used to generate a good initial guess about the translation vector. If the initial error of the translation vector is 2cm, the rate of successful reconstructions (averaged over all angles) is 88%, if the initial translation error is 5cm, then the rate of successful reconstructions is 77%. Note that part of the error comes from poses that are upside-down to the ground truth: since many objects are close to cylindrical shape, the final result can put the top of the glass to the bottom.

See more examples at Fig. 5. Also see Fig. 6 for example of the algorithm failure. The algorithm returned the pose which is upside down of correct one because the object has nearly cylindrical shape.

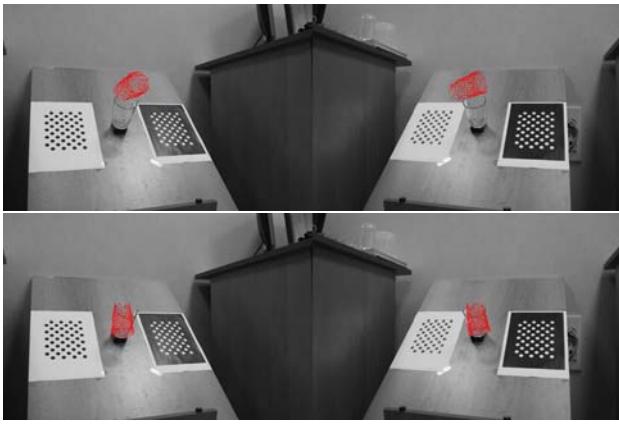


**Figure 4:** Statistics of the algorithm working on rigid transparent objects. Pose can be refined successfully if an initial pose is not very far from the correct pose. The algorithm is robust to incorrect initial rotation but it is more sensitive to initial translation.

The proposed algorithm can refine poses of transparent objects in some cases, but it has several limitations. The approach demands good initial hypothesis of the object pose, otherwise the search for the global minimum takes too much time. The algorithm is unstable in clutter e.g. if the object is surrounded by other objects. But in the case of low clutter the algorithm works with



**Figure 5:** Examples of successful pose refinement for different objects. Left images are initial poses and right images are refined poses. Only one image from the stereo pair is shown.

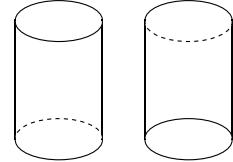


**Figure 6:** Example of the algorithm failure due to cylindrical shape of the object. Two images from a stereo pair are shown: initial pose (upper row) and refined pose (lower row).

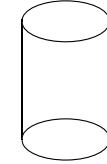
sufficient speed and quality to be applied for pose refinement of rigid transparent objects.

Another limitation of the proposed method is using a calibrated stereo pair for generating test images instead of a single monocular camera. The main obstacle for a monocular camera is ambiguity that cannot be resolved from a single image without additional assumptions or priors.

There exist significantly different poses that have very good projections to a test image and it is specificity of transparent objects. For example, two different poses of an opaque object are shown in the Fig. 7 and there is no ambiguity between them. However, if the same object is transparent then there are two different plausible interpretations of the same projection (Fig. 8) because transparent objects don't have self-occlusions and all edges are visible.

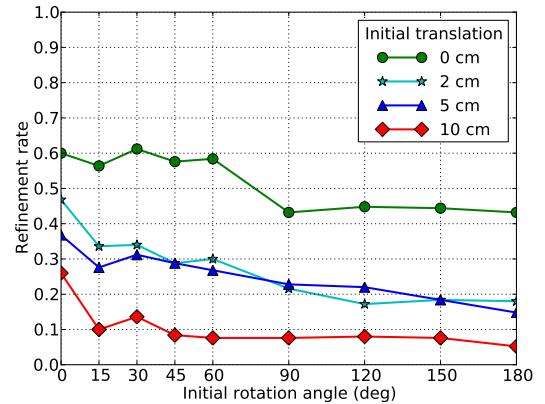


**Figure 7:** Two different poses of an opaque object. There is no ambiguity between them because different edges are visible in different poses.



**Figure 8:** Ambiguous projection of a transparent object. Two plausible poses of the object are possible because all edges are visible on the same image.

We evaluated the algorithm with a monocular camera on the same dataset using only left images of our stereo test set. The statistics of pose estimation is shown in the Fig. 9. One can see that there is a significant degradation of accuracy compared to the stereo case.



**Figure 9:** Statistics of the algorithm working when using a monocular camera only. The results degrade significantly comparing to the stereo camera due to inherent ambiguity of pose estimation of transparent objects from a single view.

## 5. CONCLUSION

The paper presents the algorithm for refining the 6-DOF pose of transparent objects. Our method only requires a calibrated stereo pair during the online stage. Given an initial estimate that has an error in translation less than 5cm, the rate of accurate pose estimations is higher than 75%. The method allows to grasp transparent objects without using expensive sensors such as TOF cameras.

## 6. REFERENCES

- [1] A. Collet, D. Berenson, S.S. Srinivasa, and D. Ferguson, "Object recognition and full pose registration from a single image for robotic manipulation," in *IEEE International Conference on Robotics and Automation*, 2009.
- [2] B. Rosenhahn, T. Brox, and J. Weickert, "Three-dimensional shape knowledge for joint image segmentation and pose tracking," *International Journal of Computer Vision*, vol. 73, no. 3, pp. 243–262, 2007.
- [3] Ivo Ihrke, Kiriakos N. Kutulakos, Hendrik P. A. Lensch, Marcus Magnor, and Wolfgang Heidrich, "State of the Art in Transparent and Specular Object Reconstruction," in *STAR Proceedings of Eurographics*, 2008, pp. 87–108.
- [4] U. Klank, D. Carton, and M. Beetz, "Transparent Object Detection and Reconstruction on a Mobile Platform," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011.
- [5] C.J. Phillips, K.G. Derpanis, and K. Daniilidis, "A Novel Stereoscopic Cue for Figure-Ground Segregation of Semi-Transparent Objects," in *1st IEEE Workshop on Challenges and Opportunities in Robot Perception*, 2011.
- [6] I. Lysenkov, V. Eruhimov, and G. Bradski, "Recognition and pose estimation of rigid transparent objects with a kinect sensor," in *Robotics: Science and Systems Conference*, 2012.
- [7] P. Lagger, M. Salzmann, V. Lepetit, and P. Fua, "3d pose refinement from reflections," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008.
- [8] J.Y. Chang, R. Raskar, and A. Agrawal, "3d pose estimation and segmentation using specular cues," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009.
- [9] A. Netz and M. Osadchy, "Using specular highlights as pose invariant features for 2d-3d pose estimation," in *Computer Vision and Pattern Recognition, 2011. CVPR 2011. IEEE Conference on*. IEEE, 2011.
- [10] J. Shotton, A. Blake, and R. Cipolla, "Multiscale categorical object recognition using contour fragments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1270–1281, 2007.
- [11] I. Biederman and G. Ju, "Surface versus edge-based determinants of visual recognition," *Cognitive Psychology*, vol. 20, no. 1, pp. 38–64, 1988.
- [12] B. Rosenhahn, *Pose estimation revisited*, Ph.D. thesis, Universität Kiel, Sept. 2003.
- [13] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab, "Dominant orientation templates for real-time detection of texture-less objects," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2257–2264.
- [14] D.M. Gavrila, "A bayesian, exemplar-based approach to hierarchical shape matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1408–1421, 2007.
- [15] C. Reinbacher, M. Ruther, and H. Bischof, "Pose estimation of known objects by efficient silhouette matching," in *2010 International Conference on Pattern Recognition*. IEEE, 2010, pp. 1080–1083.
- [16] D.G. Lowe, "Three-dimensional object recognition from single two-dimensional images," *Artificial intelligence*, vol. 31, no. 3, pp. 355–395, 1987.
- [17] M.Y. Liu, O. Tuzel, A. Veeraraghavan, R. Chellappa, A. Agrawal, and H. Okuda, "Pose estimation in heavy clutter using a multi-flash camera," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.
- [18] M. Ulrich, C. Wiedemann, and C. Steger, "CAD-based recognition of 3d objects in monocular images," in *International Conference on Robotics and Automation*, 2009, vol. 1191, p. 1198.
- [19] P.J. Besl and N.D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992.
- [20] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *International Journal of Computer Vision*, vol. 13, no. 2, pp. 119–152, 1994.
- [21] K. Pulli, "Multiview registration for large data sets," in *3-D Digital Imaging and Modeling, 1999. Proceedings. Second International Conference on*. IEEE, 1999, pp. 160–168.
- [22] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau, "Towards a general multi-view registration technique," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 5, pp. 540–547, 1996.
- [23] A.W. Fitzgibbon, "Robust registration of 2D and 3D point sets," *Image and Vision Computing*, 2003.
- [24] E. Hazan, S. Safra, and O. Schwartz, "On the hardness of approximating k-dimensional matching," in *Electronic Colloquium on Computational Complexity, TR03-020*, 2003.
- [25] R. Singh, J. Xu, and B. Berger, "Global alignment of multiple protein interaction networks," in *Proc. Pacific Symp. Biocomputing*. Citeseer, 2008, vol. 13, pp. 303–314.
- [26] M. Hubert, P.J. Rousseeuw, and S. Van Aelst, "High-breakdown robust multivariate methods," *Statistical Science*, vol. 23, no. 1, pp. 92–119, 2008.
- [27] P.J. Rousseeuw, "Multivariate estimation with high breakdown point," *Mathematical statistics and applications*, vol. 8, pp. 283–297, 1985.
- [28] D.H. Douglas and T.K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.
- [29] J. Matas, Z. Shao, and J. Kittler, "Estimation of curvature and tangent direction by median filtered differencing," in *Image Analysis and Processing*. Springer, 1995, pp. 83–88.
- [30] D.P. Huttenlocher, G.A. Klanderman, and WA Rucklidge, "Comparing images using the hausdorff distance," *IEEE Transactions on pattern analysis and machine intelligence*, pp. 850–863, 1993.
- [31] D.R. Jones, C.D. Perttunen, and B.E. Stuckman, "Lipschitzian optimization without the Lipschitz constant," *Journal of Optimization Theory and Applications*, vol. 79, no. 1, pp. 157–181, 1993.
- [32] Steven G. Johnson, "The nlopt nonlinear-optimization package," <http://ab-initio.mit.edu/nlopt>.

# Detection of swapped views in stereo image

Alexey Shestov, Alexander Voronov, Dmitriy Vatolin

Department of Computational Mathematics and Cybernetics

Moscow State University, Moscow, Russia

{ashestov, avoronov, dmitriy}@graphics.cs.msu.ru

## Abstract

An algorithm for automatic swapped views detection is proposed. No analogues in literature were found for this problem solution. It is based on occlusion detection and motion vectors histogram. The algorithm was tested on 780 frames from 13 movies. The recall\precision diagrams were constructed using two parameters. The complexity is estimated. The drawbacks are analyzed and further directions are proposed.

**Keywords:** Image Processing, Stereo Vision, Swapped Views, Channel Mismatch

## 1. INTRODUCTION

During a film production in some scenes left and right views can be occasionally swapped. Our goal is to reveal such scenes in the movies. Actually, this type of artifact is not easily detectable by human eye from the first look, because when you see such scene, you understand that something is wrong, but can't understand what exactly is. But correctness of views arrangement can be checked on the basis of foreground-background segmentation and inter-view optical flow analysis.

Currently we have performed early stage research and implemented initial version of the algorithm for swapped views detection.

## 2. THE MAIN IDEAS

### 2.1 The necessary definitions

**Binocular disparity** refers to the difference in image location of an object seen by the left and right eyes, resulting from the eyes' horizontal separation. We will call it **disparity** further.

To estimate disparity we used the **Optical Flow** (or **OF**) algorithm described in the paper[4]. We treated disparity as motion vectors between views. We will use words "**motion vectors**" and "**disparity**" as synonyms.

By **occlusions** we mean regions which are presented in only one view of a stereogram.

**Left-right consistency** (or **LRC**) is the confidence measure for optical flow [2]. We take the vector A in the point X in the left image, then we take the vector B in the point A+X in the right image. In ideal situation B+A should be equal to zero. So the greater is B+A – the less confident is the vector in the point X.

### 2.2 Ideas

Two main ideas, which are used in our algorithm:

- The first idea is based on the fact that in the left view occlusions are always leftwards the object and in the right view occlusions are always rightwards the object (see Figure 1).
- The second idea uses the fact that negative parallax regions are one-third of the zone of stereo comfort perception, and

positive parallax regions are two-third of the zone of stereo comfort perception, so by an analysis of disparity histogram we can say which disparity values correspond to negative parallax and which values correspond to positive parallax (see Figure 3).

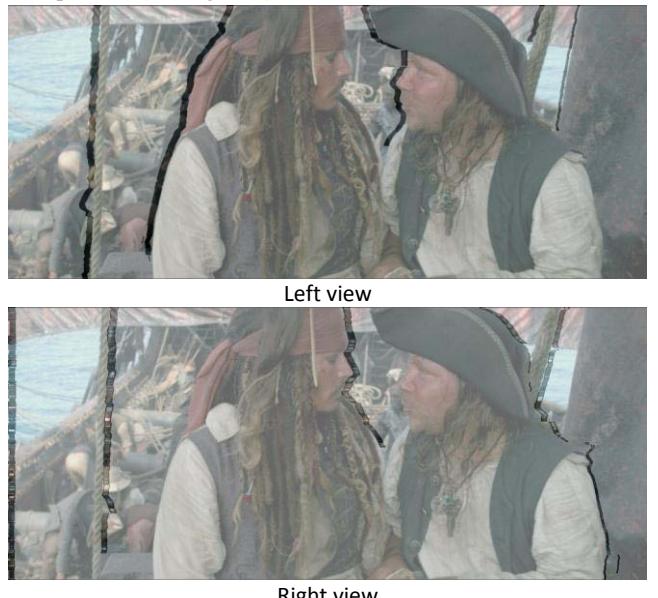


Figure 1. Here you can see, that on the left view occlusions are always leftwards the object, and on the right view – rightwards. The frame is taken from the movie "Pirates of the Caribbean: On Stranger Tides".

An illustration of the first idea is presented in Figure 2:

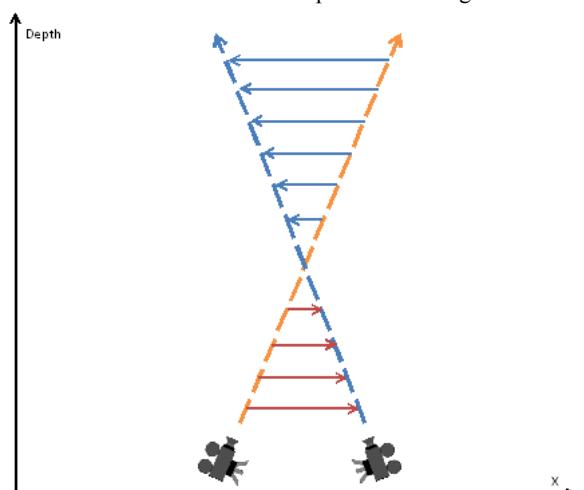


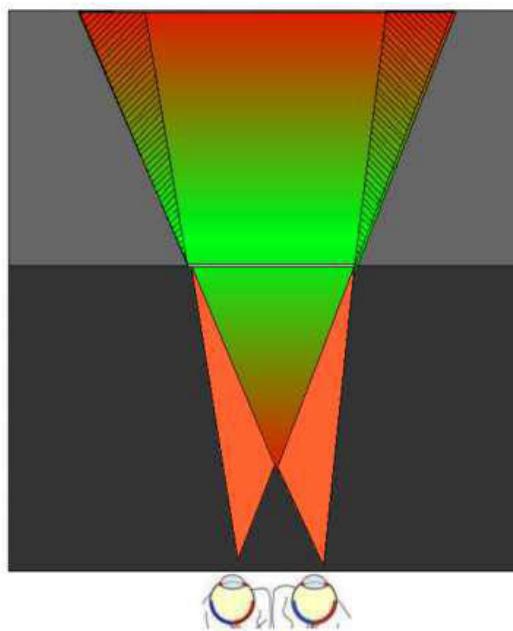
Figure 2. Here rays from the left and right cameras and the motion vector field from the right view to the left view are shown. You can see that disparity is linearly dependent on the depth. On the right view disparity value of an object always is larger than disparity value of background. So,

**objects in the right view will always have occlusions rightwards (and in the left view - leftwards).**

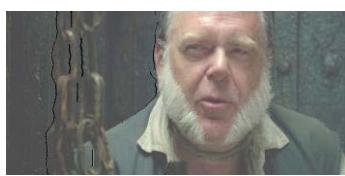
Disparity is linearly dependent on the depth. In the right view disparity value of an object always is larger than disparity value of background. So, objects in the right view will always have occlusions rightwards (and on the left view – leftwards).

So, we can use some edge detector, then count where are more image edges, rightwards or leftwards the occlusions, and calculate a probability that a current view is left or right.

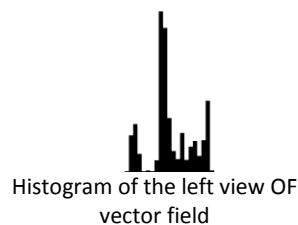
But there are some frames, where occlusion are so thin, that they are not detected by our algorithm, or if they are detected, we can't say if edges are rightwards or leftwards them. In such cases we use the next considerations: if occlusions are thin, then there mustn't be foreground objects, which depth is much different from background depth. We know that negative parallax regions are one-third of the zone of stereo comfort perception, and positive parallax regions are two-third of the zone of stereo comfort perception (see Figure 3).



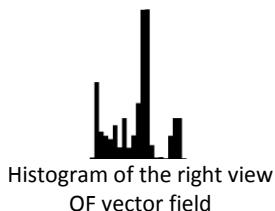
**Figure 3. Stereo perception zones. Zone of stereo comfort perception is marked with green color. You can see that**



A left view of the source frame with marked occlusions



OF vector field of the left view



**negative parallax regions are one-third of the zone of stereo comfort perception, and positive parallax regions are two-third of the zone of stereo comfort perception.**

So if depth of foreground objects isn't much different from background depth we can expect that left view will have more positive disparity values than negative and right view will have more negative disparity values than positive. So in such cases a mass center of the left view disparity histogram must have positive coordinate and a mass center of the right view disparity histogram must have negative coordinate (see Figure 4).

**Figure 4. An example of OF vector field histograms. You can see that histograms are almost symmetric. Here we have thin occlusions and the mass center of left OF vector field must be rightwards the mass center of right OF right vector field. taken from the movie “Pirates of the Caribbean: On Stranger Tides”.**

### 3. THE ALGORITHM

#### 3.1 Steps of the algorithm

1. Preprocessing step: for each view estimate a necessary data :
  - Estimate occlusions using left-right consistency thresholding.
  - Estimate image edges using Canny edge detector[1].
  - Estimate image  $L^*a^*b^*$  gradient using Sobel filter[3].
  - Estimate a disparity histogram.
2. For each view for each occlusions side (left or right) calculate a sum of products of:
  - occlusion width,
  - border confidence,
  - inverted distance between the occlusion and the boundary of the each occlusion row.

We will call these sums  $LS$  (left sum) and  $RS$  (right sum) respectively. So, we obtain four numbers:  $LS$  and  $RS$  of the left view ( $left\ LS$ ,  $left\ RS$ ) and  $LS$  and  $RS$  of the right view ( $right\ LS$ ,  $right\ RS$ ).

3. Calculate a confidence sum:  $confidence\ sum = left\ LS + left\ RS + right\ LS + right\ RS$ , compare it with the *confidence threshold*. If it is less than threshold compare moments of disparity histograms. If it is less than the *distance threshold* then views are swapped, otherwise they are not swapped.

If it is higher than threshold calculate a probability that views of the current frames are not swapped as  $(left\ LS + right\ RS) / (left\ LS + left\ RS + right\ LS + right\ RS)$ . If it is less than *probability threshold* then views are swapped, otherwise they are not swapped.

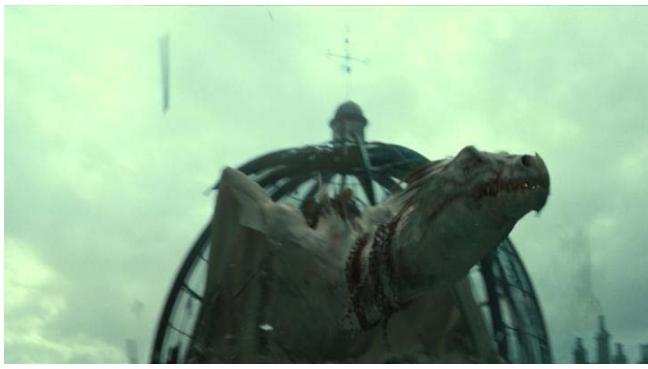
#### 3.2 The detailed algorithm explanation

##### 1. Occlusions estimation

. We use LRC thresholding and then perform median filtering of the obtained binary mask.

##### 2. Edge detection

We tried simple and reliable Canny algorithm and it produced satisfactory results (see Figure 5). We have chosen parameters for Canny using 30 test sequences from 5 films. In future we will probably use more complicated versions of edge detector.



Source frame



Source frame with edges

**Figure 5. A source image and an image with marked edges produced by Canny detector. The frame is taken from the movie Harry Potter and the Deathly Hallows.**

### 3. Gradient calculation

We use gradient values as a confidence measure for edges. An example of gradient calculated with Sobel filter is presented in Figure 6.



Source frame



Sobel gradient

**Figure 6. An example of Sobel gradient. The source frame is taken from the movie “Pirates of the Caribbean: On Stranger Tides”.**

### 4. Disparity histogram

In Figure 4 examples of left and right disparity histograms were presented. You can see that here we have thin occlusions and

the mass center of left OF vector field must be rightwards the mass center of right OF right vector field.

### 5. Occlusion-based confidence values calculation

In this step we calculate how well occlusions boundaries are aligned with image edges. We use the next considerations:

- wide occlusions are more confident than thin occlusions
- edges with higher values of gradient values are more confident
- the closer an occlusion and an edge are – the more likely they correspond to one object’s border.

### 6. Decision making

Calculate a confidence sum:  $confidence\ sum = left\ LS + left\ RS + right\ LS + right\ RS$  (these values were described above), compare it with the *confidence threshold*. If it is less than threshold compare moments of disparity histograms. If it is less than the *distance threshold* then views are swapped, otherwise they are not swapped.

If it is higher than threshold calculate a probability that views of the current frames are not swapped as  $(left\ LS + right\ RS) / (left\ LS + left\ RS + right\ LS + right\ RS)$ . If it is less than *probability threshold* then views are swapped, otherwise they are not swapped.

So we have two parameters which can be used for algorithm tuning and recall/precision regulating.

## 4. RESULTS AND ANALYSIS

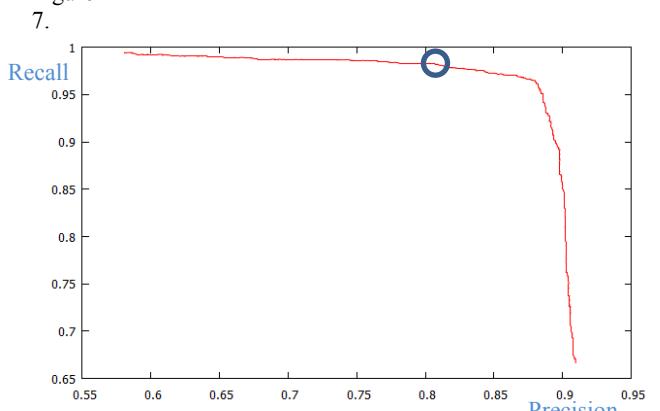
### 4.1 Results

For our test set we took 780 random frames from 13 movies, 60 frames from each movie. These frames were manually checked. For our task it is easy to obtain samples with swapped channels, we need only to swap views of good frames. So we obtain a test set of 1560 frames – 780 frames with non-swapped views and 780 frames with swapped views.

We searched for optimal parameter values in order to minimize false negatives, because our main goal is to find frames with swapped channels in real films. So in our tests we preferred to preserve high recall values and in some cases loose in precision.

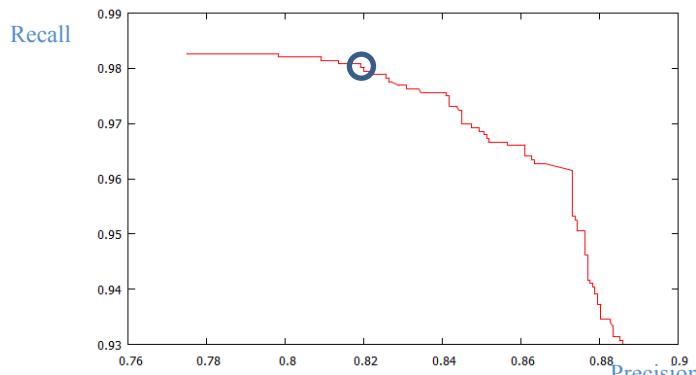
Results of our tests are presented as recall/precision diagrams.

We run our algorithm on the test set varying *probability threshold* from 0.01 to 1.0 with all other parameters fixed and obtained the recall/precision diagram, which is presented in Figure 7.



**Figure 7. The recall/precision diagram obtained varying *probability threshold* from 0.01 to 1.0 with all other parameters fixed. The test set included 780 frames from 13 movies.**

We run our algorithm on the test set varying *distance threshold* from -2.0 to 12.0 with all parameters fixed and obtained the next recall/precision diagram Figure 8.



**Figure 8.** The recall/precision diagram obtained varying *distance threshold* from -2 to 12 with all other parameters fixed. The test set included 780 frames from 13 movies.

Using our algorithm we have found swapped views in the “The Smurfs” trailer. It’s frame number 1969 (see Figure 9).



“Left” view in the trailer (which indeed must be the right view)



“Right” view in the trailer (which indeed must be the left view)

**Figure 9.** Here is the frame with swapped views. It’s frame number 1969 from the trailer “The Smurfs”.

## 4.2 Performance speed

Algorithm was tested on Intel Corei7-2630QM CPU @ 2.00GHz, 8 cores, 8 GB RAM. The test set contained 780 frames from 13 films.

An average time of work for one frame:

SD (720x480) resolution: 13.79 sec. per frame.

HD (1280x720) resolution: 51.55 sec. per frame

Now regions overlapping occlusions algorithm is based on Optical Flow algorithm implemented on CPU, so most of the time takes Optical Flow computation.

The most time consuming part during the tests was Optical Flow computation, but our optical flow also has GPU

implementation and with adapting algorithm to it we will increase speed of processing. We expect speed increase near 5–10 times.

If we apply this algorithm to the whole film, we take only one frame from a scene, because if views are swapped, they are swapped for the whole scene.

It would take about 8 hours to process all scenes from the whole film in SD resolution.

## 4.3 Further improvements

1. We can combine methods for decision making in a more complicated way, than simple thresholding.
2. We can use the consideration that a color of pixels in occlusion must be similar with background color, not with the object color.
3. There is the fact that on a left view the object with the lowest depth value must have the highest disparity value (if we consider disparity as a signed value, not only its module) and all the way round on a right view – the object with the lowest depth value must have the lowest disparity value (see Figure 2). If we detect where an object is and where a background is, i.e. find out what is further and what is closer, we can make a decision based on this knowledge. We can use rough occlusions produced by Motion Estimation for the consequent frames (not between views) and their color similarity with background for local background/foreground segmentation. Also we can use segmentation from motion for objects detection.

## 5. CONCLUSION

An algorithm for automatic swapped views detection is proposed. It is based on occlusion detection and motion vectors histogram. The algorithm was tested on 780 frames from 13 movies. The recall/precision diagrams were constructed using two parameters. The complexity is estimated. The drawbacks are analyzed and further directions are proposed.

## 6. ACKNOWLEDGMENTS

This work is partially supported by the Intel/Cisco Video-Aware Wireless Network (VAWN) Program and by grant 10-01-00697a from the Russian Foundation of Basic Research.

## 7. REFERENCES

- [1] Canny, J., *A Computational Approach To Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986.
- [2] Egnal, G., Wildes, R., P., *Detecting Binocular Half-Occlusions: Empirical Comparisons of Five Approaches*, IEEE Trans. Pattern Analysis and Machine Intelligence, 24(8):1127–1133, 2002.
- [3] H. Farid and E. P. Simoncelli, *Differentiation of discrete multi-dimensional signals*, IEEE Trans Image Processing, 13(4):496–508, 2004.
- [4] Ogale, S., A., Aloimonos, Y., *Shape and the Stereo Correspondence Problems*, International Journal of Computer Vision, 65(3):147–162, 2005.

# Problem of auto-calibration in image mosaicing

Alexey Spizhevoy, Victor Eruchimov

Lobachevsky State University of Nizhni Novogord, Russia

Itseez Ltd., Russia

{alexey.spizhevoy, victor.eruhimov}@itseez.com

## Abstract

The paper investigates the auto-calibration problem for mobile device cameras. We extend existing algorithms to get a robust method that computes internal camera parameters given a series of distant objects images. The algorithm is tested on real images generated by several different cameras. We estimate the impact of errors in camera calibration parameters on the image mosaicing problem.

**Keywords:** *auto-calibration, stitching, camera parameters, errors effect, real datasets, image mosaicing.*

## 1. INTRODUCTION

The goal of calibration is to determine internal camera parameters within the given projection model. The problem arises in a number of emerging computer vision applications such as augmented reality, 3D reconstruction, and image mosaicing (or stitching). As academy and industry becomes gradually more interested in using mobile devices for computer vision, the importance of phone/tablet cameras calibration is clear.

Nowadays the problem of camera calibration is usually solved by using special calibration patterns (see [3], [4], [5]). While pattern-based methods are quite accurate, it can be difficult to use them due to necessity of taking shots of a special calibration object like a chessboard. Also, manual calibration harms user experience that is considered crucial for mobile applications. As a result software developers and researchers are very interested in auto-calibration methods.

Auto-calibration is the process of estimating internal camera parameters directly from multiple uncalibrated images. This area of computer vision is in active research stage. From one hand there are papers describing successful attempts of using auto-calibration methods in practical tasks (e.g. augmented reality, 3D reconstruction, image mosaics, see [7], [8], [9], [11], [12]). As the topics of these papers aren't camera auto-calibration itself, they don't contain thorough investigations of the used methods with numerical evaluation, tested on challenging dataset. As a consequence, when one faces a computer vision problem that requires camera parameters, it's very difficult to select a robust auto-calibration method and reuse previous results. There is research that is directly devoted to the auto-calibration problem (see [10], [13]). Unfortunately, these papers either don't compare with state-of-the-art pattern-based calibration methods or provide evaluation for synthetic datasets only. Some of these papers describe results for real datasets, but obtained under almost ideal conditions like no noise, no hand shaking, see [13]. So to the best of our knowledge we are not aware of a research paper that describes an auto-calibration method and provides sufficient experimental evidence showing robustness for practical applications.

While classical calibration methods are well studied, they suffer from some drawbacks, which follow from the fact that these methods use some extra information. For instance, there are calibration methods (see [1]) which require location of vanishing points (i.e. points where infinite lines are terminated under projective transformation) as input, but finding of these points automatically is a difficult problem.

This paper shows that under moderate assumptions an autocali-

bration algorithm for rotational cameras presented in [1] can be used for practical applications with a necessary pre-processing step. We evaluate an implementation of the method for both simulated datasets and real image sequences generated by mobile phone cameras.

## 2. PROBLEM STATEMENT

We use the following camera model which describes how a 3D scene point  $(X, Y, Z)^T$  is projected into an image pixel with coordinates  $(u, v)^T$ :

$$w \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K(R|T) \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

where  $K$  is camera matrix of internal parameters ( $f_x, f_y$  are focal lengths in pixels,  $c_x, c_y$  are principal point coordinates);  $R, T$  are camera rotation  $3 \times 3$  matrix and translation 3-dimensional vector (external parameters);  $w$  is scale factor.

The class of auto-calibration methods that we will consider requires an existence of homography mapping between all input images. The easiest way of generating a sequence of images with homography relationship using a mobile camera is to take shots of distance objects. Hence, within the scope of this paper we will make an assumption that camera translation  $T$  is negligibly small compared to the distance to the objects. We will call a device with  $T = 0$  a "rotational camera".

We formulate the auto-calibration problem in the following way: given keypoints in input images taken by a rotational camera, and the keypoint correspondences between images, find the camera matrix  $K$ .

## 3. CAMERA MATRIX ERRORS EFFECT

The estimation of  $K$  is never the final goal of a computer vision application. So, in order to understand how precise an auto-calibration method has to be, we need to consider a specific application. This section contains a theoretical and experimental analysis for the image mosaicing problem and provides experimental evaluation on the stitching module of OpenCV library [18]. Throughout this section we make an assumption that  $f_x$  equals to  $f_y$  for the sake of simplicity and without loss of generality, as images always can be scaled to achieve of unit pixel aspect ratio.

It is possible to stitch images without involving camera matrix. In that case a user wouldn't be able to select another surface for projection except for plane, that can be inappropriate for big panoramas because of big deformations. A plane projection surface generates deformations in the panorama image are visible when the vector of camera orientation differs a lot from the projection plane normal. The most convenient projection surface for the case of rotational cameras is a sphere.

Below we analyze warping errors when the projection surface is a sphere. To compute the error for each image we do the following:

1. For each pixel  $q = (x, y, 1)^T$  of the source image we find a ray, passing through the corresponding scene point from camera center, as  $r = K^{-1}q$ , where  $K$  is the camera matrix.
2. We find the intersection point  $(X, Y, Z)^T$  of the ray with the unit sphere centered at the origin. This point spherical coordinates  $u, v$  after scaling by constant  $s$  are point coordinates on the final panorama ( $s$  is usually selected being roughly close to the focal length in pixels):

$$u = s \cdot \tan^{-1}\left(\frac{X}{Z}\right) \quad (1)$$

$$v = s \cdot (\pi - \cos^{-1}\left(\frac{Y}{\sqrt{X^2 + Y^2 + Z^2}}\right)) \quad (2)$$

3. To calculate per pixel error we project points using the ground truth camera matrix

$$K^{(gt)} = \begin{pmatrix} f^{(gt)} & 0 & c_x^{(gt)} \\ 0 & f^{(gt)} & c_y^{(gt)} \\ 0 & 0 & 1 \end{pmatrix}$$

and its estimation

$$K^{(est)} = \begin{pmatrix} f^{(gt)} f^{(rel)} & 0 & c_x^{(gt)} c_x^{(rel)} \\ 0 & f^{(gt)} f^{(rel)} & c_y^{(gt)} c_y^{(rel)} \\ 0 & 0 & 1 \end{pmatrix}$$

where  $f^{(rel)}$ ,  $c_x^{(rel)}$ ,  $c_y^{(rel)}$  are estimated camera parameters relative to the ground truth. The distance between two points obtained using  $K^{(gt)}$  and  $K^{(est)}$  is the warping error in the pixel  $p$ .

According to the presented algorithm we first get two ray directions:

$$r^{(gt)} = \begin{pmatrix} X^{(gt)} \\ Y^{(gt)} \\ Z^{(gt)} \end{pmatrix} = (K^{(gt)})^{-1} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3)$$

$$r^{(est)} = \begin{pmatrix} X^{(est)} \\ Y^{(est)} \\ Z^{(est)} \end{pmatrix} = (K^{(est)})^{-1} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (4)$$

Then we use (1) and (2) to get pixels coordinates  $(u^{(gt)}, v^{(gt)})^T$  and  $(u^{(est)}, v^{(est)})^T$ . The differences between these pixel coordinates are:

$$u_{err} = s(\tan^{-1}\left(\frac{x}{f^{(gt)}}\right) - \tan^{-1}\left(\frac{x}{f^{(est)}}\right))$$

$$v_{err} = s\left(\cos^{-1}\left(\frac{y}{\sqrt{(x - c_x^{(gt)})^2 + (y - c_y^{(gt)})^2 + (f^{(gt)})^2}}\right) + \cos^{-1}\left(\frac{y}{\sqrt{(x - c_x^{(est)})^2 + (y - c_y^{(est)})^2 + (f^{(est)})^2}}\right)\right)$$

The final pixel warp error equals to  $\sqrt{u_{err}^2 + v_{err}^2}$ . We assess warping errors for the case of  $2048 \times 1536$  images and using the following camera matrix as a reference:

$$K^{(gt)} = \begin{pmatrix} W + H & 0 & \frac{W}{2} \\ 0 & W + H & \frac{H}{2} \\ 0 & 0 & 1 \end{pmatrix}$$

where  $W$  and  $H$  are image width and height respectively. The warping error function charts for 5% relative errors in camera internal parameters are shown in figures 1, 2, and 3.

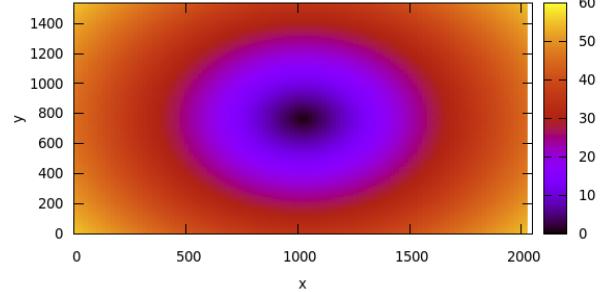


Figure 1: Pixel warp error for  $f^{(rel)} = 1.05$ .

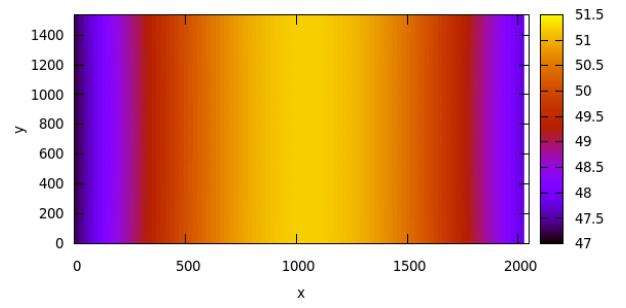


Figure 2: Pixel warp error for  $c_x^{(rel)} = 1.05$ .

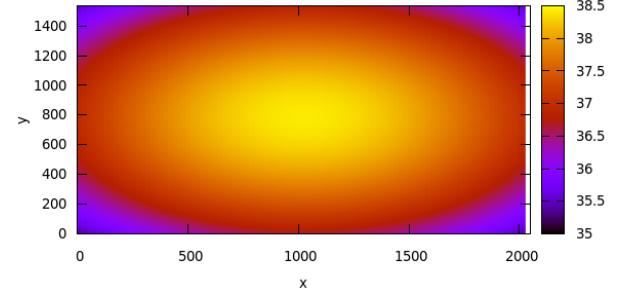


Figure 3: Pixel warp error for  $c_y^{(rel)} = 1.05$ .

We can see from charts, that when relative error in camera parameters is 5% warp error reaches 60 pixels, that seems to be high enough for leading to visible artifacts.

In order to evaluate the artifacts, we stitched  $1536 \times 2048$  images using camera matrix  $K^{(pt)}$  as ground truth  $K^{(gt)}$ , where  $K^{(pt)}$  was the camera matrix obtained via a pattern based calibration method. Also we did experiments using camera matrix  $K^{(est)}$ , where each parameter was modified (one at a time) to get 10% error (relative to  $K^{(pt)}$ ). We got panoramas without visible artifacts, see figures 4 and 5. Small artifacts are highlighted with red color, but the quality of the panoramas is much higher than we could expect from theoretical analysis. Such results are obtained because current stitching applications (including the one used for testing) use seam estimation methods to minimize visible artifacts, see [14]. After estimating seams special blending methods are used to hide discrepancies between images, see [15]. So even if the image registration step introduces moderate errors,

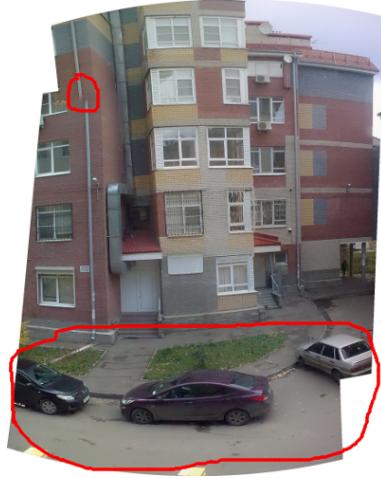
a combination of modern seam estimation and blending methods can remove a lot of possible artifacts. But if errors in camera parameters is too high then it's almost impossible to hide stretches and other artifacts, see figure 6 with results for  $f^{(rel)} = 0.7$  (i.e. 30% relative error)



**Figure 4:** Left, right source images and panorama obtained with  $K^{(pt)}$ .



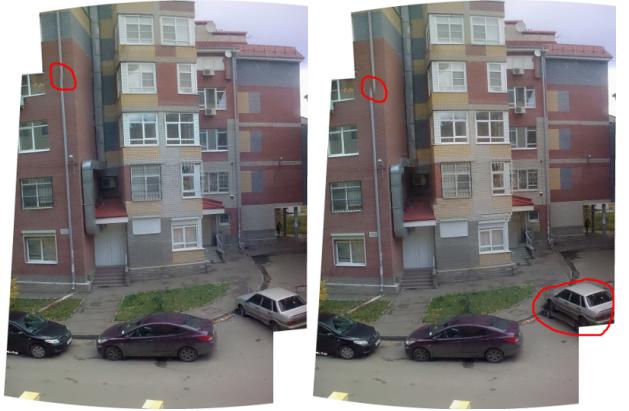
**Figure 5:** Panoramas for  $f^{(rel)} = 1.1$ ,  $c_x^{(rel)} = 1.1$ , and  $c_y^{(rel)} = 1.1$  respectively.



**Figure 6:** Panorama for  $f^{(rel)} = 0.7$  with visible artifacts and stretches.

Also it should be mentioned that motions between images are estimated to minimize overall re-projection error (that is minimizing visible mis-registration error) according to the current camera matrix. This step is very important as minimizing re-projection errors leads to minimizing visible artifacts even if the camera matrix was estimated inaccurately. figure 7 shows stitching results of two images with relative motion estimated using  $K^{(pt)}$  and with relative motions refined to minimize re-projection errors under assumption that  $f^{(est)} = f^{(pt)} f^{(rel)}$  where  $f^{(rel)} = 1.1$ .

From these results it follows that if one has a high quality stitching algorithm then the effect of errors in camera matrix isn't very high, and methods less accurate than pattern based calibration can be used for camera parameters estimation. This is a good application for auto-calibration that is not as precise as pattern-based



**Figure 7:** Panorama for  $f^{(rel)} = 1.1$  and refined motion vs. panorama for  $f^{(rel)} = 1.1$  and motion obtained using  $K^{(pt)}$ .

calibration but still generates a reasonable estimation of internal camera parameters.

#### 4. AUTO-CALIBRATION ALGORITHM

A robust auto-calibration algorithm faces many challenges coming from data generated by a mobile device. Some input images can be noisy, can differ in illumination, and undesired objects such as user fingers can be present in the camera field of view. All these issues can affect the quality of extracted features, and can lead to mis-registration. Hence, let alone the core auto-calibration problem, we have to address these issues. This is why we start with a description of our registration algorithm.

The outputs of the registration algorithm is the images graph, where vertices are images from the input image sequence, and two images are connected with the edge iff we were able to register them with a homography transformation. Here is the description of the registration pipeline:

1. Find keypoints and their descriptors of each image. We use SURF detector and descriptor implemented in OpenCV library, see [16].
2. For each image pair find matches between keypoints. We use FLANN matcher integrated into OpenCV library, see [17].
3. For each image pair estimate 2D homography and compute number of inlier matches, see 4.1.
4. For each image pair determine whether matches between these images are trustworthy, see section 4.1. The decision is made for image pair, not for each match. So if we're confident then we add an edge between two corresponding vertices into images graph.
5. Retain the biggest connected component from the images graph. Also retain only matches for confident image pairs and continue working with this connected component.

##### 4.1 Computing match confidence

We follow the method proposed in [2], where it is applied to extract a subset of images from the original raw set for subsequent stitching.

Suppose we have  $n_f$  feature matches. The correctness of an image match is represented by the binary variable  $m \in \{0, 1\}$ . The event that the  $i^{th}$  feature match  $f^{(i)} \in \{0, 1\}$  is an inlier/outlier is assumed to be independent Bernoulli event, so the total number of inliers  $n_i$  is Binomial. If  $m = 1$  then  $n_i$  has the  $B(n_i; n_f, p_1)$  distribution function, and  $B(n_i; n_f, p_0)$  otherwise, where  $p_1$  is

the probability that a feature is an inlier given a correct image match, and  $p_0$  is the probability a feature is an inlier given a false image match.

Here is the final criterion used by the authors to accept an image match

$$\frac{B(n_i; n_f, p_1)P(m=1)}{B(n_i; n_f, p_0)P(m=0)} \geq \frac{p_{min}}{1-p_{min}} \quad (5)$$

Choosing the values for  $p_1 = 0.6$ ,  $p_0 = 0.1$ ,  $P(m=1) = 10^{-6}$  and  $p_{min} = 0.999$  gives the condition

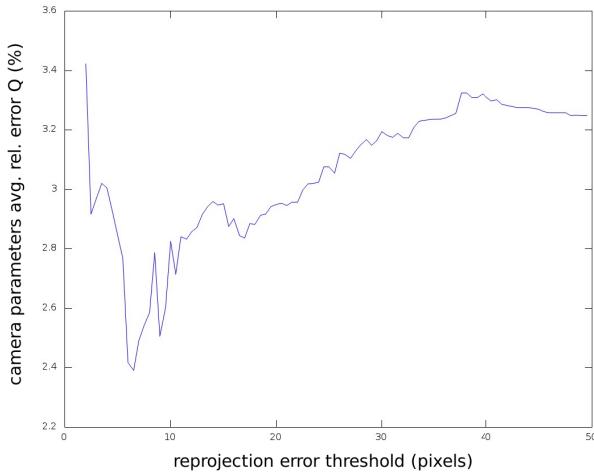
$$n_i > \alpha + \beta n_f \quad (6)$$

for a correct image match, where  $\alpha = 8.0$  and  $\beta = 0.3$ . We decide whether a feature match is an inlier or an outlier by comparing reprojection error with a fixed threshold. We used the same value of 3 pixels for all datasets and that value worked good enough in practice, while for each particular dataset another threshold value can be better.

The value  $\frac{n_i}{\alpha + \beta n_f}$  is used as the measure of confidence that it makes sense to use matches between an image pair. If it's greater than 1 then an image match is correct, false otherwise. In some practical cases it could be useful to increase this threshold as was found in experiments.

Figure 8 shows how reprojection error threshold affects on average camera parameters estimation relative error  $Q$  for one of real datasets.

$$Q = \frac{1}{4}(|f_x^{(rel)} - 1| + |f_y^{(rel)} - 1| + |c_x^{(rel)} - 1| + |c_y^{(rel)} - 1|) \quad (7)$$



**Figure 8:** Reprojection error threshold effect on camera parameters estimation errors. When the threshold is too low the algorithm is too sensitive to noise, while in the case of too high threshold even incorrect matches can be classified as inliers.

## 4.2 Rotational camera auto-calibration

For auto-calibration we use the algorithm for the rotation only cameras case proposed in [1]. Here is the brief description of that algorithm:

1. Normalize the homographies  $H_{i,j}$  between views  $i$  and  $j$  such that  $\det H_{i,j} = 1$ .
2. Compute  $\omega = (KK^T)^{-1}$  from the equations

$$\omega = H_{j,i}^T \omega H_{j,i}$$

for all image pairs  $i, j$ .

No. of images	Distance (m)	Relative errors (%)			
		$f_x^{(err)}$	$f_y^{(err)}$	$c_x^{(err)}$	$c_y^{(err)}$
6	2	8.5	11.1	4.7	4.6
7	0.5	-3.8	-2.6	-12.4	-6.2
9	2	-3.4	0.1	2.5	5.4
13	2	2.6	7.6	1.5	8.9
14	30	5.6	6.5	-1.9	4.2

**Table 1:** Relative errors for the auto-calibration of Nokia 6303C camera.

3. Compute  $K$  solving  $\omega = (KK^T)^{-1}$  with the Cholesky decomposition.
4. Refine  $K$  by minimizing the re-projection error function

$$err(K, R_1, \dots, R_n) = \sum_{i,j,k} \|x_j^{(k)} - H_{i,j}x_i^{(k)}\|$$

using parametrization of  $H_{i,j} = KR_jR_i^T K^{-1}$  over camera rotations  $R_i, R_j$  and camera matrix  $K$ , where  $n$  is the number of images and  $x_i^{(k)}, x_j^{(k)}$  are the position of  $k$ -th point measured in the  $i$ -th and  $j$ -th images respectively. We parametrize a rotation with a 3-dimensional vector directed parallel to the rotation axis and with the length equal to the rotation angle.

## 5. EXPERIMENTS

We performed experiments on real datasets taken with Nokia 6303C mobile phone ( $1536 \times 2048$  resolution) and Logitech QuickCam Pro 900 ( $1600 \times 1200$  resolution).

### 5.1 Nokia 6303C

Table 1 presents results we got using Nokia 6303C camera. We compare the auto-calibration results with pattern-based calibration:  $f_x^{(err)} = f_x^{(rel)} - 1 = \frac{f_x^{(est)}}{f_x^{(pt)}} - 1$ . The auto-calibration algorithm gives relative errors less than 10% on 3 out of 5 datasets. We have showed before that a relative error of less than 10% in camera parameters is enough for getting visually acceptable panoramas.

There are two factors affecting calibration quality. The first factor is the number of images in input dataset, because if the input dataset is too small then it doesn't provide enough information for camera auto-calibration. The second factor is non-zero translation presence, as the auto-calibration method we use was designed under the rotational camera assumption. This assumption is easily violated in practice as a user tends to rotate camera not around its optical center, but around device center (or itself), which is not the same.

### 5.2 Logitech QuickCam Pro 900

Table 2 presents result we got using Logitech QuickCam Pro 900 camera. For this camera we achieved the relative error less than 9% in comparison with OpenCV pattern based calibration results.

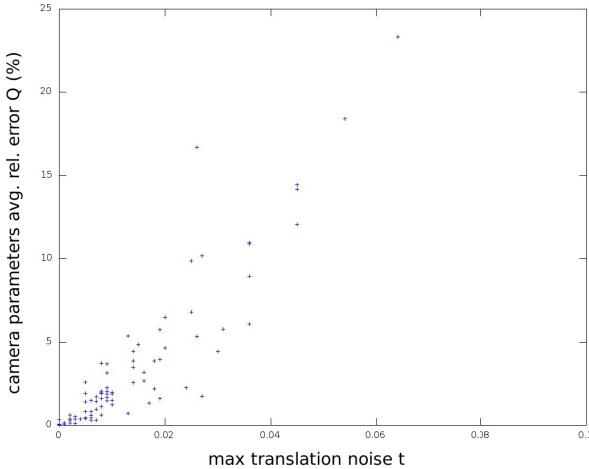
## 6. TRANSLATION NOISE IMPACT

We also performed experiments on synthetic data to analyze the dependence between camera parameters estimation errors and the translation between camera positions. We created a synthetic scene consisting of 1000 points located randomly on a unit sphere (uniformly in spherical coordinates) centered at the world frame origin and a camera located at the point  $(0, 0, -10)^T$  directed to the origin. We rotated the camera randomly to generate a sequence of images. The number of shots was uniformly distributed

No. of images	Distance (m)	Relative errors (%)			
		$f_x^{(err)}$	$f_y^{(err)}$	$c_x^{(err)}$	$c_y^{(err)}$
10	2	0.5	5.3	3.6	-0.3
30	2	1.2	4.4	0.7	2
57	2	0.3	3.1	1.5	3.2
10	2	-1.8	0.7	-2.5	1.3
30	2	1.9	6	-0.3	8.6
74	2	0.1	4.3	0.2	7.6

**Table 2:** Relative errors for the auto-calibration of Logitech QuickCam Pro 900.

in the region [3, 30]. Camera position translation noise was generated uniformly in range  $[-t, t]$  where  $t \in [0, 0.07]$  (distributed uniformly between experiments). Experimental results are shown in figure 9. It should be mentioned that in the case of high noise translation error the auto-calibration method is unstable and the method can end up with no solution. That's why we have less points on the figure when  $t$  is high.



**Figure 9:** Translation noise effect on camera parameters estimation errors.

The chart implies that the camera parameters estimation errors are highly correlated with translational noise. A 3D reconstruction from images can be done only up to a scale, so instead of using absolute values in the chart we plot the ratio of the translation noise level to the distance from the camera to the nearest object in the scene.

From what we said above follows, that in practice one must try to take shots of distant objects or select images from source dataset where objects are distant. For distant objects homography registration will be more accurate, so in principle we can filter out close objects by high re-projection error.

According to the chart, one can say that in order to get a relative auto-calibration error about 10% it is recommended to make sure that the camera translation component divided by the distance to the objects is less than  $\frac{0.025}{10}$  (according to the chart 0.025 translation noise corresponds to 10% level of auto-calibration error, where 10 is the distance from the camera to the sphere). That means about 2.5 cm shaking amplitude in case of 10 m distance must be ensured to get about 10% errors.

## 7. CONCLUSION

In our work we investigated the problem of auto-calibration for the case of rotational cameras and built a robust auto-calibration pipeline, which were tested successfully on real datasets.

Russia, Moscow, October 01–05, 2012

We performed analysis of error in camera parameters impact on final results in such computer vision problem as image mosaicing, and showed that using modern stitching algorithms relaxes requirements on camera parameters accuracy, when theoretically errors in camera parameters can lead to big warping errors.

It is possible to calibrate cameras without patterns, but the main point is that the quality of input data is important for achieving accurate auto-calibration. For the case of rotational cameras auto-calibration it is necessary to ensure that translation noise relative to distance from camera to the nearest object is small enough.

## 8. REFERENCES

- [1] Hartley, R. and Zisserman, A., “Multiple View Geometry in Computer Vision,” Second Edition, Cambridge University Press, ISBN: 0521540518, 2004.
- [2] M. Brown and D. G. Lowe, “Automatic Panoramic Image Stitching using Invariant Features,” International Journal of Computer Vision, vol. 74, no. 1, pp. 59–73, 2007.
- [3] Z. Zhang, “Flexible camera calibration by viewing a plane from unknown orientations,” Proceedings of the 7th International Conference on Computer Vision, pp. 666673, Corfu, September 1999.
- [4] Z. Zhang, “A flexible new technique for camera calibration,” IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2000): 13301334.
- [5] Gary Bradski, Adrian Kaehler, “Learning OpenCV: Computer Vision with the OpenCV Library,” O’Reilly Media, 2008.
- [6] Richard Szeliski, “Image Alignment and Stitching: A Tutorial,” Microsoft Research, TechReport, MSR-TR-2004-92, 2004.
- [7] Richard Szeliski and Heung-Yeung Shum, “Creating full view panoramic image mosaics and texture-mapped models,” Association for Computing Machinery, Inc., Computer Graphics (SIGGRAPH’97 Proceedings).
- [8] Ashley Eden, Matthew Uyttendaele, and Richard Szeliski, “Seamless Image Stitching of Scenes with Large Motions and Exposure Differences,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’2006).
- [9] M. Pollefeys, “Visual 3D Modeling from Images,” Tutorial Notes, <http://www.cs.unc.edu/~marc/tutorial/>, 2000.
- [10] D. Nister, “Untwisting a projective reconstruction,” International Journal of Computer Vision, 2004.
- [11] Simon Gibson, Jon Cook, Toby Howard, Roger Hubbold, Dan Oram, “Accurate Camera Calibration for Off-line, Video-Based Augmented Reality,” ISMAR ’02 Proceedings of the 1st International Symposium on Mixed and Augmented Reality.
- [12] Jing Chen, Baozong Yuan, “Metric 3D reconstruction from uncalibrated unordered images with hierarchical merging,” 2010 IEEE 10th International Conference on Signal Processing (ICSP)
- [13] M.K. Chandraker, S. Agarwal, D.J. Kriegman and S. Belongie, “Globally Optimal Algorithms for Stratified Auto-calibration,” IJCV 90(2):236–254, November 2010.
- [14] Vivek Kwatra, Arno Schodl, Irfan Essa, Greg Turk, Aaron Bobick, “Graphcut Textures: Image and Video Synthesis Using Graph Cuts,” To appear in Proc. ACM Transactions on Graphics, SIGGRAPH 2003.

- [15] Adelson, E. H., and Burt, P. J. "Multi-resolution Splining Using a Pyramid Image Representation," SPIE Applications of Digital Image Processing IV, pp. 204-210, 1983.
- [16] Bay, Herbert and Tuytelaars, Tinne and Van Gool, Luc "SURF: Speeded Up Robust Features," Computer Vision ECCV, Lecture Notes in Computer Science, Volume 3951/2006, 404-417, 2006.
- [17] FLANN - Fast Library for Approximate Nearest Neighbors, <http://people.cs.ubc.ca/~mariusm/index.php/FLANN/FLANN>, 2011.
- [18] OpenCV v2.4.0 stitching module documentation, <http://opencv.itseez.com/modules/stitching/doc/stitching.html>, 2012.
- [19] Ji, Q., Dai, S., "Self-Calibration of a Rotating Camera with a Translational Offset," IEEE Trans on Robotics and Automation, V. 20, N.1, Feb 2004
- [20] Junejo, I., Foroosh, H., "Practical PTZ Camera Calibration using Givens Rotations," IEEE ICIP 2008

## ABOUT THE AUTHORS

Alexey Spizhevoy is a 2nd year M.Sc. student at Nizhny Novgorod State University and software engineer at Itseez where he created stitching and video stabilization modules in OpenCV library. He also ported pedestrian detection and optical flow estimation on GPU and integrated it into OpenCV GPU module. His contact email is [alexey.spizhevoy@itseez.com](mailto:alexey.spizhevoy@itseez.com).

Victor Eruhimov is chief technical officer of Itseez company. Prior to co-founding the company, he worked as a project manager and senior research scientist at Intel, where he applied computer-vision and machine-learning methods to automate Intel fabs and revolutionize data processing in semiconductor manufacturing. Before joining the manufacturing group he was developing technologies for human-motion capture, image retrieval, super-resolution, and face analysis. He is the author of more than 20 computer-vision and machine-learning papers, and holds several U.S. and international patents. His contact email is [victor.eruhimov@itseez.com](mailto:victor.eruhimov@itseez.com).

## PANORAMAS

In this section we show panoramas that were obtained using OpenCV stitching pipeline [18] with camera parameters obtained with the described auto-calibration method.



# MONOCULAR OBJECT LOCALIZATION BY SUPERQUADRICS CURVATURE REPROJECTION AND MATCHING

Enrico Zappia, Ilya Afanasyev, Nicolo' Biasi, Mattia Tavernini, Alberto Fornaser, Antonio Selmo and Mariolino De Cecco

Department of Mechanical and Structural Engineering (DIMS), University of Trento, via Mesiano, 77, Trento, Italy

enrico.zappia@gmail.com, {ilya.afanasyev, mariolino.dececco}@unitn.it

## Abstract

This paper presents a new method for 3D object localization from a single image. It is known that single camera provide 2D image data, annihilating valuable 3D information about object and its localization in space. The main new idea is to match 2D image gradient to the reprojection of 3D curvature to retrieve objects position relative to the camera. The object parameters are a-priori known and modelled by SuperQuadratics (SQ) that enable the calculation of the analytical form of curvature. The image processing stage includes object detection and segmentation by the Histogram of Oriented Gradients (HOG) algorithm. The method proposed uses the dependencies between SQ curvature and image gradient also considering the illumination model and object contour embedded in a proper cost function. To manage local minima we propose the use of particle swarm optimization (PSO).

**Keywords:** *SuperQuadratics, single camera, 3D object localization, Histogram of Oriented Gradients (HOG), curvature matching, and particle swarm optimization (PSO).*

## 1. INTRODUCTION

Object localization is an important task of computer vision and robotics with many applications in the fields of autonomous-guided vehicles, robot picking and manipulation, augmented reality, non-contact measurement, etc. In recent years, thanks to the increasing interest in these fields, some different approaches have been proposed. This work presents a new and effective method of object localization by matching image properties acquired by a single CCD to object curvature in 3D whose model is obtained in analytical form by SuperQuadratics. There are some papers related to the pose estimation with SQ [1, 3, 14], but all of them match 3D range data (point cloud) to SQs models. On the one hand this approach becomes very efficient and robust to outliers [22], but on the other hand it requires 3D depth cameras, lasers rangefinders or multicamera setup. Other related papers focus on the pose estimation from a single image. Such papers can be divided into two main groups according to their main model descriptors[8]: the first uses the model edges (wireframes), the second spatial localized model features (regions). The main idea of the edge-based methods like [5, 9, 15] is to reproject a model contour on the gradient image. These methods are very time efficient and can be used also for object tracking, but the object's profile is often strongly varied along the edges due to e.g. clutter, shading, and texture. For these reasons, the edge detection is usually performed on the maximal image gradient. The region-based methods such as [24, 21] rely on the homogeneity of spatially localized features (e.g. RGB values, curvatures, etc.). The assumption is that the features of all pixels of a region are distributed with statistical independence according to the same probability density function. Often this assumption leads to incorrect results if e.g. the distributions of RGB values of foreground and background depend on the object location within the image. Fusing edge-based and region-based approaches gives a more effective and robust way for objects segmentation and matching [10]. Other works have been proposed to retrieve 3D information from set of images like [4]. In this case object identification and 3D parameters are obtained from SIFT features. The model is stored

with a sample image that is correlated to the real object. This method gives good results with variegated texture.

**SuperQuadratics** are an extension of basic quadric surfaces, which were introduced in computer vision by Alan Barr [2]. These mathematical functions allow the representation of a pretty high number of elementary solids, e.g. sphere, box, cylinder, toroids. Advantages of this formulation are compactness and its closed-form mathematical expression. Furthermore, SQ can be roughly described as deformation of a sphere so they are continuous surfaces, even through edges.

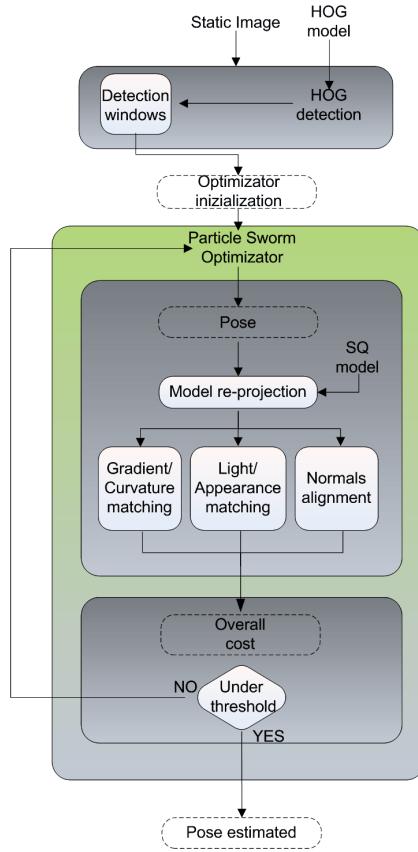
**Histogram oriented gradient (HOG)** is able to retrieve objects in the picture. It is a stochastic algorithm that uses the distribution of intensity gradients for object detection. This method has been first introduced to solve the problem of pedestrian identification in static images [17, 18]. The algorithm focuses on finding the robust features descriptor of human model, maintaining invariance to a wide variety of articulated pose minimizing the influence of background and illumination. HOG processes a sample image comparing it with a inner object model. It is obtained through a training session with several different samples of the object acquired from different points of view. All data are collected for training support vector machine that compares the sample image with its model. The output is the positive detection status and a rough localization of the object in the image plane. We propose a novel method of object detection and localization that, exploiting SQ analytical formulation, implements curvature re-projection keeping into account contours, edges and region properties at once. Detection is obtained using state of the art HOG algorithm. With this approach we rely on a wider set of informations than just edges or key points as traditional model based approaches.

## 2. ALGORITHM OUTLINE

The algorithm (1) starts with the HOG object detection. The result is a detection window on the acquired image containing the object. This defines a lower and an upper bounds for the optimizer research domain. Then a first pose guess is generated by a transformation matrix applied to the a-priori known SQ-model. From analytical SQ formulation the curvature, the normals and the lighting model are computed and reprojected onto the image to evaluate the level of matching with image features (gradients) by means of a properly defined cost function. Until the overall cost exceeds a certain threshold, the optimization process continues, when the threshold is reached, object localization is returned.

## 3. OBJECT DETECTION USING HISTOGRAMS OF ORIENTED GRADIENTS

This work focuses on the estimation of the object pose and location with respect to the camera. The first stage of image processing is the HOG object detection and identification of the Region Of Interest (the Detection Window). Then the resulting window is used to set the initial guess on object location for the optimizer initialization). HOG involves two main phases: features extraction and learning.



**Figure 1:** The flowchart of the object localization algorithm from static images.

### 3.1 Features extraction

The object images are normalized (e.g. by gamma normalization) and then their gradients magnitude and orientation evaluated. The detection window is divided into sub cells. The histogram of oriented gradients, weighted according to the magnitude of the gradients intensity, is then computed. Data are collected into Blocks with an additional normalization to provide better illumination invariance. To complete the dataset of positive images, also negative ones are processed (i.e. images without the target). After that HOG data are collected for all the detection windows and then feature vectors (both negative and positive images) are combined together to be processed by the support vector machine (SVM) for the Learning phase.

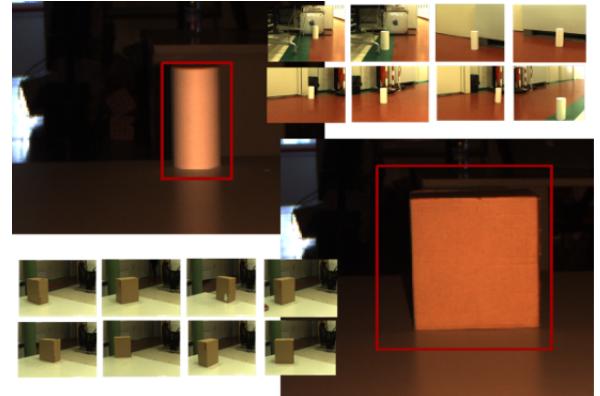
### 3.2 Learning phase

Images from the dataset created in the preview phase are encoded as spatial feature vectors. This set is processed into a binary classifier for object / non-object class identification. At this stage the detection/recognition is not robust as a high number of false positives can be obtained from the first experimental dataset, i.e. the non-object class is not properly acquired/described. To reduce the false positive detection the second sample dataset with only negative images is prepared and processed. In this way all positive results in this sequence are false positive, so they can be re-introduced in the classifier as the hard negative example to perform better non-object class description. A new classification is finally obtained using the two classes. False positives are now reduced by an order of magnitude.

### 3.3 HOG implementation

The HOG implementation is the same as described in [19]. Some of the improvements comprised the UoC-TTI LSVM-MDPM

entry in the PASCAL VOC 2009 comp3 challenge [6]. The dataset chosen concerned simple objects like a box (219 x 120 x 230 mm) and a cylinder (190 x 90 mm). Training is performed with different images of the same object (80 positives image and 30 negatives for each model). As shown in figure 2 the final HOG result is a bounding box around the detected object. This information provides the segmentation of the region of interest from the background to initialize the optimizer.



**Figure 2:** The set of some training images (for a cylinder and a box) and the final results of HOG detection (the red rectangle surrounding the objects for test images).

This kind of segmentation provides information only into 2D domain, whereas pose and/or depth information are not estimated. Anyway this stage allows concentrating the optimization algorithm attention only on a limited zone of the image thus speeding up the subsequent steps.

## 4. SUPERQUADRICS FORMULATIONS

SQ surfaces can be obtained as spherical product of two parametric curves.

Given two parametric curves:

$$h(\omega) = \begin{bmatrix} h_1(\omega) \\ h_2(\omega) \end{bmatrix} \quad \pi \leq \omega \leq \pi \quad (1)$$

$$m(\eta) = \begin{bmatrix} m_1(\eta) \\ m_2(\eta) \end{bmatrix} \quad \frac{\pi}{2} \leq \eta \leq \frac{\pi}{2} \quad (2)$$

where  $\omega$  and  $\eta$  are spherical coordinates respectively for the horizontal and vertical curves. Spherical product is defined as:

$$m(\eta) \otimes h(\omega) = \begin{bmatrix} m_1(\eta) \cdot h_1(\omega) \\ m_1(\eta) \cdot h_2(\omega) \\ m_2(\eta) \end{bmatrix} \quad (3)$$

For SQ-representation the known explicit formula [11] is used:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_1 \cos^{\varepsilon_1} \eta \cos^{\varepsilon_2} \omega \\ a_2 \cos^{\varepsilon_1} \eta \sin^{\varepsilon_2} \omega \\ a_3 \sin^{\varepsilon_1} \eta \end{bmatrix} \quad (4)$$

where  $x, y, z$  - SQ coordinate system;  
 $a_1, a_2, a_3$  - scale parameters of the object;  
 $\varepsilon_1, \varepsilon_2$  - object shape parameters;  
 $\omega, \eta$  - spherical coordinates;

The implicit SQ formulation is anyway more suitable for mathematical modeling:

$$\mathbf{F}(x, y, z) = \left( \left( \frac{x}{a_1} \right)^{2/\varepsilon_2} + \left( \frac{y}{a_2} \right)^{2/\varepsilon_2} \right)^{\frac{\varepsilon_2}{\varepsilon_1}} + \left( \frac{z}{a_3} \right)^{2/\varepsilon_1} \quad (5)$$

## 4.1 Curvature estimation

As shown in (3), if the generative curves of the spherical product are continuous, the surface created is also continuous. This means that it is possible to use differential geometry to retrieve curvature information about the parametric surface. Definition (5) describes the different surfaces, so that, according to the scale and shape parameters, every point on the shape is analytically known. The leading idea of this work is to use the relation between the curvature and the object change of appearance. For this reason the accurate evaluation of the curvature is crucial. In this work we followed the approach of Ron Goldman's work on the calculation of surfaces curvature [7], especially focusing on mean curvature and normal directions.

### 4.1.1 Identification of the Normals

Normal can be easily calculated starting from the evaluation of the gradient of the parametric surfaces. Since the gradient of  $\mathbf{F}(x, y, z)$  is perpendicular to the level curves  $\mathbf{F}(x, y, z) = \text{const}$ , the gradient  $\nabla \mathbf{F}$  is parallel to the normal of  $\mathbf{F}(x, y, z) = 0$ . Therefore we have the following formulas:

$$\mathbf{N}(x, y, z) = \frac{\nabla \mathbf{F}(x, y, z)}{|\nabla \mathbf{F}(x, y, z)|} \quad (6)$$

where  $\mathbf{N}$  is a set of unitary vectors in normal direction.

### 4.1.2 Mean Curvature

Usually mean curvature is taken as the divergence of unit vector:

$$\mathbf{K}_M(x, y, z) = -\nabla \cdot \mathbf{N}(x, y, z) \quad (7)$$

This formulation is very compact, but it may be computationally hard to handle so therefor formulation is suggested:

$$\mathbf{K}_M = \frac{\nabla F \cdot H(F) \cdot \nabla F^T - |\nabla F|^2 \text{Trace}(H)}{2|\nabla F|^3} \quad (8)$$

## 4.2 SuperQuadratics Representation

The representation of SuperQuadratics is obtained from its explicit formula 4. According with the scaling and shape parameters it is possible to obtain the coordinates of each point by varying the spherical coordinates  $\omega$  and  $\eta$ . Unfortunately for an equally spaced sampling of these coordinates does not correspond and equally spaced sampling of  $x, y, z$ . To take this problem we followed the approach described in [16]. The idea is to model the surfaces like *Superellipsoids* to obtain a linear arc-length parameterization in order to provide a regular sampling along the surface.

The following formulation regards to equally spaced samples along the transversal arc  $\eta$ :

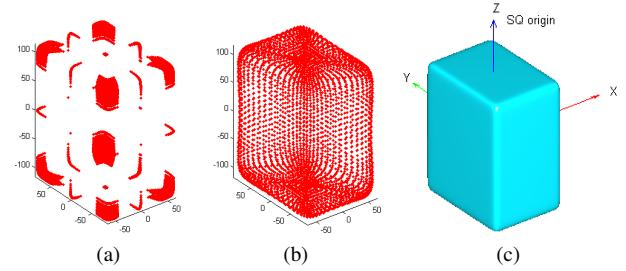
$$\begin{cases} \mathbf{x}_s = x \left(1 + \frac{k_1}{c^2} z^2\right) \left(1 + \frac{k_2}{b_T^2} y^2\right) \\ \mathbf{y}_s = y \frac{b a_T}{a b_T} \left(1 + \frac{k_1}{c^2} z^2\right) \left(1 + \frac{k_2}{a_T^2} x^2\right) \\ \mathbf{z}_s = z \left(1 + \frac{k_1}{a^2} a_T^2\right) \end{cases} \quad (9)$$

where:

$$\begin{cases} \theta_T &= \arcsin\left(\frac{z}{c}\right) \\ a_T &= a \cdot \cos(\theta_T) \\ b_T &= b \cdot \sin(\theta_T) \end{cases} \quad (10)$$

With this representation an equal distributed surface is available.

Referring to figure 3 the number of points for the first two representations is the same, but their distributions are much more homogeneous.



**Figure 3:** The Superquadric representations: (a) - mapping from the points calculated by explicit SQ form; (b) - mapping from a transformed ellipsoid with algebraic manipulations (9); (c) - the surface mapping.

## 5. MATCHING ALGORITHM

The image is elaborated by the object detection algorithm (par. 3.3) which detects the object on the image plane defining a confidence region by means of bounding box. This information is the starting point to retrieve the object position in 3D. By means of the bounding box dimensions is in fact possible to roughly estimate the distance.

Position and orientation of the object in 3D space relative to the camera is parametrized with a homogeneous transformation matrix containing the rotation matrix  $\mathbf{R}$  and the displacement from the origin  $\mathbf{T}$ .

$$\begin{bmatrix} \mathbf{P}_{3 \times 1}^W \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{T}_{3 \times 1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{P}_{3 \times 1}^{SQ} \\ 1 \end{bmatrix} \quad (11)$$

Using a pinhole camera model with lens distortion compensation [12], it is possible to re-map the model from 3D space to image coordinates. In other words, starting from the 3D points with their attributes (e.g. curvature, normals, and light appearance) it is possible to match with the corresponding candidate pixel that is a function of the coordinates transformation matrix. The goal is to find the homogeneous transformation that best matches the above attributes.

### 5.1 Curvature matching

It is well known that strong variations on images gradients are located along edges and corners. Methods that rely only on high image gradients along the edges while neglecting the small variations inside the surfaces, become instable if borders are partially occluded. As already said the SQ formulation allows to obtain a continuous surface, so that edges and corners are modelled as local high curvature. According to the edge based methods this high curvature can be related with high gradient intensity, on the other side also low curvature can be connected with low gradient magnitude regions. With our method is possible to use both informations. From equation (8) it is calculated the mean curvature for each sampled point (9). In the image domain it is possible to evaluate the gradient. The distance between gradient and curvature images is evaluated as vector norm of the matrix difference. The lower the distance, the better the matching.

$$\mathbf{E}_f = \frac{\sum_{i=1}^n ||G^N(i) - K^N(i)||^2}{n} + ||\mathbf{G}^N - \mathbf{K}^N||^\infty \quad (12)$$

$G^N$  and  $K^N$  are the normalized values for the mean curvature and the gradient. The norms to infinity count the maximum displacement between curvature and gradient.

## 5.2 Light appearance matching

Directions of normals are known for each point from equation (6). If it is also supposed known the position of the light source that makes the object illumination appearance available. The known light position provides the information about bright and dark sides. We used the light model of Phong [20]. It provides a spot light source and includes a combination of diffuse and specular reflection.

$$\begin{aligned} I_p &= k_a i_a + (k_d (\mathbf{L} \cdot \mathbf{N}) i_d + k_s (\mathbf{R} \cdot \mathbf{V})^n i_s) \\ \mathbf{R} &= 2(\mathbf{L} \cdot \mathbf{N})\mathbf{N} - \mathbf{L} \end{aligned} \quad (13)$$

$k_a, k_d, k_s$  are the ambient, diffuse and specular constants;  $i_a, i_d$  and  $i_s$  are the respective light intensities.  $\mathbf{L}$  is the light direction vector,  $\mathbf{R}$  is the specular reflection vector,  $\mathbf{V}$  is the viewer direction, and  $\mathbf{N}$  is the surface normals. With this information is possible to compare the expected illumination appearance with the pixel intensities. This time the key idea is to compute the convolution directly among the images. Illumination model (13) is built so as illumination values  $I_p$  span from 0 to 1, same as for the gray levels  $L_g$  of the normalized image. To compute the convolution the mean value is subtracted in order to have a zero-centred distribution. The value for the *normalized* convolution is expressed by the formula:

$$\mathbf{C}_f = \sum_{i=1}^n \frac{I_p^N(i) \cdot L_g^N(i)}{\|I_p^N\|^2 \cdot \|L_g^N\|^2}; \quad (14)$$

The higher the convolution the better the matching.

## 5.3 Normal and gradient alignment

Another method to improve the matching between the static image and the re-projected model is to check the alignment of normals onto image plane with the gradient pixels orientation.

It is well known that along edges the gradient magnitude is higher and its orientation is orthogonal to the border. Normal directions are 3D data; anyway if we project, as in the case of the sampled 3D SQ model, we obtain normals on the image plane. Normals will have the same orientation as the image's gradient.

With this assumption the matching is computed by the angle between silhouette normal  $\mathbf{V}_n$  vectors on the image and the gradients orientation  $\mathbf{V}_g$ . If the scalar product is considered, the cosine of  $\theta_{ng}$  is close to 1:

$$\mathbf{V}_f = \frac{1}{n} \sum_{i=1}^n \text{If}(|\cos(\theta_{ng}(i))| > 0.9, 1) \quad (15)$$

Quality of matching is estimated counting the number of approximately parallel vectors (15).

## 6. OPTIMIZATION

All the contributions (12 - 14 - 15) are mixed into cost function (18). Practically we recognized that the problem is multi-modal so square regression algorithm and other gradient based techniques are not well suited as far as their result is dependent on the initial conditions that easily leads to missing the global minimum. That is why we implemented PSO (Particle Swarm Algorithm) to randomize the starting point [13].

The algorithm starts from a set of agents, call particles, that move along the trust region. This region is initialized by the centroid of HOG's detection window  $\mathbf{X}_n$  and its area  $A$ . From this data we can infer a guess position  $\mathbf{X}_w$  for the object, with the formula:

$$\mathbf{X}_w = \frac{k}{A} \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix} \quad (16)$$

where  $k$  is a fixed proportional factor. The reaching of minimum is leaded by force  $\mathbf{F}_k^{(s)}$  that can be divided into two contributions. The first is relative to *cognitive behavior*  $\mathbf{F}_k^{(s)}(\mathbf{p}_k^{(s)})$  focused on the single particle experience (personal best  $\mathbf{p}^{(s)}$ ). The second is the *social behavior*,  $\mathbf{F}_k^{(s)}(\mathbf{g}_k)$  focused on the swarm attitude (global best  $\mathbf{g}_k$ ). The update of particle's position is obtained through a velocity with the following expression:

$$\mathbf{v}_{k+1}^{(s)} = \omega \cdot \mathbf{v}_k^{(s)} + \underbrace{\frac{C_1 r_1 (\mathbf{p}_k^{(s)} - \mathbf{x}_k^{(s)})}{\mathbf{F}_k^{(s)}(\mathbf{p}_k^{(s)})}}_{\text{Cognitive Term}} + \underbrace{\frac{C_2 r_2 (\mathbf{g}_k - \mathbf{x}_k^{(s)})}{\mathbf{F}_k^{(s)}(\mathbf{g}_k)}}_{\text{Social Term}} \quad (17)$$

Where  $C_1, C_2$  are the acceleration coefficients,  $\omega$  is the inertial weight, and  $r_1, r_2$  are random variables;  $k$  is the iteration step and  $(s)$  is particle's index.

The goal is to achieve the transformation matrix  $\mathbf{M}^*$ . This represents the transformation from camera to object frames. Cost contributions depend on the transformation of the SQ model onto the image plane. Minimizing the cost function  $\mathbf{Fc}$  we find iteratively the matrix  $\mathbf{M}$  with the best matching.

$$M^* = \min_M \Rightarrow \frac{\mathbf{E}_f(M)}{\mathbf{C}_f(M) \cdot \mathbf{V}_f(M)} \quad (18)$$

The localization of the object is supposed on a plane known with a non negligible level of accuracy. The camera position relative to the plane is known by the calibration process by placing one of the reference target acquisitions on the plane. Localization is performed in  $xy$  direction and orientation around  $z$ -axis. Because of plane's uncertainty along  $z$  the search is allowed also along that direction but for a limited displacement. The parameters that the optimizer tries to retrieve are displayed in table 1, with the trust region upper and lower bound settings.

As already stated, multi-modality introduces local minima is-

$\theta_z$	$X_w$	$Y_w$	$Z_w$
$-\pi/2$	$X_w -  X_w  \cdot 0.3$	$Y_w -  Y_w  \cdot 0.3$	$Z_w - 0.02$
$\pi/2$	$X_w +  X_w  \cdot 0.3$	$Y_w +  Y_w  \cdot 0.3$	$Z_w + 0.02$

**Table 1:** Optimization parameters initialization (first row is lower bound, second row is upper bound). Angles are expressed in radians and translations - in meters.

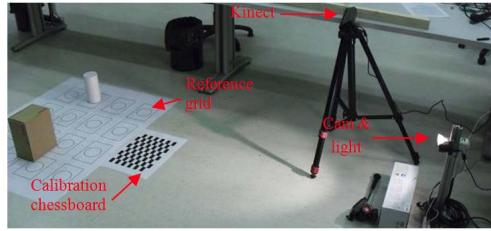
sues. The use of normals alignment (15) and light (14) together with curvature decrease the failure rate. Anyway there are still situations in which the optimizer is “trapped” into a local minima.

## 7. EXPERIMENTAL RESULTS

In this paragraph some results are shown with the aim to prove detection correctness and to provide a quantitative estimation of the localization accuracy in different conditions. We also propose a comparison between our method and state of the art algorithms that use RGB-d camera. For this we chose Microsoft Kinect sensor for its high quality to cost ratio.

The setup was assembled with an RGB camera (1280 x 960) and a Kinect sensor (ref. to figure 4). The camera was equipped with a known position spot light. Both sensors were calibrated in reference to the chessboard (origin on the top left corner). The localization algorithm was limited to the plane estimated with the calibration process.

A properly designed grid has been printed and used as a reference for calibration. This grid is arranged in 25 cells organized with a fixed displacement and variable attitude. For each cell we place the objects with a relative position accuracy of about 1 mm. We also know the absolute position with respect to the camera and the Kinect thanks to the calibration. The localization's range is spanned from 2 to 3 m.

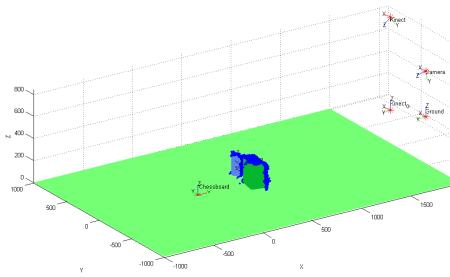


**Figure 4:** Experiment setup layout, data are acquired both from the camera and Kinect. Chessboard is used to estimate camera and Kinect positions in order to provide the reference frame for the localization.

Kinect uses IR structured light to provide full 3D data information. From each snapshot it retrieves a corresponding points cloud with colour information. The idea is to use the cloud of 3D points to fit the SQ model.

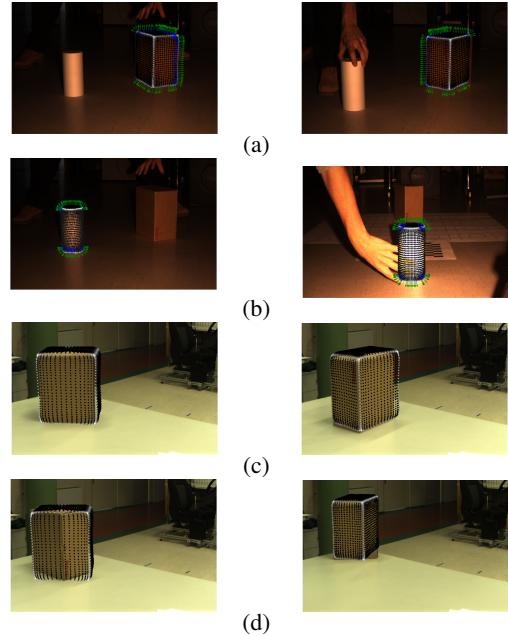
A very common approach is to use robust fitting algorithm (e.g., RANSAC), anyway the results of these depend upon the number of inliers and outliers allowed and its related parameters, this particularly when the number of outliers is large. To overcome this is preferable to cluster the complete cloud of points to segment only those corresponding to the object and then use a regression algorithm to estimate a more accurate fitting over the selected points.

First the initial cloud of points is processed with RANSAC to



**Figure 5:** Example of SuperQuadratics fitting with depth data.

delete those lying on the floor plane (green in the picture). Then clustering of the 2 objects in the view is obtained with k-means. For each cluster is performed a fitting with Levenberg-Marquardt. Final results are shown in figure 5. Figure 6 presents some localization results of objects in general positions and also with occlusions. Figure 6 also presents different conditions, with known light source and without. In the same figure are also reported some local minima solutions. In those cases is possible to note an overlap of the figure but the orientation is completely wrong despite good normals alignment and curvature matching. Figure 7 shows the estimated uncertainty ellipses (green - for Kinect, blue - for our method). Each experimental sample is computed with respect to the reference positions of each SQ centre on the reference grid. Y-direction is aligned with the camera/Kinect axes. The ellipses are estimated using the k factor defined in [23] with a confidence level of 95%. To quantify uncertainty we report the eigenvalues along principal directions for each ellipse (values in mm)  $\lambda_1^B = 18.85$  and  $\lambda_2^B = 27.23$  (box) and  $\lambda_1^C = 11.35$  and  $\lambda_2^C = 13.55$  (cylinder) for our localization respect to  $\lambda_1^B = 11.65$  and  $\lambda_2^B = 40.56$  (box) and  $\lambda_1^C = 13.83$  and  $\lambda_2^C = 47.05$  (cylinder). The figure 8 shows the dependence of the optimized cost function values on the percentage (relative) errors. Unfortunately we found no clear correlation between errors and cost function. This we believe is an important point to further analyze in order to find an estimate of the actual uncertainty directly from the final (optimized) value of the cost function.



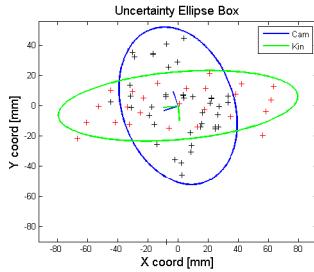
**Figure 6:** In (a) are presented some localization results in general locations; in (b) the same is proposed for cylinders. In the second picture a partially occluded scene is proposed while the localization remains correct. In pictures (c) and (d) localization is estimated without light informations. Obtained results are quite good in (c), but in (d) the localization faces local minima problems which stuck the optimization.

## 8. CONCLUSION

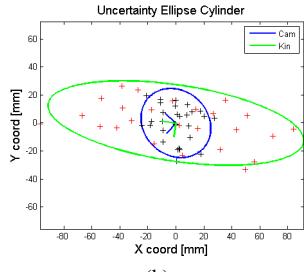
The article presents an innovative approach for monocular object localization. The object detection from the image has been made with HOG algorithm in a pre-processing stage. Then 3D localization is performed by a cost-function that considers: a) the matching of the object reprojected curvature with the image gradient; b) the convolution with object-Light appearance and the image gray levels; c) the quantification of the reprojected contours aligned with the image gradients. The 3D pose estimation accuracy has been quantified with a calibration grid giving the eigenvalues of the uncertainty ellipses identifying the dispersion of data along principal directions. These data are also compared to Microsoft Kinect. Our method gives an object localization accuracy comparable and in some cases even better using only a single camera. The algorithm is competitive in cluttered scenes as it relies on the whole object in contrast to only edges as in the case of edge/gradient methods. Figure 8 represent the limit of detection: for box the percentage error is around 2-2.5% and for cylinder is around 1.5%. Limitations are the knowledge of the light direction and the homogeneity of the surface texture. Nevertheless, in industrial fields for example, the method can cover a broad spectrum of applications. Also outdoor, where Kinect cannot be exploited, the use of a simple camera in combination to our method could represent a good alternative. Optimization is a tricky aspect. Especially the correct orientation retrieval generate many minimal local problems (ref. figure 6.d). It is important to point that the lowest values of the cost function always lead to correct localization while higher values are correlated to local minima. We are therefore confident that, automatizing the optimization process, it is possible to cope with multi-modal problems.

## AKNOLEDGMENTS

The authors are very grateful to colleagues from Mechatronics dep. (University of Trento) and EU grants, incl. Marie Curie-COFUND-Trentino postdoc program, 2010-2013.



(a)

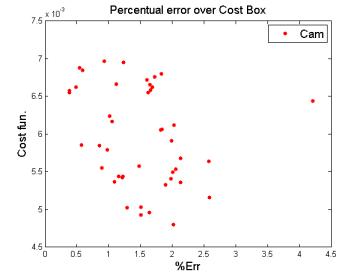


(b)

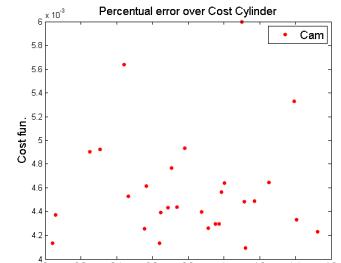
**Figure 7:** The comparison between the ellipses of uncertainty (green - for Kinect, blue - for our method) for the box (a) and the cylinder (b) localizations according to the reference (calibration) grid.

## 9. REFERENCES

- [1] Ilya Afanasyev, Massimo Lunardelli, Nicolò Biasi, Luca Baglivo, Mattia Tavernini, Francesco Setti, and Mariolino De Cecco. 3d human body pose estimation by superquadrics. In *VISAPP (2)*, pages 294–302, 2012.
- [2] A.H. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1:11–23, 1981.
- [3] G. Biegelbauer and M. Vincze. Efficient 3d object detection by fitting superquadrics to range image data for robot’s object manipulation. In *2007 IEEE International Conference on Robotics and Automation*, pages 1086 – 1091, april 2007.
- [4] Alvaro Collet, Dmitry Berenson, Siddhartha S. Srinivasa, and Dave Ferguson. Object recognition and full pose registration from a single image for robotic manipulation. In *ICRA*, pages 48–55. IEEE, 2009.
- [5] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):932 –946, jul 2002.
- [6] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88(2):303–338, June 2010.
- [7] Ron Goldman. Curvature formulas for implicit curves and surfaces. *Comput. Aided Geom. Des.*, 22(7):632–658, October 2005.
- [8] Robert Hanek, Thorsten Schmitt, Sebastian Buck, and Michael Beetz. Fast image-based object localization in natural scenes. In *In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2002*, pages 116–122, 2002.
- [9] Ilie S. Sturm P. Navab N. Fua P. Hinterstoisser S., Cagniart C. and Lepetit V. Gradient response maps for real-time detection of textureless objects. In *IEEE Trans. on PAMI V.34(5)*. P. 876-888, 2012.
- [10] I. Huerta, A. Amato, J. González, and J. J. Villanueva. Fusing edge cues to handle colour problems in image segmentation. In *Proceedings of the 5th international conference on Articulated Motion and Deformable Objects*, AMDO ’08, pages 279–288, Berlin, Heidelberg, 2008. Springer-Verlag.
- [11] Aless Jaklvc, Alevs Leonardis, and Franc Solina. *Segmentation and Recovery of Superquadrics*, volume 20 of *Computational imaging and vision*. Kluwer, Dordrecht, 2000. ISBN 0-7923-6601-8.
- [12] Fryer J.G. Lens distortion for close-range photogrammetry. *Photogrammetric engineering and remote sensing.*, pages P.30–37, 1986.
- [13] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942 –1948 vol.4, nov/dec 1995.
- [14] Jaka Krivic and Franc Solina. Superquadric-based object recognition. In *Proceedings of the 9th International Conference on Computer Analysis of Images and Patterns*, CAIP ’01, pages 134–141, London, UK, UK, 2001. Springer-Verlag.
- [15] Bouthemy P. Marchand E. and Chaumette F. A 2d-3d model-based approach to real-time visual tracking. *Image and Vision Computing.*, V.19(13):P. 941–955, 2001.
- [16] Eugenia Montiel, Alberto S. Aguado, and Ed Zaluska. Surface subdivision for generating superquadrics. *The Visual Computer*, 14(1):1–17, 1998.
- [17] Bill Triggs Navneet Dalal. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*, 2005.
- [18] Bill Triggs Navneet Dalal. Object detection using object histogram of oriented gradients. 2006.
- [19] David McAllester Pedro F. Felzenszwalb, Ross B. Girshick and Deva Ramanan. Object detection with discriminatively trained part based models. *Computer Society Conference on Computer Vision and Pattern Recognition.*, V.32(9).P. 1627–1645, 2010.
- [20] Bui-Tuong Phong. Illumination for Computer Generated Pictures. 18(6):311–317, 1975.
- [21] Christian Schmaltz, Bodo Rosenhahn, Thomas Brox, and Joachim Weickert. Region-based pose tracking with occlusions using 3d models. *Mach. Vis. Appl.*, 23(3):557–577, 2012.
- [22] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. *Comput. Graph. Forum*, pages 214–226, 2007.
- [23] Randall C. Smith and Peter Cheeseman. On the representation and estimation of spatial uncertainty. *Int. J. Rob. Res.*, 5(4):56–68, December 1986.
- [24] Fan-Tung Wei, Sheng-Ting Chou, and Chia-Wen Lin. A region-based object tracking scheme using adaboost-based feature selection. In *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, pages 2753 –2756, may 2008.



(a)



(b)

**Figure 8:** The dependence of the optimized cost function values on the errors percentage for the box (a) and the cylinder (b). Percentage error is the ratio between object distance and position errors. Good matching are considered under 0.006 value of the cost function.

# Irradiance Cache for a GPU Ray Tracer

Vladimir Frolov<sup>(1,2)</sup>, Konstantin Vostryakov<sup>(2)</sup>, Alexander Kharlamov<sup>(2)</sup>, Vladimir Galaktionov<sup>(1)</sup>,

(1) Keldysh Institute of Applied Mathematics (Russian Academy of Sciences), Moscow, Russia; (2) Nvidia



**Figure 1.** The presented screenshots were rendered at 1920x1200 resolution on a GTX 560 HW under 5 minutes using Irradiance Caching (IC) technique. We achieved from 5 to 15 times acceleration compared to our naive path tracing implementation.

## Abstract

This work proposes a GPU friendly irradiance caching (IC) solution, where performance critical parts of an irradiance cache algorithm are done completely on the GPU. We discuss some practical problems arising in the implementation of GPU irradiance caching, and propose solutions for them. The modified algorithm for the GPU is different from a CPU implementation in 2 ways. The first distinction is a multi-pass construction of irradiance cache followed by a final rendering stage and the second distinction is to insert a large record set at once instead of one by one, as used in traditional approaches. We also consider some details to efficiently implement look-up operations on the GPU.

**Keywords:** GPU, Irradiance Cache, Global Illumination.

## 1. INTRODUCTION

For the last decade Graphics Processor Units (GPUs) have made a great advance in performance and have become fully programmable processors. Several commercial GPU photorealistic renderers are available today. Most of them use unbiased path tracing methods in order to minimize intermediate data creation (photon maps, lightcuts, etc). This results in tracing up to ten times more rays than biased alternatives. Moreover, path tracing of complex scenes suffers from highly irregular workload (per ray) and memory access tends to be random. These issues lead to inefficient hardware utilization. On the other hand, although biased approaches have lower complexity they are more difficult to implement on the GPU.

Our paper illustrates the research that we have performed for GPU accelerated biased rendering via irradiance caching and path tracing techniques. The key results of this research are:

- A new GPU friendly IC generation algorithm, performed before final render pass.
- Introduction of a new method for records insertion into octree-based cache for fast irradiance interpolation on the GPU.

Our main contribution is a high quality IC solution that provides from 5 to 15 times acceleration (with an average PSNR of 40 compared with a GPU accelerated path tracing).

## 2. RELATED WORK

### 2.1 GPU ray-tracing

Although fast GPU ray tracing for complex scenes is still a challenge we do not focus on ray tracing acceleration in this paper. Aila and Laine's work [Aila and Laine 2009] provides comprehensive performance analysis of ray tracing on the GPU. Our ray tracing implementation has approximately the same performance on diffuse rays (i.e. rays, randomly shot from the single point over the hemisphere) although it's of 1.5x factor slower for coherent frustum tracing. However we have found that path tracing rays after several bounces can be from 2 to 4 times slower (than diffuse rays) – this is the case of poor HW utilization due to random memory access and non-uniform workload.

### 2.2 CPU Irradiance cache

Irradiance caching decreases the overall cost of indirect illumination computation by performing full hemisphere sampling (or final gathering) only at selected points in the scene, caching the results, and reusing the cached indirect illumination values between those points through interpolation. It was introduced in [Ward et al. 1988]. The algorithm can be summarized as follows:

```
if interpolation is possible then
    reuse cached values through interpolation;
else
    compute new value;
    store it in the cache;
end if;
```

The number of irradiance cache points is usually of 1 - 2 orders of magnitude less than the number of pixels – so the irradiance cache is quite efficient and it can greatly speed up the whole rendering. However, IC is a challenging algorithm, even on a CPU. It has a lot of issues and heuristic approaches that make it practical and suppress its artifacts [Krivánek et al. 2008]. A well-known irradiance cache algorithm [Ward et al. 1988, Krivánek et al. 2008] cannot be implemented on a GPU in a straightforward way because of its serial nature:

- Trace one ray
- Evaluate and insert one record at given “transaction”.

It is difficult to parallelize irradiance cache on multi-core CPUs, although there are several papers available regarding parallel irradiance caching on CPU ([Debattista et al. 2006, Dubla et al. 2009]. These papers focus on solving the problem of sharing

irradiance cache data structure between different CPU threads and different machines (cluster systems).

However, the problem is not only in data sharing between threads and redistributing computational resources (for example between rendering and IC records evaluation), but it also in the fact that IC depends on the records insertion order. For example, irradiance gradients [Krivanek et al. 2008] rely on a serial records insertion. If one places two records in parallel near each other, the gradients and, as a result, validity radiiuses of these records will be different compared to serial insertion. The more threads run in parallel, the more serious this problem becomes so we introduce validity clamping heuristics to solve it in our GPU implementation.

PBRT 2.0 [Pharr and Humphreys 2010] has multithreaded implementation of IC. It does the first pass to compute the cache and the second one to render final image. This approach needs to be refined for the case of massive parallelism.

### 2.3 GPU Irradiance cache

GPU irradiance caching was introduced in [Gauton et al. 2005] and described in details in [Krivanek et al. 2008]. These papers mainly focus on replacing irradiance interpolation via octree lookups with splatting to avoid traversing hierarchical structures on GPUs. The approach used in [Gauton et al. 2005] can be used for primary rays or interactive visualization in computer games, however, it has one serious limitation: only one light bounce can be evaluated either for hemisphere sampling or for final rendering. Thus, it will be hard to have precise photorealistic result with this approach. Besides, it was done mainly for rasterization based engines and cannot be combined directly with a GPU path tracer.

Wang et al. [Wang et al. 2009] presents an efficient approach for global illumination using photon mapping on the GPU. The key aspect of this work is to use irradiance cache with photon mapping and final gathering [Jensen et al. 2002] to quickly compute smooth indirect illumination. Direct lighting is computed using ray tracing and supports hard shadows from point light sources. In this paper irradiance cache point positions are predicted from the geometry discontinuities. Wang's approach to build IC was combined with path tracing in [Frolov et al. 2011] to focus on rendering images with glossy reflections and shadows offline. However, both of these approaches work with geometry term and they use predictive nature without further refinement.

The radiance hints method introduced in [Papaioannou 2011] is a stable (for animation) and a fast solution for diffuse global illumination. The method is based on grid based radiance caching with reflective shadow maps and can handle multiple light bounces. This method works for interactive rendering with view-independent algorithm so it can't control image error that is strictly needed for photorealistic rendering. Besides, using regular grid will not allow one to have high precision with reasonable memory consumption.

## 3. SUGGESTED APPROACH

Similar to PBRT 2.0 our algorithm consists of 2 main phases. The first phase is “irradiance cache creation” and the second phase – “final rendering”. The goal of the first phase is to generate a set of irradiance cache points that will completely cover the space where future samples can occur. This separates computing irradiance cache from using it.

### 3.1 Creation of irradiance cache

IC creation process consists of multiple passes (20-30 passes, the maximum number is user controlled). It can be summarized in the following pseudo code:

```

procedure Create_IC(ic : out Irradiance_Cache) is
    geomDiscMap : Image;
    irradDiscMap : Image;
    discMap : Texture2D;
    candidates : array of IC_Record;
    smallGroup : array of IC_Record;
    candGroups : array of (array of IC_Record);
    iterNum : Integer;
    -- user controlled
    MAX_PASS_NUMBER : Integer := 30;
    MIN_CAND_THRESHOLD : Integer := 100;
begin
    geomDiscMap := CreateGeometryDiscMap();
    discMap := Build2DMipMapChain(geomDiscMap);
    candidates := Dithering(discMap);

    ic.Insert(candidates);
    iterNum := 0;
    candidates.resize(MIN_CAND_THRESHOLD+1);

    while (candidates.size() >= MIN_CAND_THRESHOLD and
           iterNum < MAX_PASS_NUMBER) :

        -- screen space stage
        irradDiscMap := CreateIrradianceDiscMap();
        discMap := Build2DMipMapChain(irradDiscMap);
        candidates := Dithering(discMap);

        -- insert candidates except for pixels
        -- for which we already have records
        ic.Insert({candidates} \ {ic.records});

        -- world space stage
        candidates := SelectIfInterpErrorIsLarge();
        candidates := SortWithZCurve(candidates);
        candGroups := GroupRecords(candidates);

        candidates := [];
        for group in candGroups:
            smallGroup := SelectSeveralCands(group);
            candidates.append(smallGroup);
        end for;

        ic.Insert(candidates);
        iterNum := iterNum + 1;
    end while;

end Create_IC;

```

The ‘Insert’ procedure also evaluates irradiance for each records. We will discuss its implementation later.

Each pass consists of 2 independent stages. The first stage works only for visible points. The second stage works for visible and for points that are not directly visible from the eye. During each pass and within each stage new irradiance cache records are inserted into the cache. The very first pass is different from the others and works with geometry discontinuity like [Wang et al. 2009] and [Frolov et al. 2011] do. The important aspect of the irradiance cache generation process is that a large set of points (several hundred or even thousand points) are selected at once, irradiance for these points are computed on the GPU and these points are added to the cache structure in one transaction.

#### 3.1.1 The very first pass (coarse screen space pass)

In the very beginning of the process of irradiance cache creation we have no information about the scene at all. Owing to this fact, the goal of this pass is to create first approximation of irradiance cache that will be used as a starting point for further passes. Our algorithm tries to predict complexity of different screen parts, using geometry discontinuity maps and image processing. It

attempts to place more records in areas where more of them are needed.

First, we trace rays from the eye position and store hit positions and normals in separate full screen textures. A mip-map pyramid for each of these textures is built. Then, each mip-level of surface discontinuity texture map is evaluated according to this formula:

```
surfDisc := k*normDiff + worldPosDiff;
```

where `worldPosDiff` and `normDiff` is a maximum difference between positions (and normals accordingly) within neighboring pixels (in an appropriate mip-level). And  $k$  is a parameter that depends on the world scale. Having a discontinuity map, we blend all of up-scaled mip levels and perform the dithering algorithm on the resulting image with a binary quantization (i.e. each pixel in resulting image can have a value equal to 1 or 0). The result of this dithering is a binary image - a set of initial points; this initial set is our first approximation of irradiance cache (Fig. 2).

The main idea behind that binary dithering is that it allows us to represent discontinuity maps (both geometry and irradiance) in terms of sparse point set – potential IC records.

#### Dithering Implementation:

Our dithering implementation is deterministic (but we suppose random or combined solution is also possible). For a given part of screen it decides (based on user defined threshold) whether we need to put irradiance cache record in each 32-d, 16-th, 8-th, 4-th or 2-d pixel.

#### 3.1.2 Screen space stage (1)

The same operation but for irradiance discontinuity is performed in subsequent passes – calculate irradiance by fetching it from the cache, build difference image, create mip-map pyramid, blend up-scaled images, perform dithering and insert newly obtained points into irradiance cache. This procedure is repeated several times. Because screen space solution stops producing new points relatively fast, we disable it after several (3-4) passes and continue creating the cache only with the world space stage.

#### 3.1.3 World space stage (2)

The presented screen space algorithm works on primary visible smooth surfaces. However it cannot be used for indirectly visible surfaces and it tends to miss tiny geometric details. The red triangle on Fig. 3 represents viewers' frustum. Red points are not visible from the camera, however secondary rays can still reach such regions.

Our goal is, to generate a set of irradiance cache points that completely cover the space where rays can hit a surface during path tracing process. We used an idea similar to the clustering approach that had been used in [Gassenauber et al. 2011].

Rays are traced from the eye and all hit points are saved on each bounce if interpolation has unacceptable error estimation from geometric considerations. All such points are stored in separate buffer during ray tracing using CUDA 'atomicAdd' operation similar to how DirectX10/11 'append buffer' works.

However, a very large set of points is produced and we need to select a subset of the best candidates from it. We believe several approaches can be applied to form clusters; however we used the simple one that can be easily ported to GPU. At first, we sort candidates according to 3D Z-Curve [Morton 1966]. After that we start inserting points into a cluster (around the first point in the sorted array). However we want to keep the bounding box of the current cluster within certain limits. If after inserting a point the bounding box of the current cluster exceeds the maximum bounding box size, we create a new cluster and continue inserting candidates into it.

Next, it is possible to select a single point from each cluster with maximum error however it is not the best approach. Let us consider a cluster that was formed around the corner of the Cornell Box. The corner consists of 3 walls and if our cluster contains points on every wall, we need to select at least one point on each wall, otherwise we lose useful candidates, because we know that radiance difference usually corresponds to rapid changes of the normal field. So, from each cluster we select several candidates with unique normals and thus, deal with corner cases.

We terminate the creation process when the maximum number of passes is reached or when 'candidates.size()' becomes small enough (this parameter is user defined). Due to the stochastic nature of our IC creation process (world space stage uses random 'path tracing style' rays) on some complex scenes there can be a regions that were not covered by IC records and candidates are still produced. However, if we stop IC creation process in that case, it will not introduce a valuable error in the final image because the probability of rays hit such regions tends to be zero.

#### 3.2 Final rendering

After we have irradiance cache computed we do adaptive path tracing as described in [Frolov et al. 2011] with fetching indirect smooth lighting from irradiance cache. Thus, for fast and smooth indirect lighting we use irradiance caching technique and we use path tracing for other effects, such as soft shadows, glossy reflections and refractions, depth of field and motion blur.

### 4. IMPLEMENTATION AND RESULTS

Our implementation is done using CUDA and C++. All performance critical parts of the algorithm are done in CUDA. However, such things as tree construction and some other algorithms are implemented in C++ on the host.

#### 4.1 Hemisphere sampling

For irradiance computations we use the progressive evaluation algorithm with Monte Carlo path tracing. All irradiance cache records are placed in 'active records list'. For each active record we use a sequence of randomly distributed (but coherent) hemisphere samples – 4096, 16384, 65536 and etc. At first we use 4096 rays for all records in the list. If estimated error for a record is small enough we discard that record from 'list of active records' and process remaining records with 16384 rays (the next value in the sequence). We repeat this process until all records are evaluated or the maximum number of samples per irradiance record is achieved. Using the sequence of pre-generated samples instead of simple random rays is important because we can save rays coherency at least for the first bounce and have valuable speed-up (~ 2-4 times) for ray tracing on GPUs.

To evaluate convergency for a record we use the approach described in [Krivánek et al. 2006] accumulating odd and even partial sums of lighting integral.

We used 'Hammersley' sampling technique described in [Suffern 2007] to cover hemisphere with samples. To have more coherent groups of rays we used stratification (subdivide the hemisphere into sectors and generate  $32*k$  rays for each sector where  $k \geq 1$ ). Actually we just need to group 'Hammersley' points in groups of size  $32*k$ . Initially we do that on the CPU in tangent space. On the GPU we transform directions from tangent to object space to get correct hemisphere sampling.

During the hemisphere sampling, we also calculate initial validity radius for each irradiance cache point using 'sphere split approximation' [Krivánek et al. 2008].

## 4.2 Insertion records into octree

The insertion process is done on the CPU. Our implementation inserts a set of records in one transaction, and we modified the original insertion algorithm, described in [Krivanek et al. 2008] and [Pharr and Humphreys 2010].

```
procedure Insert (
    self      : inout Irradiance_Cache;
    records   : in array of IC_Record ) is
begin
    EvaluateIrradiance(records);
    self.auxOctree.Insert(records);
    ValidityRadiusClamping(records, self.auxOctree);
    self.mainOctree.Insert(records);
end Insert;
```

The problem with inserting multiple records is that in several cases, we can find a large set of closely-located records, with overlapping validity radii. This is a problem, because in those regions octree leafs will contain a large list of points and interpolation becomes slow. This motivated us to develop a special algorithm for decreasing validity radii during insertion. Our insertion consists of 3 phases. First, we consider irradiance cache records as points (not as spheres!) and insert them into an auxiliary octree. This octree will be used to speed-up location of k-nearest points (irradiance cache records).

We call the second step ‘validity radius clamping’. It treats irradiance cache records as spheres. The goal of this step is to decrease validity radius for each point. For each irradiance cache record it locates  $k$  nearest neighbors ( $k$  is 4-7) in ‘different directions’ and if the validity radius of the current point is greater than the distance to the farthest point, the validity radius is clamped to this distance.

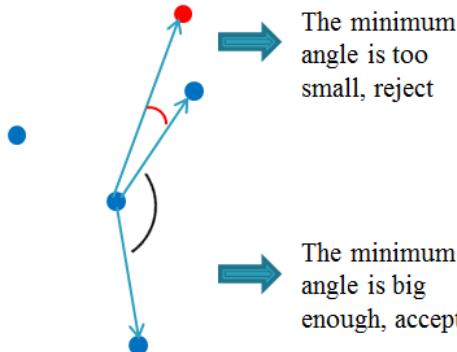


Figure 1: Angle criterion of filtering nearest neighbors.

By ‘different directions’ we mean that while we look for neighboring points we calculate the angle between a new candidate and all the points that we have already found (Fig. 6). If the angle between the direction to a new point and any direction to a point we already found is too small, we do not consider this point, i.e. we do not add it into the nearest neighbors list.

Last we also consider irradiance cache records as spheres. But validity radii of these records were clamped by the previous step. The goal of the last step is to insert all points into the final octree that will be used for fetching irradiance from the cache on the GPU. Validity radius clamping is an important part of the algorithm. Table 1 shows performance improvements gained by introducing our validity clamping approach.

Scene	IC1	IC2	look-up acceleration
-------	-----	-----	----------------------

Teapot	29 ms	5.9 ms	4.9 times
Dragon	18 ms	5.0 ms	3.7 times
Conference	25 ms	4.4 ms	5.6 times
Sponza	16 ms	7.1 ms	2.3 times
Cry-Sponza	33 ms	8.3 ms	4.0 times

Table 1. The column marked IC1 presents time (in milliseconds) required to perform one million look-up operations when validity radius clamping is disabled. The column marked IC2 presents time, required to perform one million look-up operations with enabled validity radius clamping. The last column represents acceleration factor. All measurements were done with GTX560 HW.

Thus, inspired by Krivánek’s Neighbor Clamping, we introduce a new validity clamping radius criterion in order to accelerate look-up operation by means of density control.

## 4.3 Fast Octree Look-Up

We use interpolation formula proposed by Tabellion and Lamorlette in [Tabellion and Lamorlette 2004] and stackless octree look-up as described in [Krivanek et al. 2008]. We have found that the stackless approach is very efficient on GPUs if only several irradiance cache records are stored in octree leaves. The key advantage of the multiple reference octree is the stackless ‘root to leaf’ look-up algorithm. To find all irradiance cache records, which validity radii overlap with a given point, we need to traverse from the root to a leaf and then just iterate through the list of cache points we stored in a leaf.

## 4.4 Results overview

The results of our renderer are presented in Table 2. We target high quality images at 1920x1200 resolution and we used world space irradiance interpolation. As a result our irradiance cache contains a large record set (100K-200K). Due to the high precision requirements we usually start irradiance evaluation with 4096 Monte-Carlo samples. The “Conference Room” scene has 6 area lights under the ceiling and a significant part of rendering time was taken by soft shadows. Due to a weak indirect component in this scene path tracing converges fast enough and acceleration factor is lower (only 3 times) compared to other scenes.

“Sponza” and “Crytek-sponza” scenes (in contrast to Conference Room) have strong indirect illumination and acceleration on these 2 scenes was even higher (14 and 18 times accordingly) than expected. For example, having  $\sim 200K$  records for the last scene, one can’t expect more than  $1920 \times 1200 / 200000 = 11$  times acceleration. However we found that on some complex scenes (even disregarding total triangle count), like “Crytek-sponza” naive path tracing is inefficient and the ray tracing performance is far from 60M rays per second. As a result grouping rays to coherent packets, when sampling hemisphere, provides a great advantage for GPU ray tracing performance and for the “Sponza” and “Crytek-sponza” scenes we have super-linear acceleration.

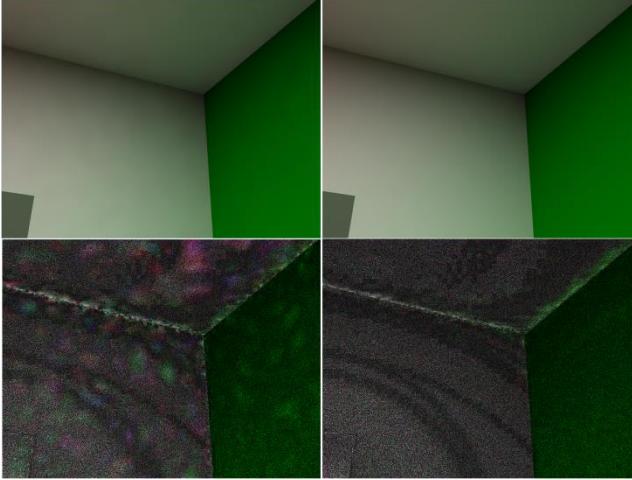
We calculated square error (with ‘The Compressor’ [Compressor] and PSNR (with MatLab) metrics to have a numerical estimation of an image difference. PSNR, for HDR images, is much higher (than for LDR) because absolute value of the signal is higher on HDR images.

## 4.5 Quality discussion and analysis

For our current implementation we used Monte-Carlo path tracing to evaluate irradiance. We start from 4096 hemisphere samples with 2 path tracing bounces and continue to increase the number of samples with our progressive evaluation algorithm. In contrast to evaluating IC records, path tracing requires on average  $\sim 1000$

## 4.5 Quality discussion and analysis

For our current implementation we used Monte-Carlo path tracing to evaluate irradiance. We start from 4096 hemisphere samples with 2 path tracing bounces and continue to increase the number of samples with our progressive evaluation algorithm. In contrast to evaluating IC records, path tracing requires on average ~1000 samples per pixel. The produced noise is high frequency and it is filtered by the human eye. However, when we consider irradiance cache, the error will be splashed over the surface resulting in low frequency noise that appear to a human eye as “dirty spots” (Fig. 3). To suppress these artifacts we use more samples per record.



**Figure 3:** Top row: IC results produced with 1024 and 16384 samples per record. The bottom row: magnified difference (x32) between corresponding IC result and path traced reference.

## 4.6 Bottleneck analysis

We have measured that during irradiance cache construction ~80% of the time is spent on evaluating records (i.e. hemisphere sampling with Monte-Carlo path tracing) and it takes ~50-90% of the total rendering time. One way to reduce this time is to use fewer samples with one bounce. This will work much faster because on the first bounce we have coherent sets of rays (and the noise is less than for 2 or more bounces). However this will prevent us from computing indirect lighting from multiple diffuse bounces. Another choice is to use photon mapping with final gathering [Krivanek et al. 2008] instead of Monte-Carlo path tracing. We believe this idea should give us a performance benefit and we’ll investigate this in our future research. We also think that using recursive irradiance cache [Krivanek et al. 2008] is a promising idea; it allows tracing only coherent set of rays to transport light from one level of the cache to another (or even use rasterization).

We perform interpolation in the world space and as a result the IC generation algorithm places a lot of records near tiny geometry details. We suppose screen space IC should be used for primary visible points.

The octree construction (insertion of records) is not a bottleneck in our implementation; it usually takes ~15% of irradiance cache construction time. The remaining 5% of the time was spent on ray tracing during IC construction, building of discontinuity maps (both geometry and irradiance), data transfers between GPU and CPU.

Regarding the final rendering, the irradiance cache look-up operation takes 20% in average of the total rendering time. The ray tracing (incoherent rays) takes 60-70% of this time and the rest is taken by the shading related works and pipeline overhead.

## 5. LITERATURE

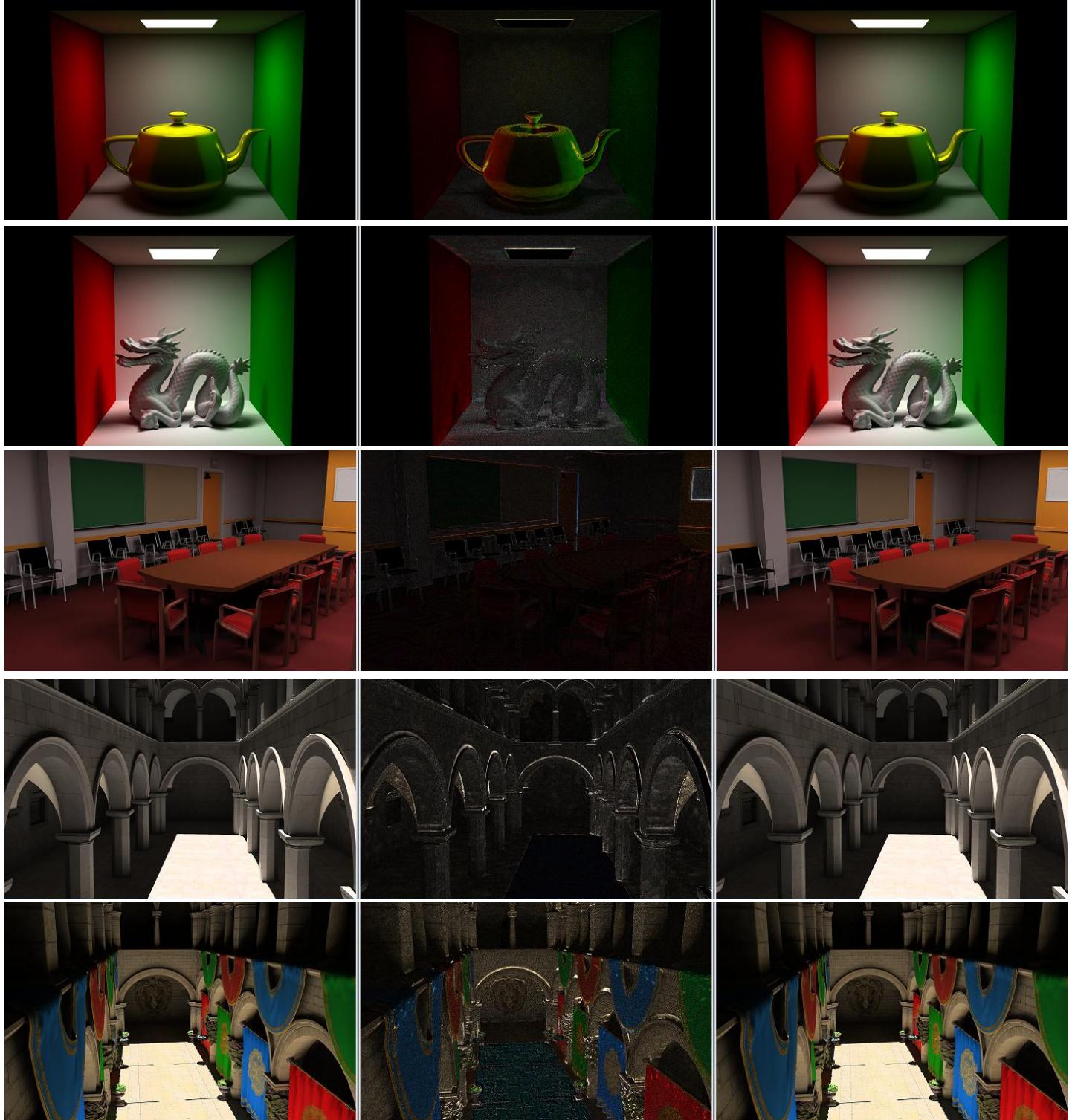
- [1] [Aila and Laine 2009] Aila, T. and Laine, S. 2009. *Understanding the efficiency of ray traversal on GPUs*. In *Proceedings of the Conference on High Performance Graphics 2009* (New Orleans, Louisiana, August 01 - 03, 2009). S. N.
- [2] [Debattista et al. 2006] K. Debattista, L.P. Santos, A. Chalmers, *Accelerating the irradiance cache through parallel component-based rendering*, in: EGPGV2006 - 6th Eurographics Symposium on Parallel Graphics Visualization. Eurographics, May 2006, pp. 27-34.
- [3] [Dubla et al. 2009] Piotr Dubla, Kurt Debattista, Luis Paulo Santos, and Alan Chalmers. *A wait-free shared-memory irradiance caching*. IEEE Computer Graphics and Applications, 2010.
- [4] [Frolov et al. 2011] V. Frolov, A. Kharlamov, A. Ignatenko. “Biased solution of integral illumination via irradiance caching and path tracing on GPUs”. Programming and Computer Software. Volume 37, Number 5 (2011), 252-259, DOI: 10.1134/S0361768811050021
- [5] [Gauton et al. 2005] Pascal Gautron, Jaroslav Krivánek, Kadi Bouatouch, and Sumanta Pattanaik. *Radiance cache splatting: A GPU-friendly global illumination algorithm*. In *Proceedings of Eurographics Symposium on Rendering*, June 2005.
- [6] [Gassenbauer et al. 2011] Václav Gassenbauer, Jaroslav Krivánek, Kadi Bouatouch, Christian Bouville, Mickaël Ribardière. *Improving Performance and Accuracy of Local PCA*. 4 NOV 2011 DOI: 10.1111/j.1467-8659.2011.02047.x
- [7] [Jensen et al. 2002] 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).
- [8] [Krivánek et al. 2006] Jaroslav Krivánek, Kadi Bouatouch, Sumanta Pattanaik, Jiríára. *Making Radiance and Irradiance Caching Practical: Adaptive Caching and Neighbor Clamping*. Eurographics Symposium on Rendering, 2006.
- [9] [Krivanek et al. 2008] Krivánek, 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.
- [10] [Manfred and Gunther 2007] Manfred Ernst, Gunther Greiner. *Early Split Clipping for Bounding Volume Hierarchies*, Proceedings of the 2007 IEEE Symposium on Interactive Ray Tracing, p.73-78, September 10-12, 2007 doi=10.1109/RT.2007.4342593
- [11] [Morton, G. M. (1966) *A computer Oriented Geodetic Data Base; and a New Technique in File Sequencing*, Technical Report, Ottawa, Canada: IBM Ltd.
- [12] [Papaioannou 2011] G. Papaioannou, *Real-Time Diffuse Global Illumination Using Radiance Hint*. Presented at High Performance Graphics 2011, Vancouver, Canada, Aug. 2011.
- [13] [Pharr and Humphreys 2010] Pharr M., Humphreys G.: *Physically Based Rendering: From Theory to Implementation*, 2<sup>nd</sup> edition. Morgan Kaufmann, 2004.
- [14] [Suffern 2007] Kevin Suffern. *Ray Tracing from the Ground Up* A. K. Peters, Ltd. Natick, MA, USA©2007 ISBN:1568812728
- [15] [Tabellion and Lamorlette 2004] Eric Tabellion and Arnauld Lamorlette. An approximate global illumination system for computer-generated films. In *Proceedings of SIGGRAPH 2004*. DOI: 10.1145/1186562.1015748
- [16] [Ward et al. 1988] Ward, G., Rubinstein, F., and Clear, R. 1988. A ray tracing solution for diffuse interreflection. In *SIGGRAPH 1988, Computer Graphics Proceedings*.
- [17] [Wang et al. 2009] 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. ACKNOWLEDGEMENTS

This work was sponsored by the RFFI (Russian Foundation for Fundamental Investigations) grant “MOL\_A 12-01 31027”.

Scene	Number of Triangles	Number of IC records	IC creation time	Render pass time	Total time	Naive path tracing time	Acceleration	Square error (png)	PSNR (png)
Teapot	25612	27356	18s	35s	56s	300s	5.3 times	2.5	45
Dragon	871426	71883	52s	36s	88s	490s	5.6 times	3.4	43
Conference	331191	89336	64s*	108s	172s	500s	2.9 times	2.3	47
Sponza	66456	161245	123s	17s	140s	1980s	14.1 times	3.9	41
Cry-Sponza	262267	245369	228s	20s	248s	4632s	18.6 times	5.7	38

**Table 2:** Test setup. All scenes were rendered in 1920x1200 on GTX560 HW. For path tracing - max samples per pixel was 4000 (however, this number was reached for the 2 last scenes). Image difference and square error were computed with ‘The Compressorator’ tool [Compressorator]. For conference Room we started hemisphere evaluation with 1024 hemisphere samples (instead of usual 4096 samples) (\*).



# Audio-aware on-the-fly Animation from Single Photo

Konstantin Kryzhanovsky, Aleksey Vil'kin, Ilia Safonov, Zoya Pushchina  
 Samsung Moscow Research Center, Russia  
 {k.kryzhanovs, a.vilkin, ilia.safonov, p.zoya} at samsung.com

## Abstract

In this paper we propose new approach of automatic generating **real time content adaptive** animation effects from the still images adapted for the low-powerful embedded HW platforms. Displayed animation behaves uniquely each time it's played back, and does not repeat itself during playback duration, creating vivid and lively impression for the viewer. Adaptation of the effect parameters according to background audio greatly increases aesthetic impression of the viewer. Three animation effects such as *Flashing Light*, *Soap Bubbles* and *Sunlight Spot* are described in details. We propose several ways of controlling the effect parameters by music. User opinion survey demonstrates that majority of users are excited by such effects and wants to see them in their devices with multimedia capability.

**Keywords:** *animation from photo, audio-adaptive effect, multimedia slide-show, attention zones detection.*

## 1. INTRODUCTION

Creation and sharing of multimedia presentations and slideshows has become a pervasive activity. The development of tools for automated creation of exciting, entertaining and eye-catching photo transitions and animation effects, accompanied by background music and/or voice comments, has become a modern trend [1]. One of the most impressive effects is the animation of still photo, for example, grass swaying in the wind or rain drop ripples in the water, etc.

Special interactive authoring tools, such as Adobe After Effects and Ulead Video Studio, are used to create animation from an image. Development of fast and realistic animation effects is hard task itself, and it is a topical problem of modern computer

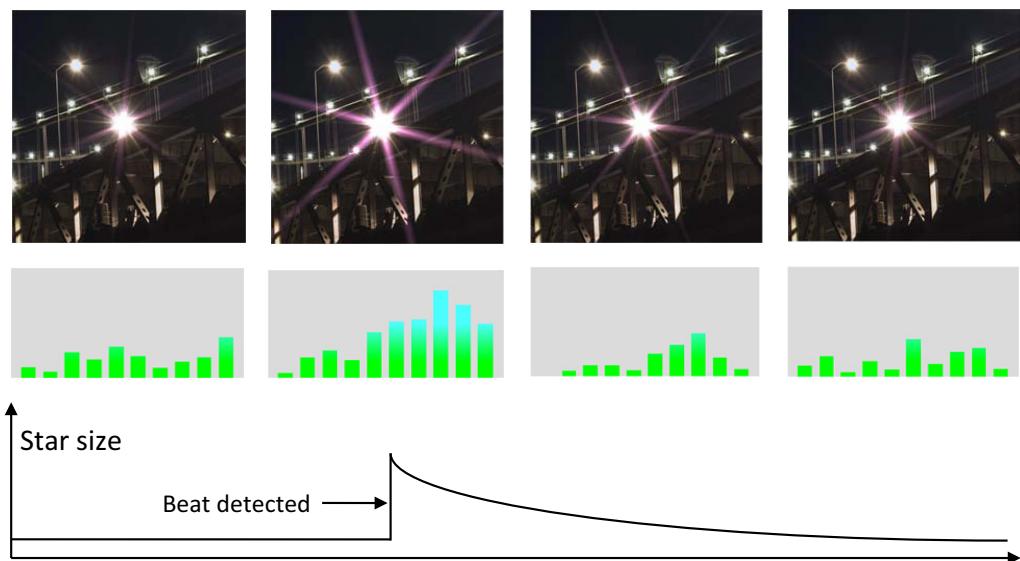
graphics. For example, paper [2] discusses algorithm for generation of plausible motions animation. In authoring tools the effects are selected and adjusted manually, which may require considerable effort from a user. Resulting animation is saved as video clip, thus requiring noticeable amount of space for storage. During playback, such movie will always be the same, thus leading to repetitiveness feeling of the viewer.

For the multimedia presentations and slideshows it's preferable to generate animated effects on-the-fly with a high frame rate. Very fast and efficient algorithms are necessary to provide required performance. It's especially difficult for low-powerful embedded HW platforms.

Our research was defined as development and implementation of automatically generated animated effects of Full HD images on ARM Cortex A8 and A9 – based embedded platforms, with the CPU frequency 800 - 1000 MHz and without use of GPU – based APIs, such as OpenGL. Only ARM commands were available to use, including SIMD instructions of ARM NEON co-processor. Creation of realistic and complex animated effects in such limited conditions is a challenging task itself, particularly for the well experienced in computer games on powerful PCs and play stations users.

We have developed several algorithms for generation of content-based animation effects from still images, such as *Flashing Light*, *Soap Bubbles*, *Sunlight Spot*, *Magnifier Effect*, *Rainbow*, *Portrait Morphing Transition Effect*, *Snow*, *Rain*, *Fog*, etc. For those effects we propose a new approach of automatic audio-aware animation generation.

In the paper we demonstrate our concept, i.e. adaptation of effect parameters according to background audio, for three effects: *Flashing Light*, *Soap Bubbles* and *Sunlight Spot*. Obviously the concept can be extended to other animated effects.



**Figure 1.** Detected beats affect size of flashing light.

## 2. RELATED WORKS

Recently, some content-adaptive automatic techniques for generation of animation from static photo were proposed. Paper [3] describes Animated Thumbnail which is a short looped movie demonstrating main objects of the scene in sequence. Animation simulates camera tracking-in, tracking-out and panning between detected visual attention zones and whole scene.

Music plays an important role in multimedia presentations. There are some methods towards to aesthetical audiovisual composition in slideshow. Tiling Slideshow [4] describes two methods for analysis of background audio in order to select timing for photos and frames switching. First one is beats detection. Second one is energy dynamics, calculated using root mean square values of adjacent audio frames.

Also there are other concepts of combining audio and visual information with the automatic generation of multimedia presentations exists. For example, paper [5] suggests approach that focuses on an automatic sound track selection. The process attempts to comprehend what the photos depict and try to choose music accordingly.

## 3. ANIMATION EFFECTS FROM SINGLE IMAGE

### 3.1 Flashing Light

The *Flashing Light* effect displays several flashing and rotating colored light stars over the bright spots on the image. In this effect, size, position and color of flashing light stars are defined by detected position, size and color of the bright areas on the source still image.

Algorithm performs the following steps to detect small bright areas on the image:

- calculating the histogram of luma channel of the source image
- calculating segmentation threshold as luma level corresponding to specified fraction of brightest pixels of the image using the luma histogram;
- segmenting source image by thresholding; while thresholding, the majority morphological filter is used to filter out localized bright pixel groups;
- calculation of the following features for each connected region of interest (ROI):
  - a. Mean color  $C_{mean}$
  - b. Centroid  $(x_c, y_c)$
  - c. Image fraction  $F$  – fraction of the image area, occupied by ROI;
  - d. Roundness – relation of the diameter of the circle with same area as ROI to maximum dimension of the ROI:

$$K_r = \frac{2\sqrt{S/\pi}}{\max(W, H)},$$

where  $S$  is the area of the ROI and  $W, H$  are ROI bounding box dimensions;

- e. Quality – integral parameter, characterizing the possibility of to ROI to be a light source and calculated as following:

$$Q_L = w_{Y_{max}} \cdot Y_{max} + w_{Y_{mean}} \cdot Y_{mean} + w_R \cdot K_r + w_F \cdot K_F;$$

where  $Y_{max}$  - maximum luma of the ROI,

$Y_{mean}$  - mean luma of the ROI,

$K_F$  - coefficient of ROI size,

$$K_F = \begin{cases} F/F_0, & \text{if } F \leq F_0 \\ F_0/F, & \text{if } F > F_0 \end{cases}, \text{ where } F_0 \text{ - image fraction}$$

normalizing coefficient for an optimal lightspot size;

$w_{Y_{max}}, w_{Y_{mean}}, w_R, w_F$  - weighting coefficients.

Weighting coefficients  $w$  and optimal lightspot size normalization coefficient  $F_0$  are obtained by minimizing differences between automatic and manual light sources segmentation results.

- selection of regions with appropriate features.

All bright spots, with image fraction falling within appropriate range  $(F_{min}, F_{max})$ , and roundness  $K_r$  is larger than certain threshold value  $K_r^0$  are considered as potential light sources. Potential light sources are sorted by their quality value  $Q_L$ . Specified number of light sources with the largest quality is selected as final positions of “light stars” objects.

Centroids of selected light regions are used as star positions. Star size is determined by dimensions of appropriate light region. Mean color of the region determines the color of the light star. Fig. 2. shows an image with bright spots and corresponding light stars.



a)



b)

**Figure 2.** a) Bright spots on the image and b) corresponding light stars.

Every light star is composed from bitmap templates of two types, representing star shape elements: halo shape and star ray (or spike) shape. These templates are alpha maps scaled independently. Examples of templates are shown on Fig. 3.

During rendering, the alpha map of complete star of appropriate size is prepared in separate buffer, and then the star is painted with appropriate color with transparency value extracted from star alpha map.

During animation, light star sizes and intensities are changed gradually and randomly to make an expression of flashing lights.

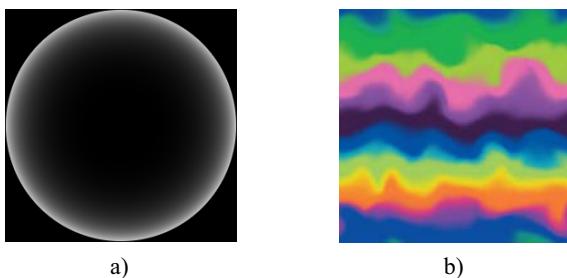


**Figure 3.** Light star shape templates: a) Halo template; b) Ray template.

### 3.2 Soap Bubbles

The effect displays soap bubbles moving over the image. Each bubble is composed from color map, alpha map and highlight map. The set of highlight maps with the different highlight orientation is precalculated for each bubble. Highlight position depends on lighting direction in corresponding area of the image. Lighting gradient is calculated using downsampled brightness channel of the image.

Fig. 4 shows Soap Bubble components. Color map is modulated with highlight map, selected according average lighting direction around the bubble, and then combined with source image using alpha blending with bubble alpha map.

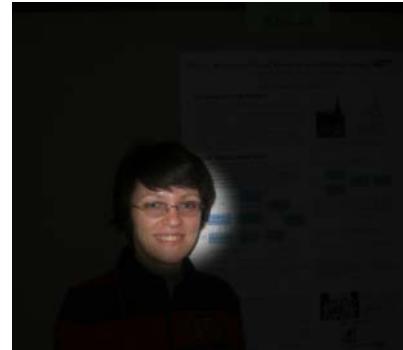


**Figure 4.** Soap Bubble components: a) Alpha map b) Color map.

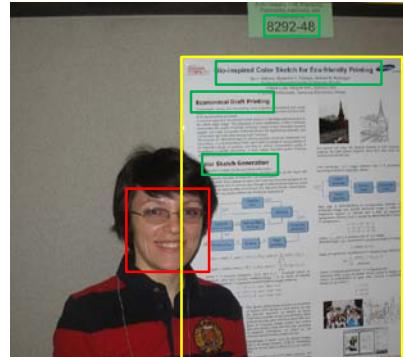
During animation, soap bubbles are moved smoothly over the image from bottom to top or vice versa while oscillating slightly in horizontal direction to make an impression of real soap bubbles floating in the air.

### 3.3 Sunlight Spot

The effect displays bright spot moving over the image. Prior starting effect, the image is dimmed according to its initial average brightness. Fig. 5 shows an image with sunlight spot effect. The spotlight trajectory and size are defined by attention zones on the photo.



a)



b)

**Figure 5.** a) Frame of *Sunlight Spot* effect; b) Selected attention zones.

Similar to many existing publications we find human faces and salient regions using pre-attentive vision model. Basing on these regions we form attention zones. In addition we consider text inscriptions as attention zones too. For example it can be the name of hotel or town on the background of which the photo was made. Or, in case of the newspaper, it will be headlines.

Well-known OpenCV software library contains implementation of face detection for front and profile faces. In general the technique that is based on state-of-the-art Viola-Jones face detector [6] provides good results. However it gets a lot of false positives. The number of false positives can be decreased with additional skin tone segmentation and processing of downsampled image [7]. We have ported OpenCV 2.3 to our embedded platform. Time of face detection for 480x320 images is about 0.8 s.

So far the universal model of human vision does not exist, but pre-attentive vision model based on feature integration theory is well-known. Since in this case, the observer is on attentive stage while viewing photo, a model of human pre-attentive vision is not strictly required. 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. Classical saliency map building algorithms like [9] have a very high computational complexity. That is why researchers recently devote a lot of efforts to develop fast saliency map creation techniques. Paper [10] compares several modern algorithms for salient regions detection. We implemented on our embedded platform *Histogram-based Contrast* (HC) method. The time of salient regions detection for 480x320 images is about 0.1 s.

While developing the algorithm for detection of areas with text, we take into account the fact that text components are ordered the same way and are similar in texture features, color. Firstly, we apply LoG edge detector and restore missing parts using combination of morphological operations. After edge detection we end up with many circuits, which can be ordered as connected tree of objects and voids

Then we filter resulting connected components based on the analysis of the texture features. We use features from [8], as well as analysis of geometric dimensions and relations. We merge closely located connected components, arranged the same order and similar in color, texture features, in groups. Then we classify resulting groups. We form final zones with the text on the basis of groups that are classified as text. Time of text regions detection for 480x320 images is about 0.5 s.

Fig. 5 shows detected attention zones. Red rectangle depicts face detected; green rectangles denote text regions; yellow is bounding box of the most salient area according to HC method.

#### 4. ADAPTATION TO AUDIO

What animation parameters may depend on characteristics of background audio signal? Firstly it is size and intensity of animated objects, also speed of their movement and rotation can be adjusted. In addition, we investigated the question: How can we change color of animate objects, depending on music? Famous Russian composer and pianist Alexander Scriabin about 100 years ago proposed a theory of connection between music and color. Colors corresponding to notes are shown on fig. 6. This theory connects major and minor tonality of the same name.

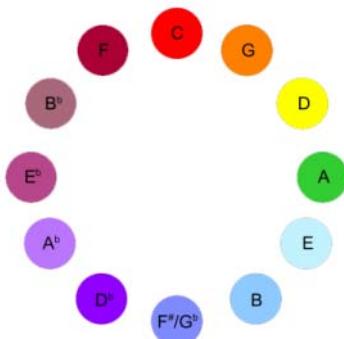


Figure 6. Accords with the circle of fifths corresponding to Scriabin's theory.

On our platform we work with stereo audio signal on frequency 44 kHz. We consider 4 approaches to connect animation of 3 effects mentioned above with background audio. In all approaches we analyze the average of two signal channels in frequency domain. The spectrum is built 10 times per second for 4096 samples. Spectrum is divided into several bands as in conventional graphic equalizer. The number of bands depends on approach selected.

For fast Fourier transform computing with fixed point arithmetic we use *kiss\_fft* library. It is open source library distributed under BSD license. This library does not use platform-specific commands and is easily ported to ARM. On our platform processing time for one buffer is about 0.004 s.

Our first approach of visualizing music by colors was inspired by Luke Nimitz demonstration of "Frequency spectrograph – Primary Harmonic Music Visualizer". It is similar to Scriabin

idea. It can be considered as a specific visualization of the graphic equalizer. In this demonstration music octaves are associated with HSL color wheel as shown in fig. 7 using statement:

$$\text{Angle} = 2\pi \log_2 \left( \frac{f}{c} \right),$$

where  $f$  is frequency,  $c$  is origin on frequency axis. Angle defines hue of current frequency.

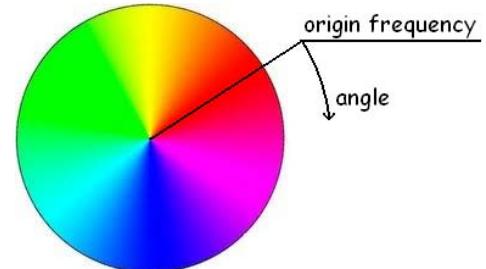


Figure 7. Color circle corresponding to each octave.

Depending on value of current note we define brightness of selected hue and draw it on color circle. We use three different approaches to display color on the color wheel: paint sectors, paint along radius or use different geometric primitives inscribed into the circle.

In *Soap Bubbles* effect, depending on generated color circle, we determine color of bubble texture. On fig. 8 there is an example of soap bubbles with color distribution depending on music. In *Sunlight Spot* effect generated color circle determines distribution of colors on highlighted spot (fig. 9).



Figure 8. Generated color distribution of soap bubbles depending on music.



Figure 9. Generated color distribution of sunlight spot depending on music.

In second approach we detect beats or rhythm of the music. We tried several techniques for beats detection in time and frequency domains [11, 12, 13, 14]. We faced constraints due to real-time performance limitation and we were dissatisfied with the outcomes for some music genres. Finally we assume that the beat is present if there are significant changes of values in several bands. This method meets performance requirements with acceptable quality of beats finding. Fig. 1 illustrates how detected beats affect size of flashing light. If the beat is detected we instantly maximize size and brightness of lights and then they gradually return to their normal state until next beat happens. Also it is possible to change flashing lights when beat happens (turn on and off light sources). In the *Soap Bubbles* effect we maximize saturation of the soap bubble color when the beat takes place. We also change the direction of moving soap bubbles as beat happened. In *Sunlight Spot* effect if the beat is detected we maximize brightness and size of spot and then they gradually returned to their normal state.

In third approach we analyze presence of low, middle and high frequencies in audio signal. This principle is used in color music installations. In *Soap Bubbles* effect we assign frequency range for each soap bubble and define its saturation according value of corresponding frequency range. In *Flashing Light* effect we assign each light star to its own frequency range and define its size and brightness depending on value of the frequency range. On fig. 10 you can see how presence of low, middle and high frequencies affect on flashing lights.



**Figure 10.** Low, middle and high frequencies affect on brightness and saturation of corresponding flashing lights.

Another approach is not to divide spectrum to low, middle and high frequencies but rather to assign it to different tones inside octaves. So, we work with equalizer containing large amount of bands, where each octave have enough corresponding bands. We accumulate values of each equalizer band to buffer cell, where corresponding cell number is calculated using the following statement:

$$num = \frac{(\log_2(\frac{f}{c}) \times 360) \bmod 360}{\frac{360}{length}} + 1,$$

where  $f$  is frequency,  $c$  is origin on frequency axis,  $length$  is number of cells.

Each cell controls behavior of selected objects. In *Soap Bubbles* effect we assign each soap bubble to corresponding cell and define its saturation depending on the value of the cell. In *Flashing Light* effect we assign each light to corresponding cell and define its size and brightness depending on the value of the cell.

Obviously, other approaches to adapt behavior of animation to the background audio are also possible. In particular, it is visible to analyze the left and right audio channels separately and apply the different behavior to the left and right sides of the screen, respectively. Other effects friendlier for music adaption can be created.

## 5. RESULTS AND DISCUSSION

The major issue is how can we implement the functions in modern multimedia devices for real-time animation? The algorithms were optimized for ARM Cortex A8 and A9 – based platforms with CPU frequency 800 - 1000 MHz. Limited computational resources of the target platform combined with absence of graphics hardware acceleration is serious challenge for implementation of visually rich animation effects. Therefore comprehensive optimization is required to obtain smooth framerates. Total performance win is 8.4 times in comparison to initial implementation. The most valuable optimization approaches are listed in table 1.

Table 2 contains performance data for described effects. Such figures provide smooth and visually pleasant animation.

TABLE 1 OPTIMIZATION APPROACHES

Approach	Speeding-up, times
Fixed-point arithmetic	4.5
SIMD CPU instructions (NEON)	3
Effective cache usage	1.5
Re-implementing of key glibc functions	1.25

TABLE 2 PERFORMANCE OF PROPOSED EFFECTS FOR HD PHOTO.

Effect	Initialization time, s	FPS
Flashing Light	0.15	20
Soap Bubbles	0.08	45
Sunlight Spot	1.4	50

As objective evaluation of the proposed audiovisual presentation is difficult, we evaluate the advantage of our technique through subjective user opinion survey. *Flashing Light*, *Soap Bubbles* and *Sunlight Spot* effects with octave based audio adaptation were used for demonstration. Two questions were asked for three audio-visual effects:

- Are you excited by the effect?
- Would you like to see that effect in your multimedia device?

23 observers participated in the survey. Diagram on fig. 11 reflects survey results. In general, absolute majority of the interviewees rates effects positively. Only two people said that they do not like not only demonstrated effects, but any multimedia effects. Some observers stated: it's entertaining, but I cannot say

“I’m excited”, because such expression would be too strong. Several participants of the survey said that they do not like photos or background music used for demonstration. It is also worth to notice that 8 of the respondents were women and, on average, they rated the effects much higher than men.

So we can claim that the outcomes of subjective evaluation demonstrate the satisfaction of the observers with this new type of audiovisual presentation, because audio-aware animation behaves uniquely each time it is played back, and does not repeat itself during playback duration, thus creating vivid and lively impression for the observer. A lot of observers were excited by the effects; and they want to see such features in their devices with multimedia capabilities.

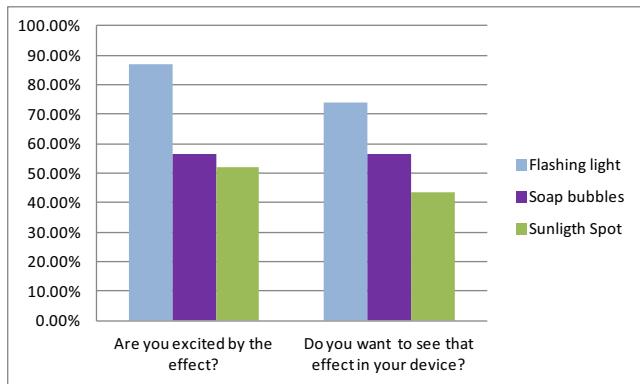


Figure 11. Survey results.

## REFERENCES

- Chen, J., Xiao, J., Gao, Y. *iSlideshow: a Content-Aware Slideshow System*, ACM Intelligent User Interface conf., 2010.
- Sakaino, H. *The photodynamic tool: generation of animation from a single texture image*. IEEE ICME, 2005.
- Safonov, I., Bucha, V. *Animated thumbnail for still image*, GRAPHICON-2010, pp. 79-86, 2010.
- Chen, J.C., Chu, W.T., Kuo, J.H., Weng, C.Y., Wu, J.L. *Tiling slideshow*. ACM Multimedia 2006, 25-35, 2006.
- Dunker, P., Popp, P., Cook, R., *Content-aware auto-soundtracks for personal photo music slideshows*. IEEE ICME 2011, 1-5, 2011.
- Viola, P., Jones, M., *Rapid object detection using a boosted cascade of simple features*, In Proc. of Conference Computer Vision and Pattern Recognition, 2001.
- Egorova, M.A., Murynin, A.B., Safonov, I.V., *An Improvement of face detection algorithm for color photos*, Pattern Recognition and Image Analysis, vol. 19, No. 4, pp. 634–640, 2009.
- Vil’kin, A.M., Safonov, I.V., Egorova, M.A., *Bottom-up Document Segmentation Method Based on Textural Features*, Pattern Recognition and Image Analysis, vol. 21, No. 3, pp. 565–568, 2011.
- Itti, L., Koch, C., Niebur, E., *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.
- Cheng, M.M., Zhang, G.X., Mitra, N.J., Huang, X., Hu, S.M., *Global Contrast based Salient Region Detection*. IEEE CVPR 2011, 409-416, 2011.
- Goto, M., *Real-time music-scene-description system: predominant-F0 estimation for detecting melody and bass lines in real-world audio signals*, Speech Communication, vol. 43, no. 4, pp. 311–329, 2004.
- Dixon, S., *MIREX 2006 Audio Beat Tracking Evaluation: BeatRoot*, in MIREX at 7th International ISMIR 2006 Conference, 2006.
- Scheirer, E.D., *Tempo and beat analysis of acoustic musical signals*, J. Acoust. Soc. Amer., vol. 103, no. 1, p. 588-601, Jan. 1998.
- McKinney, M.F., Moelants, D., Davies, M.E.P., Klapuri, A., *Evaluation of Audio Beat Tracking and Music Tempo Extraction Algorithms*. Journal of New Music Research 36(1), 1-16, 2007.

## About the authors

Konstantin A. Kryzhanovsky received his MS degree in cybernetics from Moscow Engineering Physics Institute/University (MEPhI), Russia in 2000. Since 2004 he works as an instructor of faculty of Cybernetics of MEPhI. Since 2011, K.A. Kryzhanovskiy joined Samsung Moscow Research Center, Russia where he is working on computer graphics, image and video processing projects.

Aleksey M. Vil’kin received his MS degree in mathematics from National Research Nuclear University MEPhI in 2011. From 2011 he is PhD student in MEPhI working on pattern recognition, page segmentation problems. In 2011 Aleksey joined Samsung Moscow Research Center, where he is engaged in on computer graphics, image and video processing projects.

Ilia 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 Samsung Moscow Research Center, where he is engaged in image and video processing projects.

Zoya V. Pushchina graduated from Moscow Bauman State Technical University and received her MS degree in CS in 1996. From 2009 she works in Samsung Moscow Research Center. Her role is image and video algorithm optimization for embedded platforms.

# Virtual Reality Technology for the Visual Perception Study

G. Menshikova, Yu. Bayakovski, E. Luniakova, M. Pestun, D. Zakharkin

Department of Computational Mathematics and Cybernetics and Department of Psychology

Moscow State University, Moscow, Russia

[gmenshikova@gmail.com](mailto:gmenshikova@gmail.com), [ymb-lab@yandex.ru](mailto:ymb-lab@yandex.ru), [eglun@mail.ru](mailto:eglun@mail.ru),

[max.pestun@gmail.com](mailto:max.pestun@gmail.com), [denis.zakharkin@gmail.com](mailto:denis.zakharkin@gmail.com)

## Abstract

Visual illusions have provided researchers with important insights into the rules of how the visual system interprets environmental information. In current models of lightness perception it has been suggested that 2D visual cues in a scene play a crucial role in lightness estimations. The role of depth cues was investigated in some studies, but the results were contradictory. Lately, the virtual reality (VR) techniques were applied successfully to investigate 3D visual perception. Using the CAVE system, we studied the strength of 3D visual illusions. We investigated the role of 3D articulated backgrounds in the perception of the simultaneous lightness contrast (SLC) illusion. The results showed that the illusion strength decreased for all 3D displays relative to the 2D articulated version. There were no significant differences between different types of 3D displays.

**Keywords and Phrases:** 3D visual illusions, lightness perception, simultaneous lightness contrast Virtual Reality, VR, CAVE.

## 1. INTRODUCTION

The problem of lightness perception is tightly connected with the perception of lightness illusions. They were often used as demonstrations of theoretical assumptions by different approaches to lightness perception. Recently the anchoring theory of lightness perception was frequently debated [4]. It assumed that the ratios of the test surface luminance to the luminance of other surfaces determined this process. Using these ratios it was possible to estimate the relative reflectance of all surfaces, which were equally illuminated. Since objects lying in the same surfaces were illuminated uniformly in natural scenes, it was assumed, that it was the luminance ratio of coplanar surfaces that was a basic stimulus for lightness perception. It allowed the relative reflectance of all surfaces to be computed. To estimate the absolute reflectance the anchoring rule was applied: one of relative reflectance values anchored to some absolute value, for example, to the most luminous object which supposed to be white [8; 4]. In complex scenes the lightness estimation was accomplished into two stages. At first the groups of coplanar surfaces were picked out, and then the lightness was estimated in accordance with the anchoring rule for each group. The anchoring rules “worked” in the range of local and global frameworks. Local frameworks were used to estimate the luminance ratio of the test and adjacent background patches of a surface. Global frameworks were used for estimating the luminance ratio of test and distant patches of a surface. These theoretical hypotheses were used to explain some of lightness illusions, for example the simultaneous lightness contrast (SLC) illusion [4; 3]. It was found that the SLC illusion formed mainly on the anchoring rules of local frameworks. In our study we

investigated the influences of 1) depth cues and 2) articulation cues on lightness perception to test the anchoring theory.

The role of depth cue was tested in a number of works, but the results were contradictory. The main idea of these studies consisted in manipulating of 3D positions of test surfaces relative to background surfaces. In accordance with the coplanar ratio hypothesis it would result in a shift of lightness estimations. Some works [9; 1] confirmed these predictions. These results showed that the relationship between test and background surfaces occurred only when they were coplanar to each other. Other studies did not reveal or determined the very weak influence of coplanar ratios on lightness estimations [2; 11]. So, the question of depth cue influence remained unclear. In our study the strength of the SLC illusion was investigated as a function of 3D configurations of test and background squares. In line with the coplanar ratio hypothesis lightness estimations were determined by the anchoring rules which had the relative strengths within the local and global frameworks. When the test squares were moved out of the background squares it would result in weakening local anchoring and, in their turn, in reducing the illusion strength.

The influence of the articulation cues on lightness perception was proposed and investigated by D. Katz [6]. Articulation effects were determined as the influence of the background complexity on lightness estimations. The term “complexity” referred to the number of colored patches of the background. The rule of articulation was formulated as following: the more colored patches were located around the test patch the better lightness estimations. This rule has been co-determined in the anchoring theory to accommodate the modern approaches and studies on lightness perception [5]. We proposed that “complexity” may be considered not only as the number of colored patches of the background, but also as 3D content of the background. Then the rule of articulation may be formulated as following: the more complex 3D scene would surround the test patch the better lightness estimations and, in turn, the high the illusion strength.

The last decades have seen a rise in usage of a new Virtual Reality (VR) technology in psychological research. By now its effectiveness has been proven by medicine, neuropsychology, cognitive and social psychology data. The virtual reality technology equips experimental psychology with methods that have certain differences from traditional laboratory instruments. A heated dispute of the advantages and disadvantages of the use of virtual reality systems in psychology has been held in all experimental and review works carried out within this new methodology [7; 10; 12]. As for the studies in lightness perception VR technology provides 1) active 3D viewing enable to construct visual illusions in depth and 2) complex 3D scenes with controlled parameters enable to reproduce articulated effects.

Using the VR technology, we studied the strength of 3D SLC illusion to find out the role of 1) the depth cues and 2) the articulation cues in lightness perception.

Two hypotheses were offered:

1. Locating the test and background surfaces in different space positions would result in weakening local anchoring and, in their turn, in reducing the illusion strength. So, the strength of 3D SLC illusions would decrease relative to its classical 2D configuration.
2. The more complex 3D scenes of the background would result in better lightness estimations and, in turn, in high values of the illusion strength.

## 2. METHOD

### 2.1 Observers

Twenty five observers (age range 17–30) with normal or corrected to normal vision were tested. All observers were unaware of the purpose of the experiment.

### 2.2 Stimuli

The 2D articulated version of the SLC illusion (Fig. 1.1) was used as a basic display. Three different 3D configurations of the SLC illusion were constructed. They consisted of test squares which were moved out of the backgrounds and different types of 3D backgrounds. The first type was 2D articulated patches (Fig. 1.2), the second – 3D cubes (Fig. 1.3) and the third – 3D balls (Fig. 1.4). 3D backgrounds varied from simple (the first type) to complex (the third type) variant of articulation.

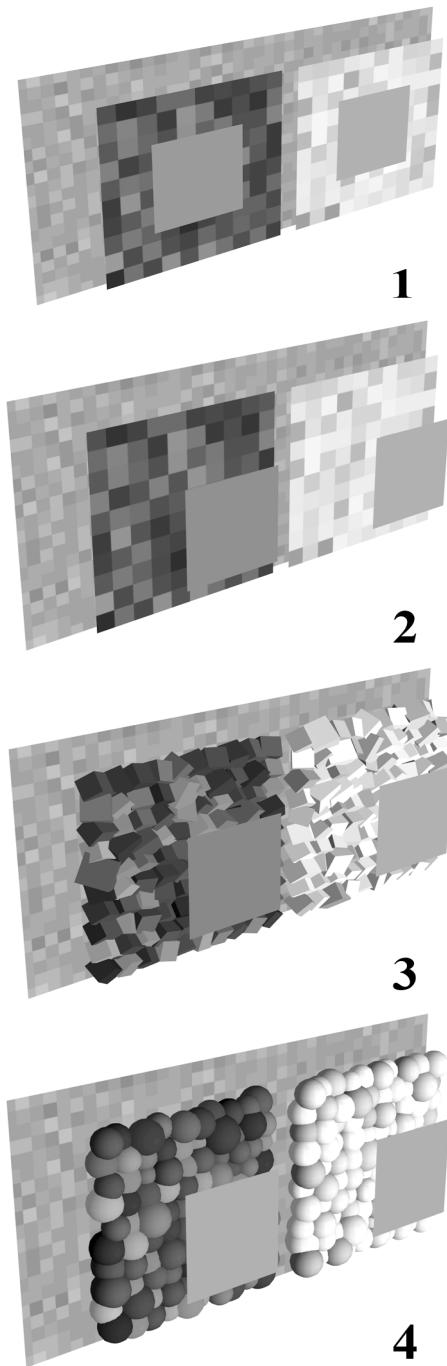
The average luminance of backgrounds was constant for all types of stimuli.

The method of constant stimuli was used to estimate the strength of the SLC illusion. The gray squares on the light backgrounds were standard. Its lightness was 30% of white shade in Grayscale units and was not changed during the experiment. Seven variable stimuli were created for every 2D-3D configuration, for which the value of lightness for the test squares lying on the dark backgrounds decreased from 30% to 12,5% of white shade with a step of 2,5%. So, 28 stimuli were created: four 2D-3D configurations, each having seven variable stimuli.

### 2.3 Apparatus

The 2D articulated version of the SLC illusion and three types of 3D displays were presented using the CAVE system (Fig. 2).

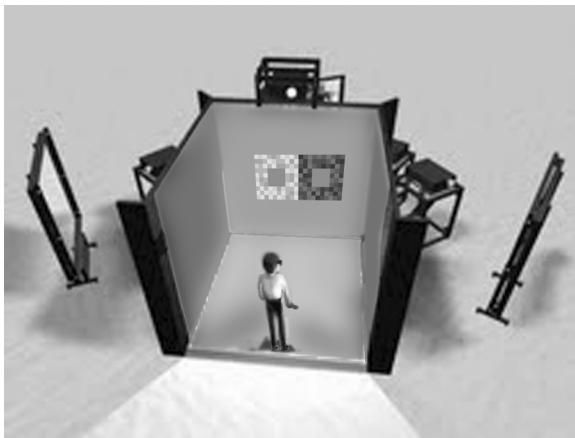
The CAVE system has four large flat screens (Barco ISpace 4), which are connected into one cube consisting of three walls and a floor. The length of each screen side is about 2.5 meters. Shutter eye glasses are made by CrystalEyes 3 Stereographics. Projection system is based on BarcoReality 909. The projector's matrix resolution is 1400x1050 with 100 Hz update frequency. Tracking system produced by ArtTrack2. VirTools 4.0 is used for software developing. It supports DX9/GL2, HAVOK, particle systems and shaders.



**Figure 1:** Different types of 2D-3D configurations of the SLC illusion: 1 – 2D classic articulated configuration; 2 – 3D configuration with 2D articulated background; 3 - 3D configuration with 3D articulated background (cubes); 4 - 3D configuration with 3D articulated background (balls).

The observer stood motionless in front of the central screen at a distance of 2.5 m. Virtual stimulus configuration was located before him with the background placed on the screen plane. It subtended 30° of visual angle horizontally and 15° vertically.

The visual angle of the test squares in 3D configurations (2, 3 and 4) was the same as those in the 2D display (1). Thus, when projected on the retina, 3D and 2D displays produced practically the same pattern.



**Figure 2:** The CAVE system.

The laboratory room was darkened; there were no any light sources, except CAVE systems projectors. The luminance range in stimulus scene was 1:230. The maximum luminance was  $5.5 \text{ cd/m}^2$ , the minimum –  $0.02 \text{ cd/m}^2$ .

#### 2.4 Procedure

The observer was given the following instructions: “Each trial you will see two gray test squares on the different backgrounds. Please, choose the lightest of two central squares, using a special joystick. Try to stand motionless during the experiment.”

The experiment included four series: 1 – 2D articulated version of the SLC illusion; 2 – 3D articulated version of the SLC illusion with backgrounds consisted of 2D patches; 3 – 3D articulated version of the SLC illusion with backgrounds consisted of cubes; 4 – 3D articulated version of the SLC illusion with backgrounds consisted of balls.

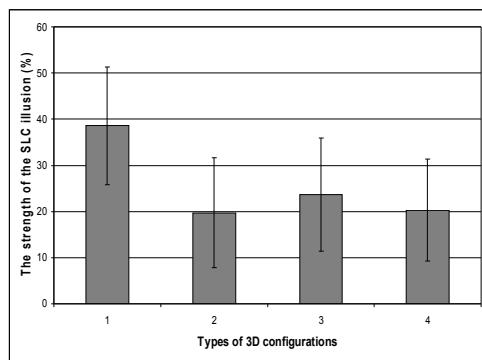
Each series lasted about 5 minutes. The stimuli sequence was completely randomized. Every series consisted of 70 trials: each of seven variable stimuli was repeated 10 times. The left/right position of light and dark backgrounds was changed randomly.

### 3. RESULTS

Psychometric functions for 2D and three different 3D configurations were obtained and used to evaluate the strength of the SLC illusion for each participant and each 2D-3D configuration. The illusion strength was calculated as  $IS = (LSt - LT) / LSt * 100\%$ , where  $LSt$  – was luminance of standard square;  $LT$  – PSE (Point of Subject Equality) – luminance of test square with 50% probability of answers “lighter”.

The results averaged across 25 observers are shown in Figure 3. The horizontal axis plots the different 2D-3D configurations. The vertical axis plots the average strength of the SLC illusion (%).

The significant differences were revealed between the type 1 (2D articulated configuration) and the other different 3D configurations ( $p < .001$ ). The strength of 2D classic display was twice more than the strength of any 3D display of the SLC illusion. As to articulation effects, there were no significant differences between the values of SLC strength calculated for three types of 3D backgrounds ( $p < .01$ ).



**Figure 3:** The strength of the SLC illusion for 4 types of stimulus configurations: 1 – 2D articulated configuration; 2 – 3D configuration with backgrounds consisted of 2D patches; 3 – 3D configuration with backgrounds consisted of cubes; 4 – 3D configuration with backgrounds consisted of balls.

### 4. CONCLUSION

The results showed that the illusion strength decreased for all 3D displays relative to the 2D articulated classic display. This result was in good agreement with the anchoring theory, predicting the reduction of the illusion strength in the conditions of different depth positions of the test and background surfaces. So, our first hypothesis was successfully confirmed.

There were no significant differences in illusion strength for different types of 3D backgrounds. It seems that articulation effects weakly depend on the type of 3D backgrounds. So, our hypothesis of the influence of 3D backgrounds on lightness estimations was not proved.

Virtual reality technologies may be effectively used in studies of lightness perception. It enables to reproduce visual illusions in depth and to construct complex 3D scenes with controlled parameters to create articulated effects.

### 5. ACKNOWLEDGMENTS

The work is supported by the grant “The application of modern information technologies to the development of innovative methods in the study of human cognitive processes” within the framework of the Federal Target Program “Scientific and scientific-pedagogical personnel of innovative Russia” for 2009-2013.

### 6. REFERENCES

- [1] Coren S. Brightness contrast as a function of figure-ground relations. *Journal of Experimental Psychology*, 80, (1969), 517-524.
- [2] Epstein W. Phenomenal orientation and perceived achromatic color. *Journal of Psychology*, 52, (1961), 51-53.
- [3] Economou E., Zdravkovich S., Gilchrist A. Anchoring versus spatial filtering accounts of simultaneous lightness contrast. *Journal of Vision*, 7,12, (2007), 2-15.
- [4] Gilchrist A.L., Kossyfidis C., Bonato F., Agostini T., Cataliotti J., Li X., Spehar B., Annan V. An anchoring theory of lightness perception. *Psychological Review*, 106, 4. (1999), 795-834.

- [5] Gilchrist, A., Annan V. *Articulation effects in lightness: Historical background and theoretical implications.* *Perception*, 31, (2002), 141-150.
- [6] Katz D. *The world of color.* (Ed. by Kegan Paul, Trench, Trubner & Co: London), 1935.
- [7] Khan Y., Xu Z., Stigant M.: *Virtual Reality for Neuropsychological Diagnosis and Rehabilitation: A Survey.* *Proceedings of the Seventh International Conference on Information Visualization* (2003), 158-163. IEEE Computer Society, Washington DC, USA.
- [8] Land E.H., McCann J.J. *Lightness and retinex theory.* *Journal of the Optical Society of America*, 61, (1971), 1-11.
- [9] Wolff W. *Ueber die kontrasterregende Wirkung der transformierten Farben.* *Psychologische Forschung*, 18, (1933), 90 – 97.
- [10] Yee N., *Psychological Research in Virtual Worlds*, <http://bps-researchdigest.blogspot.com/2007/06/psychological-research-in-virtual.html>
- [11] Zaidi Q., Spehar B., Shy M. *Induced effects of backgrounds and foregrounds in three-dimensional configurations: the role of T-junctions.* *Perception*, 26, (1997), 395-408.
- [12] Zinchenko Yu.P., Menshikova G.Ya., Bayakovskiy Yu.M., Chernorizov A.M., Voiskounsky A.E. *Technologies of virtual reality in the context of World-wide and Russian psychology: methodology, comparison with traditional methods, achievements and perspectives // Psychology in Russia. State of the Art. Scientific Yearbook / Ed. by Yu.P. Zinchenko & V.F. Petrenko.* – Moscow: Lomonosov Moscow State University; Russian Psychological Society, (2010). 11-45.

## About the authors

G. Menshikova is a PhD in Psychology, Head of the Perception Laboratory at Moscow State University, Department of Psychology. Her contact email is [gmenshikova@gmail.com](mailto:gmenshikova@gmail.com).

Yu. Bayakovskiy is a PhD in Computer Science, Associate Professor and Head of the Computer Graphics and Multimedia Laboratory at Moscow State University, Department of Computational Mathematics and Cybernetics. His contact email is [ymb-lab@yandex.ru](mailto:ymb-lab@yandex.ru).

E. Luniakova is a PhD in Psychology, Senior Researcher in the Perception Laboratory at Moscow State University, Department of Psychology. Her contact email is [eglun@mail.ru](mailto:eglun@mail.ru).

M. Pestun is a student at Moscow State University, Department of Computational Mathematics and Cybernetics and Engineer in the Perception Laboratory, Department of Psychology. His contact email is [max.pestun@gmail.com](mailto:max.pestun@gmail.com).

D. Zakharkin is an Engineer in the Perception Laboratory, Department of Psychology. His contact email is [denis.zakharkin@gmail.com](mailto:denis.zakharkin@gmail.com).

# Adaptive 3D Color Anaglyph Generation for Printing

Elena Patana, Ilya Safonov, Michael Rychagov

Samsung Moscow Research Center, Russia

{e.patana, ilia.safonov, michael.rychagov} at samsung dot com

## Abstract

Nowadays stereophotography is rapidly developing, providing a plenty of sources for stereoimages. The goal of current technology – to provide users with possibility to get high quality 3D anaglyph prints for education and entertainment. To do so it is necessary to agree color characteristics of glasses and printed colors, since errors in color transmission lead to cross-talk interference and ghosting effects. There is no easy way for user to adjust colors of anaglyph in order to coordinate characteristics of glasses and printer.

We propose a technique that allows generating anaglyphs with colors adapted to given glasses and printer colors by means of special color pattern analysis. In addition, our approach takes into account the size of the printed anaglyph image. Resulting printed images have a good quality that is confirmed by user opinion survey. The images contain fewer artifacts and look better in comparison to anaglyphs without adaptation, which are generated in existing software applications. The technique utilizes a low amount of memory and has low computational complexity.

**Keywords:** Anaglyph, stereo printing, crosstalk noise, ghosting reduction.

## 1. INTRODUCTION

At present time, there are a lot of sources of stereo images: 3D cameras, 3D movies, stereo-pairs can be created from two frames captured by conventional 2D camera, several software technologies allow to catch different views of the same 3D object,

for example, Google Earth and parallax effect in HTML 5. 3D books have been known for several decades, they are viewed through glasses with different color filters, for example blue and red. Reconstruction of three-dimensional views by anaglyphs is one of the simplest and the most economic methods. However this method has some disadvantages such as loss of color and discomfort for prolonged viewing. In spite of the drawbacks, sometimes consumers want to print 3D color anaglyph pictures for education and entertainment. There are various user groups in Web like Flickr and others, where the process of 3D color anaglyphs generation is discussed [4, 9].

There are several PC and mobile software applications for generation of anaglyphs, for example StereoPhoto Maker, Anaglyph Maker, Anaglyph Workshop, Z-Anaglyph. They have some disadvantages: settings for ghosting effect reduction do not take into account size of the printed anaglyph; there is no adaptation for given glasses and printer colors.

Usually it is assumed that viewing anaglyph on a display and adjusting of printer color profile is enough to get similarly looking printed anaglyph. However, our experiments show that it is impossible to get similar color perception while looking at a display and color hardcopy produced by a laser or ink-jet printer. From a theoretical point of view it can be easily explained, because gamut of a display and gamut of printing devices are rather different [2]. That is why printing devices manufacturers are interested in rising of printing quality of anaglyph [10].

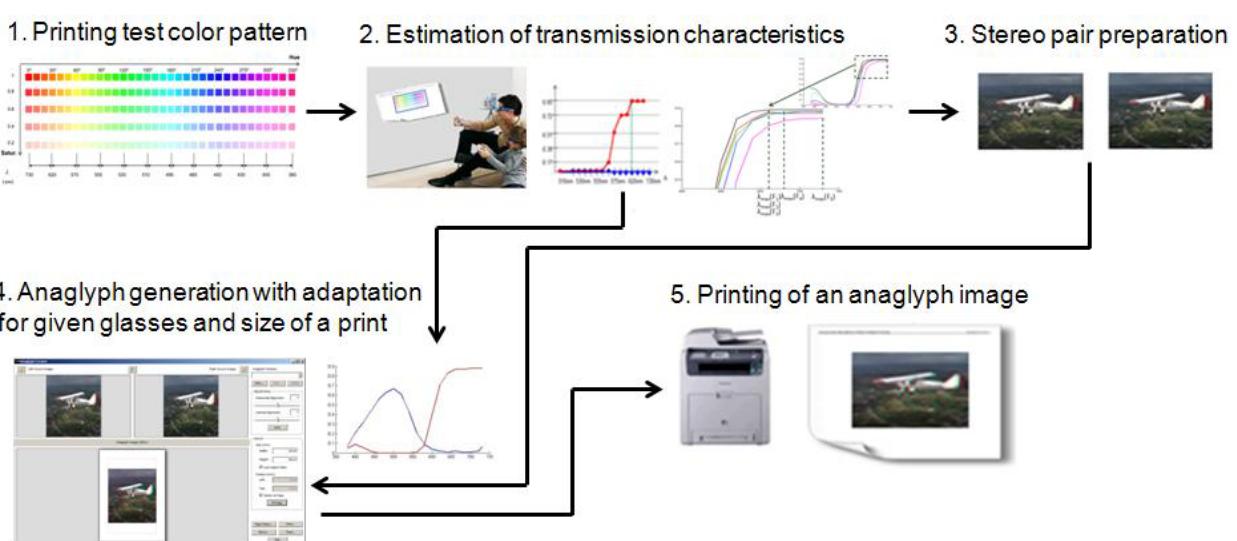


Figure 1. Algorithm of an adaptive anaglyph generation

After performing the measurements of transmission coefficient for several glasses ( $\tau_r(\lambda)$  for right filter and  $\tau_l(\lambda)$  for left filter) as well as reflection spectrum of printed colors, i.e. magenta  $g_M(\lambda)$  and cyan  $g_C(\lambda)$ , by spectrometer, and analyzing estimated reflection and transmission coefficients, it was concluded that full elimination of cross-talk effect is impossible. Nevertheless, it is possible to reduce cross-talk interference by correct color setting of a printed anaglyph. As it is evident by experiment, the primary printed colors (cyan and magenta) aren't transmitted well; but our estimations show that colors selection according to given glasses allows to decrease ghosting artifacts significantly.

In this paper we propose an adaptive to spectral characteristics of glasses and printer's colors algorithm for anaglyph creation.

## 2. RELATED WORKS

Anaglyph generation is a difficult problem. It isn't enough to put one color channel in a left image and another in a right image. Contradictory problems should be solved in anaglyph generation process, because it is needed to code two images on a single view by colors for stereo effect making and to reproduce colors with maximum naturalness. Due to stereo and color conflicts it appears to be impossible to develop an algorithm for anaglyph creation which always produces good color representation, good details, and is permanently free from typical artifacts, such as ghosting, and region merging.

Well-grounded technique for color anaglyph generation for display is described in [1,3,5,8].

Several techniques have been proposed for the production of anaglyphs for viewing on displays. In [8] three approaches are discussed: Photoshop algorithm (PS) and its variants, the least squares algorithm (LS) proposed by Eric Dubois, that optimizes colors in the CIE space, and the midpoint algorithm (MID) that minimizes the sum of the distances between the anaglyph color and the left and right eye colors in CIE L\*a\*b\*. Linear anaglyph algorithms will always map several different left/right eye colors to the same RGB color. The results show that the MID method produces excellent color and detail for color images but may suffer severe ghosting. Anaglyphs produced by the LS method are normally darker with less detail and require brightening or gamma correction but appear to have no ghosting. The PS method is easy to implement and works well for grayscale images but may also suffer from ghosting and poor color representation.

In [3] several methods for anaglyph enhancement are proposed that rely on stereo image registration, defocusing and nonlinear operations on synthesized depth maps. These enhancements substantially reduce unwanted ghosting artifacts, improve the visual quality of the images, and make comfortable viewing of the same sequence possible in three-dimensional as well as the two-dimensional mode.

The method for computing pixel colors in anaglyph images presented in [5] depends upon knowing the RGB spectral distributions of the display device and the transmission functions of the filters in the viewing glasses. It requires the solving of a nonlinear least-squares problem for each pixel in a stereo pair and

is based on minimizing color distances in the CIEL\*a\*b\* uniform color space.

The method proposed in [1] is adapted to the spectral absorption curves of the left and right filters of the anaglyph glasses. A projection technique is used to compute the anaglyph image that yields an image pair (after the glasses) as close as possible to the desired stereo pair. In order to generate anaglyph it is necessary to move into the XYZ space by transition matrix:

$$[C]_{kj} = c_{kj} = \int \bar{p}_k(\lambda) d_j(\lambda) d\lambda,$$

where  $\bar{p}(\lambda)$  - color-matching function,  $d(\lambda)$  - spectrum of standard illuminant. Reflection light from an image passes through color filters of an anaglyph glasses and is transformed by two transition matrixes:

$$[A_l]_{kj} = a_{lkj} = \int \bar{p}_k(\lambda) d_j(\lambda) f_l(\lambda) d\lambda \quad (\text{for left eye filter}) \quad \text{and} \\ [A_r]_{kj} = a_{rkj} = \int \bar{p}_k(\lambda) d_j(\lambda) f_r(\lambda) d\lambda \quad (\text{for right eye filter}).$$

An anaglyph is generated by the following formula:

$$\hat{V}_{an}(x) = N(R^T M R)^{-1} R^T M C_2 V(x),$$

where  $N$  is normalizing matrixes for condition  $\hat{V}_{aj} \in [0,1]$ ,  $M$  is weighted matrix, which allows weighting of the Y component more heavily than X and Z to favor reproduction of the correct luminance,  $R = \begin{bmatrix} A_l \\ A_r \end{bmatrix}$ ,  $C_2 = \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix}$ .

## 3. ADAPTIVE ANAGLYPH GENERATION

### 3.1 General workflow

In our paper we generate anaglyphs with a help of the method described in [1]. The algorithm takes into account spectral characteristics of stereo glasses. We propose a new approach for adaptation of anaglyph colors on hardcopy to stereo glasses to reduce ghosting effect to minimum.

Before anaglyph generation a process of transmission function estimation is performed. For this purpose firstly we print the color pattern (figure 1, step 1) on a target printer, then we estimate transmission coefficients of given glasses. Detailed description of test color pattern is present in section 3.3; and a sequence of actions for evaluation of transmission function of glasses filters (figure 1, step 2) by user is described in section 3.4.

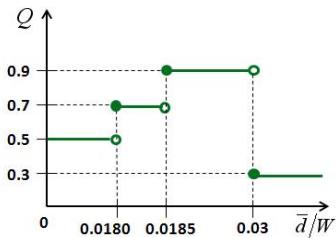
Then it is needed to prepare a stereo pair (figure 1, step 3). Preparation includes geometrical aligning, color correction and enhancement. Stereo pair color correction is performed by well-known method of histogram matching [7]. After that the stereo pair is enhanced to reduce unwanted artifacts. There are two main ideas for anaglyph enhancement: decreasing of disparity range of a stereo pair and color component defocusing. Detailed description of these ideas is present in section 3.2. Then anaglyph generation is performed by method [1] with adapted transmission functions (figure 1, step 4). Also we propose a method of disparity correction for keeping 3D effect and reduction of ghosting effect appearance on an anaglyph hardcopy, because anaglyph size can be different on display and hardcopy (figure 1, step 5). It is presented in section 3.5.

### 3.2 Anaglyph enhancement

Decreasing of disparity range of a stereo pair includes estimation of average disparity value on a stereo pair by the following method:

$$\arg_{d_i} \min \sum_{(x,y) \in W} |I(x,y) - I(x+d_i, y)|,$$

where  $d_i$  - disparity with minimal value of SAD (Sum of Absolute Differences) [6] in some region on left and right images. We compute average disparity  $\bar{d}$  after estimation of the disparity map. Decreasing of disparity is produced by horizontal shifting of stereo pair proportionally to  $\bar{d}$ . If average disparity is less than 4 pixels, stereo pair is not changed. The main idea is that disparity should not be too large relatively to image size. Therefore, horizontal shift is computed as  $Shift = Q \cdot \bar{d}$ , where  $Q$  is regularization coefficient which depends on ratio of average disparity and image width  $W$  in pixels, as presented on figure 2. If the value of the ratio  $\bar{d}/W$  is greater than 0.03 it might inform about erroneous average disparity estimation; and shifting of a stereo pair should be made with a smaller value. All constants of the empirical approach were found during a plenty of visual experiments.



**Figure 2.** Dependence of the regularization coefficient from size and disparity of a stereo pair.



**Figure 3.** The anaglyph without disparity correction.



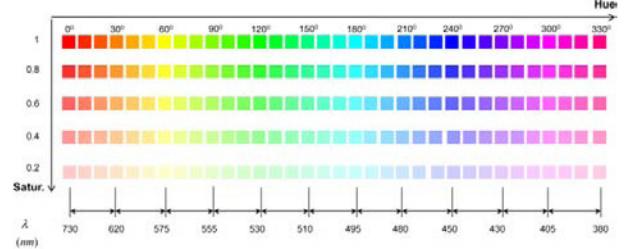
**Figure 4.** The anaglyph with corrected disparity.

Figure 3 shows the anaglyph without disparity correction. On its harcopy ghosting effect is present. Figure 4 demonstrates the anaglyph with decreased disparity. For its generation stereo pair was shifted by 5 pixels. Ghosting effect for this anaglyph harcopy is almost invisible.

Simple and effective way for decreasing of crosstalk noise is defocusing one color channel for both images of stereo-pair [1]. We carry out red or blue channel or both the channels smoothing by means of low-pass box-filter.

### 3.3 Transmission functions estimation by user

We propose to estimate transmission functions  $f_l(\lambda)$  and  $f_r(\lambda)$  for given glasses and printer colors. It allows to significantly reduce ghosting effect and improve quality of printed 3D color anaglyph image. For evaluation of  $f_l(\lambda)$  and  $f_r(\lambda)$  we print the color pattern on target printer. This pattern is a color table including all printable colors in HSL space. This pattern reveals a dependence of digital color components and reflection spectrum of these colors (another dependence of three color components (RGB) and wave length ( $\lambda$ )). Hue corresponds to wavelength, lightness is an average between maximum and minimum values of spectrum; saturation is distance from maximum (or minimum) values of spectrum to lightness. Rows of the table are colors with various saturation (step is 0.2), columns are colors with various hue (step is  $10^0$ ).



**Figure 5.** Test color pattern.

For estimation of transmission function user should examine the pattern through left and right filters of anaglyph glasses separately. By visibility level of color sample through the filter, the transmission function is evaluated and afterwards is applied for anaglyph generation. If the color sample is invisible, it fully passes through the filter and maximum of transmission function is at that location.

Let's left filter is red and right filter is cyan. Firstly user should examine first row with maximal saturation through red filter. We suppose that a color with hue  $= 10^0 \pm 10^0$  corresponds to red color of  $700 \text{ nm} \pm 27.50 \text{ nm}$  wave length range. If any color from the band  $10^0 \pm 10^0$  is invisible, the maximum of transmission function is in the  $700 \text{ nm} \pm 27.50 \text{ nm}$  range. If the color is visible, user should choose the row with less saturation (one of the lower rows) and look at corresponding color sample once more. If the color yet is visible, user should choose the row with the least saturation for analysis. Value of maximum of transmission function depends on saturation that was chosen:

Row of color pattern	Saturation	Maximum of transmission (MaxTrans)
1	1	0.90 (90%)
2	0.8	0.85 (85%)
3	0.6	0.65 (65%)
4	0.4	0.50 (50%)
5	0.2	0.45 (45%)

For cyan filter construction of the transmission function is similar. We use cyan filter of glasses and suppose cyan color has hue =  $180^\circ \pm 10^\circ$  that corresponds to  $495 \text{ nm} \pm 3.75 \text{ nm}$  range. User should examine the test pattern and define visibility level of each color by grade (0...5) which corresponds to transmission. For each clearly visible point we give a 0 grade. Values of transmission function depend on maximum of transmission associated with row of color pattern and grade evaluated by observer.

Transmission	Grade	Transmission	Grade
<i>MaxTrans</i> $\times 1$	5	<i>MaxTrans</i> $\times 0.45$	2
<i>MaxTrans</i> $\times 0.85$	4	<i>MaxTrans</i> $\times 0.20$	1
<i>MaxTrans</i> $\times 0.60$	3	<i>MaxTrans</i> $\times 0$	0

User should mark only invisible or semitransparent colors. In this way we obtain rough approximation of transmission functions for particular glasses.

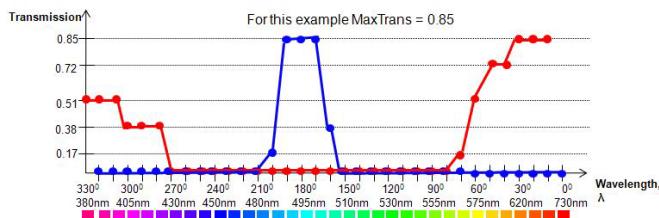


Figure 6. Transmission function evaluated by user.

### 3.4 Algorithm for selection of transmission function of existing filter

Usage of the transmission functions in such “pure” form lead to anaglyph image with wrong colors. We propose to use known transmission functions of Roscolux filters (<http://www.rosco.com/us/filters/roscolux.cfm>) and calculate functions of given glasses by selecting the nearest function to our rough approximation. Roscolux filters are used in professional photography.

In order to choose appropriate filter or glasses with transmission function ( $F_i$ ) most similar to our computed transmission function, we note position of maximum of function and its value.

Then, we choose  $F_i$  as:  $\|\lambda_{\max}(F_i) - \lambda_{\max}(f)\|_i \rightarrow \min$  - estimation position of the maximum. Then for the maximums with equal wavelength we apply the following condition:

$$\|\max(F_i) - \max(f)\|_i \rightarrow \min. \quad (1)$$

For example, compare  $f$  (users' evaluation for red filter) on figure

7 and  $F_i$  (real transmission functions of Roscolux filters) on figure 8. On the top of the figure the transmission functions are presented; in the bottom the enlarged part of the graph with transmission function maximums area is shown. By analyzing wave lengths which correspond to maximums of transmission functions and according to decision rule (1) we select one of the most appropriate filters.

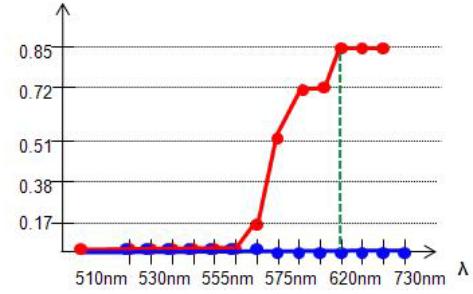


Figure 7. A part of graph of  $f$ .

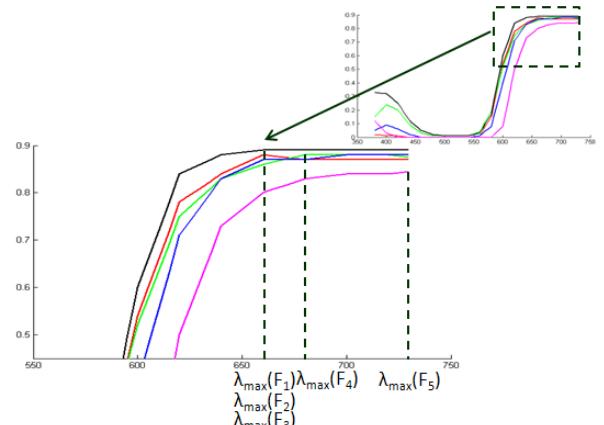


Figure 8. Choosing transmission function of Roscolux filters.

An anaglyph on figure 9 is generated with transmission functions of most common stereo glasses. Its hardcopy printed on Samsung CLP-6240 printer has perceptible ghosting effect for given glasses. An anaglyph on figure 10 is generated with transmission functions adapted for given glasses and printer colors. Ghosting effect on this anaglyph hardcopy is significantly reduced.

### 3.5 Adaptation to size of hardcopy

Anaglyph upsizing while printing leads to large disparity increase and emerging of ghosting effect. Downsizing leads to large disparity decrease and disappearance of stereo effect. We propose to decrease disparity value by shifting stereo pair images relative to each other before enlarging process.

In case of upsizing  $\Delta x = -0.3 * \bar{d}$ , where  $\bar{d}$  is average disparity value computed for whole images of stereo pair,  $\Delta x$  is value of shift of stereo pair images relative to each other for disparity decrease. The value of constant 0.3 is chosen as a result of large number of experiments.

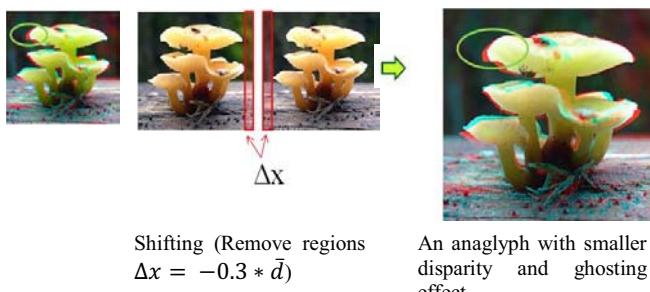
Vice versa in case of downsizing  $\Delta x = 0.3 * \text{AvrgDisp}$  is a shift for disparity increase.



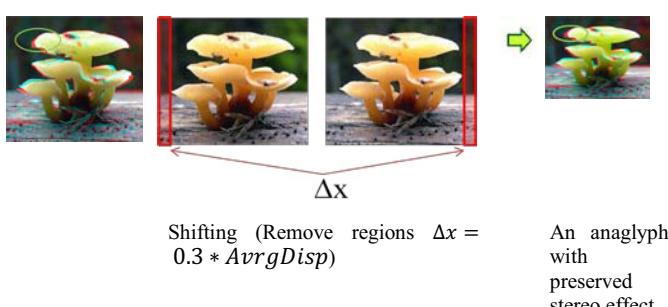
**Figure 9.** An anaglyph example for transmission function of most common stereo glasses.



**Figure 10.** An anaglyph example with adaptation for given glasses.



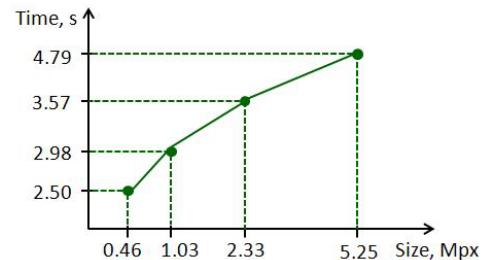
**Figure 11.** Disparity modification in case of upsizing.



**Figure 12.** Disparity modification in case of downsizing.

#### 4. RESULTS AND DISCUSSIONS

All mentioned algorithms were implemented in software application. User interface of the application is shown on figure 1. We applied Windows Presentation Foundation API for nice user interface generation. Mathematical parts of algorithms were developed on C programming language. Proposed technique utilizes low amount of memory and has relatively low computational complexity. Figure 13 demonstrates plot of processing time for anaglyph generation procedure depending on size of stereo-pair images. The computational time data were obtained on PC with dual-core 3 GHz CPU. It is already quite acceptable. Some additional platform-dependent optimizations are possible too.



**Figure 13.** Processing time for anaglyph generation procedure.

In order to perform benchmarking we used 5 mentioned above solutions for anaglyph generation and our software application. Evaluation was done by 14 observers with the same viewing conditions and glasses. We screened the subjects for normal color vision by Ishihara test plates. In our survey we used test set of 6 stereo pairs. Generated anaglyphs were printed on Samsung CLP-6240 printer. We propose to calculate quality factor of anaglyph generation as weighted sum of two subjective estimations:

$$E = \alpha_1 V_1 + \alpha_2 V_2,$$

where  $V_1$  is subjective quality of 3D visualization,  $V_2$  is subjective level of color naturalness,  $\alpha_1 = 0.8$ ,  $\alpha_2 = 0.2$ .  $V_1$  and  $V_2$  are changed from 0 to 1 with step of 0.2; lower  $V_1$  and  $V_2$  are better. We prioritized weights  $\alpha_1$  and  $\alpha_2$  by using Analytic Hierarchy Process [7]. Table 1 contains comparison of the solutions for anaglyph generation obtained during our survey.

**Table 1.** Comparison of applications for anaglyph generation.

	Average E	Maximum E
<b>StereoPhoto Maker</b>	0.33	0.52
<b>Anaglyph Maker</b>	0.57	0.72
<b>Anaglyph</b>	0.45	0.56
<b>Anaglyph Workshop</b>	0.58	0.76
<b>Z-Anaglyph</b>	0.55	0.74
<b>Proposed technique</b>	0.30	0.58

In general our algorithm outperforms all tested solutions. The anaglyph generated by StereoPhoto Maker is presented on figure 14. The anaglyph on hardcopy has well noticeable cross-talk noise. The anaglyph generated by proposed method is presented on figure 15. The picture looks better due to enhancements.



Figure 14. The anaglyph generated by StereoPhoto Maker.



Figure 15. The anaglyph generated by proposed method.

## ACKNOWLEDGMENT

Authors would like to thank Konstantin Kryzhanovsky for help with algorithm implementation and discussions about the paper. Thanks also go to Graphicon's anonymous reviewers for their comments.

## REFERENCE

- [1] Dubois Eric "A projection method to generate anaglyph stereo images", *Proc. of ICASSP, IEEE International Conference "Acoustic, Speech, and Signal Processing"*, vol.3, pp. 1661-1664, 2001.
- [2] Foley James D., van Dam Andries, Feiner Steven K., Hughes John F. "Computer Graphics. Principles and Practice", 2<sup>nd</sup> ed., 1990.
- [3] Ideses Ianir, Yaroslavsky Leonid "Three methods that improve the visual quality of colour anaglyphs", *J. Opt. A: Pure Appl. Opt.* 7, 755-762, 2005.
- [4] Lo Wan-Yen, van Baar Jeroen, Knaus Claude, Zwicker Matthias, Gross Markus "Stereoscopic 3D Copy & Paste", *Proc. of ACM SIGGRAPH Asia*, Vol. 29, Issue 6, 2010.
- [5] McAllister David F., Zhoub Ya, Sullivanc Sophia "Methods for computing color anaglyphs", *Proc. of Stereoscopic Displays and Applications XXI, SPIE Vol.7524*.
- [6] Rziza Mohammed, Aboutajdine Driss "Dense Disparity Map Estimation Using CUMULANTS", 2011 18th IEEE International Conference on Image Processing, p. 984.

[7] Saaty T.L., *Decision Making for Leaders: The Analytic Hierarchy Process for Decisions in a Complex World*, New Edition, Analytic Hierarchy Process Series, Vol. 2, 2001.

[8] Sanders William, McAllister David F. "Producing Anaglyphs from Synthetic Images", *Proc. of Electronic Imaging*, pp. 348 – 358, 2003.

[9] Zhang Yun, Xie Pingping, and Li Hui "Data fusion for multi-scale colour 3D satellite image generation and global 3D visualization", *Proc. of ISPRS Commission VII Midterm Symposium "Remote Sensing: From Pixels to Processes*, 2006.

[10] Zeng Ruzhu, Zeng Huanzhao "Printing Anaglyph Maps Optimized for Display", *Proc. of Color Imaging XVI: Displaying, Processing, Hardcopy, and Applications*, vol. 7866, 2011.

## About the authors

**Elena I. Patana** received her MS degree in applied mathematics and informatics from Taganrog State University of Radioengineering (TSURE), Russia in 2006 and her PhD degree in computer science from Southern Federal University in 2009. Since 2010, Dr. E. Patana has joint Samsung Moscow Research Center, Russia where she is engaged in image and video processing projects.

**Ilia 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 2004, Dr. I. Safonov is with Samsung Research Center in Moscow, where he is engaged in image and video processing projects.

**Michael N. Rychagov** received his MS degree in physics from Department of Physics at Lomonosov Moscow State University, Russia in 1986, PhD degree and Dr. Sc. degree from the same university in 1989 and 2000 correspondingly. Since 1989, he has been involved in R&D and teaching on Faculty of Electronic and Computer Engineering at Moscow Institute of Electronic Technology (Technical University). Since 2004, he joined Samsung Research Center, Moscow, Russia, where he is currently Director of Algorithm Laboratory.

# GPU Ray Tracing – Comparative Study of Ray-Triangle Intersection Algorithms

Vladimir Shumskiy  
 Moscow Institute of Physics and Technology  
 Air Graphics  
 v.a.shumskiy{@}gmail.com

Alexandre Parshin  
 Air Graphics  
 nomad{@}renderbro.com

## Abstract

We present a comparative study of GPU ray tracing implemented for two different types of ray-triangle intersection algorithms used with BVH (Bounding Volume Hierarchy) spatial data structure evaluated for performance on three static scenes. We study how number of triangles placed in a BVH leaf node affects rendering performance. We propose GPU-optimized SIMD ray-triangle intersection method evaluated on GPU for path-tracing and compare it's performance with plain Moller-Trumbore and Unit Triangle intersection methods.

**Keywords:** *ray-triangle intersection, GPU programming, Direct3D, DirectCompute, performance study, ray tracing, bounding volume hierarches.*

## 1. INTRODUCTION

While modern graphics cards (GPUs) allow for general computation in a parallel manner, one of the most prominent applications for a GPU is image synthesis. This is thanks to the inherent parallel nature of ray tracing and other global illumination algorithms – the decomposition of images into pixels provides a natural way of creating individual tasks for many parallel processors. Unlike the GPUs a few years ago, modern ones allow us full programmability similar to general CPUs, while the streaming computation model has its own specific issues. This has to be taken into account when adopting the data structures, traversal algorithms and intersection test routines for ray tracing on GPU architecture.

Testing framework for this paper is based on formerly published papers that implement ray-tracing with spatial data structures in GPU. We use bounding volume hierarchies as described in [Gün07] and few different ray-triangle intersection methods, especially Moller-Trumbore [Möl97] and Unit Triangle [Woo05] routine. While performance of each of those algorithms was successfully studied separately, little attention was paid to how triangle-intersection method can affect spatial data structure traversal performance and vice versa. Furthermore, the performance has not been carefully compared on a current programmable GPU architecture, especially using a cross-vendor APIs like OpenCL, DirectCompute or C++ AMP. In this paper we first present such a comparison study dealing with efficiency of two different types of ray-triangle intersection algorithms for ray tracing on GPU.

This paper is further structured as follows. Section 2 summarizes the previous work of ray-triangle intersection on both CPU and GPU and performance comparison on those algorithms. Section 3 describes our choices for implementation. Section 4 shows the result from measurements on two GPUs for a set of scenes. Further it discusses the bottlenecks of a contemporary GPU architecture for ray tracing algorithms. Section 5 concludes the paper with possible perspectives for future work.



Figure 1: Testing scenes: Stanford Buddha, Bunny and Dragon.

## 2. PREVIOUS WORK

Due to important role in computer graphics plenty of research has been done in the field of intersection testing algorithms. Algorithms proposed by Snyder and Barr [Sny87], Badouel [Bad90], Moller-Trumbore [Möl97], Woop [Woo05], Wald and Shevtsov et al. [She07], have been successfully compared and studied [Seg01], [Are88], [Bad90], [Möl97]. In our research we divide algorithms on those which use precomputed data and on those which do not. Based on previous work we decided to use Moller-Trubmore algorithm as a minimal storage, fast non-precomputed type and Swen Woop's Unit Triangle Test as the precomputed one as it requires only 48 bytes per triangle and doesn't need to store vertex list. In this section we describe chosen algorithms along with BVH spatial data structure.

We omit ray-packet algorithms in our work, because the coherence of the rays within the packet is very important since the vector instructions are fully used only if all rays go through the same branch of computation. In situations like physical simulation, collision detection or ray-tracing in scenes, where rays bounces into multiple directions (spherical or bumpmapped surfaces), coherent ray packets break down very quickly to single rays or do not exist at all. Ray-packets have proven [Ail09], [Hav10] to be ineffective in the above mentioned tasks.

### 2.1 Moller-Trumbore's Algorithm

The algorithm proposed by Moller and Trumbore does not test intersection with the triangle's embedding plane and therefore does not require the plane equation parameters. This is a big advantage mainly in terms of memory consumption especially on the GPU execution performance. The algorithm goes as follows [Möl97]:

1. In a series of transformations the triangle is first translated into the origin and then transformed to a right-aligned unit triangle in the y-z plane, with the ray direction aligned with x. This can be expressed by a linear equation

$$\begin{pmatrix} t \\ u \\ v \end{pmatrix} = \frac{1}{P \cdot E_1} \begin{pmatrix} Q & E_2 \\ P & T \\ Q & D \end{pmatrix}$$

Where  $E_1 = V_1 - V_0$ ,  $E_2 = V_2 - V_0$ ,  $T = O - V_0$ ,  $P = D \times E_2$  and  $Q = T \times E_1$ .

2. This linear equation can now be solved to find the barycentric coordinates of the intersection point  $(u, v)$  and its distance  $t$  from the ray origin.

## 2.2 Unit Triangle Algorithm

The so called Unit Triangle intersection algorithm performs ray transformations and consists of two stages [Woo05]. First the ray is transformed, using a triangle specific affine triangle transformation, to a coordinate system, in which the triangle looks like the unit triangle  $\Delta_{unit}$  with the edge points  $(1, 0, 0)$ ,  $(0, 1, 0)$  and  $(0, 0, 0)$ . In the second stage, a simple intersection test of the transformed ray with the unit triangle is done. The affine triangle transformation to a triangle  $\Delta = (a, b, c)$  is an affine transformation  $T_\Delta(x) = m \cdot x + n$  with  $m \in Mat_R(3 \times 3)$  and  $n \in R^3$  that maps the triangle  $\Delta$  to the unit triangle  $\Delta_{unit}$ .

## 2.3 Bounding Volume Hierarchies

Bounding volume hierarchies were successfully implemented on GPU. Thrane and Simonsen [Thr05] in fact compare kd-trees, uniform grids, and bounding volume hierarchies implemented on a GPU (2005-year hardware). They conclude the performance of BVHs is low, however higher than the performance of other two data structures when no ray packets are used. Carr et al. [Car06] implemented a variant of BVHs in combination with geometry images. Günther et al. [Gün07] use ray packets and yield interactive performance comparable or exceeding CPU-based implementation, but only for primary and shadow rays. Recently, Lauterbach et al. [Lau09] present an algorithm for fast BVH construction on a GPU, where they report performance comparable to kd-trees [Zho08] only for one scene. Torres et al. [Tor09] published an algorithm for stackless BVH traversal, where the use of stack is replaced by ropes connecting the nodes of a BVH in a sibling order.

## 3. IMPLEMENTATION

We have implemented a standalone compact program called RenderBro, that does not need the support of other 3<sup>rd</sup> party libraries along with Autodesk 3DS Max Plugin (both can be obtained at <http://renderbro.com>). Standalone program is capable of loading 3D scene form OBJ file format along with MTL materials files. 3DS Max plugin is capable to work with any kind of geometry loaded into editor in question. While the data structures are built offline on a CPU, the created data structures and scene geometry are transferred to a GPU and used for ray tracing algorithm entirely on the GPU. This methodology is sufficient to study the efficiency of shooting rays with different intersection algorithms. The traversal and intersection algorithms were implemented using Microsoft DirectX DirectCompute (Compute Shaders). Although this implementation limits target platforms to Microsoft Windows, it gives freedom to choose any GPU vendor to run GPU ray tracing, such as ATI/AMD, NVIDIA, Intel. DirectCompute code can be translated to C++ AMP version, which can be executed on any OS. We designed our solution to support as many hardware as possible, though only DirectX 10 compatible or newer hardware is supported. All shaders were compiled using latest Windows SDK 8.0 D3DCompiler\_45.

In Moller-Trumbore setting the geometry of a scene consisting solely of triangles is represented by a list  $L_v$  of vertices and list of materials  $L_m$ , where each triangle has a list of three indices to  $L_v$  plus an index to the  $L_m$ . In the Unit Triangle Test each triangle is represented directly by the affine transformation matrix. For our tests we implemented path-trace setting with physically-based importance sampled shading including Phong, Blinn-Phong, Lambertian Diffuse, Oren-Nayar Diffuse, Ashikhmin-Shirley, Glass and Perfect Mirror BSDFs.

The BVHs were built in top-down fashion with surface area heuristics using the centroids of bounding boxes for scene triangles, following the paper by Günther et al. [Gün07]. Each BVH node consists of AABB extents and indices to child nodes. If it's a leaf node, child indices are replaced with triangle offset along with triangle count. Those parameters are packed in 32 byte BVH node structure. As a BVH does not need to store the min and max intersection distances along the ray, only the node address is saved to the stack. Stack does not need to be shortened to only several entries, which minimizes the number of traversal steps. Serialization of write operation may occur as threads record their information. Each BVH node can contain any number of triangles. This hypothetically reduces the number of nodes of a hierarchy along with GPU memory needed and gives space for GPU traversal optimization. Traversal is done with “while-if” method and listed in appendix 8.3.

## 3.1 GPU Optimized Intersections

Moller-Trumbore and Unit triangle intersection tests are pretty straight-forward to implement and require a little knowledge of GPU architecture (see appendix, section 8.1). Such methods can experience poor register usage in architectures like VLIW which is used in many AMD GPUs.

In this work we introduce a method that strongly benefit from denser GPU register usage. The main idea is to exploit the wide vector width SIMD (Single Instruction Multiple Data) by testing intersection with one ray and four triangles at a time.

Firstly, at precompute time we need to try to fill BVH with four triangles per node, so each node will contain four pointers to triangle list. If this is not possible we would have one triangle per node at worst. Secondly, we need to align GPU scene data according to BVH node structure. So if particular node has only one triangle, we need to place three degenerate triangles in a triangle list to fulfill the alignment. Of course, this will result in a GPU memory footprint by the means of performance. Thirdly, when performing BVH traverse we will be able to linearly fetch four triangles. Here we will need to construct additional vectors, like:

```
// fetch triangle vertices
float3 v01, v11, v21;
float3 v02, v12, v22;
float3 v03, v13, v23;
float3 v04, v14, v24;
...
// construct per-component dir & orig vectors
float4 dir4x = ray.dir.xxxx;
float4 dir4y = ray.dir.yyyy;
float4 dir4z = ray.dir.zzzz;
float4 orig4x = ray.orig.xxxx;
float4 orig4y = ray.orig.yyyy;
float4 orig4z = ray.orig.zzzz;
```

This allows us to compute temporary values on per-component SIMD basis, so all non SIMD operations like scalar addition and multiplication can be performed on each triangle simultaneously. For example, when performing scalar multiplication on GPU we use only one computing block while using new approach we will perform four multiplication operations by the same cost (see appendix 8.2 for full listing):

```
// one triangle per pass
float divisor = dot(pvec, e1);

// four triangles per pass
float4 divisor4 = pvecx*e1x + pvecy*e1y +
pvecz*e1z;
```

Scene	# Triangles per BVH node	GPU Scene Size, MB	Moller-Trumbore, seconds (average)	Unit Triangle, seconds (average)	Quad Moller-Trumbore, seconds (average)	Quad Unit Triangle, Graphics (average)
Buddha, 100.006 triangles	1	9,35	103,33	103,94	-	-
	2	7,26	95,13	95,44	-	-
	3	6,38	92,72	93,07	-	-
	4	5,89	90,91	91,17	89,67	85,36
	8	5,15	92,57	94,01	-	-
	16	4,73	104,53	108,83	-	-
Bunny, 69.678 triangles	1	6,51	72,18	73,52	-	-
	2	4,87	65,48	66,67	-	-
	3	4,45	62,07	63,94	-	-
	4	3,99	60,98	62,17	57,35	52,84
	8	3,45	62,1	63,29	-	-
	16	3,14	69,67	70,61	-	-
Dragon, 100.012 triangles	1	9,35	210,89	211,67	-	-
	2	7,26	210,71	211,61	-	-
	3	6,35	210,68	211,44	-	-
	4	5,86	210,52	211,21	173,37	168,85
	8	5,11	210,69	211,37	-	-
	16	4,65	210,91	211,66	-	-

**Table 1:** Test scene properties, number of triangles per BVH leaf node, rendering times for different ray-triangle intersection method for Path Trace setting with 500 samples and max ray depth of 16. GPU NVIDIA GeForce GT 240M. Image resolution: 800x600.

Scene	# Triangles per BVH node	GPU Scene Size, MB	Moller-Trumbore, seconds (average)	Unit Triangle, seconds (average)	Quad Moller-Trumbore, seconds (average)	Quad Unit Triangle, seconds (average)
Buddha, 100.006 triangles	1	9,35	49,27	49,43	-	-
	2	7,26	49,78	50,31	-	-
	3	6,38	51,94	52,18	-	-
	4	5,89	54,15	55,16	34,3	31,37
	8	5,15	66,3	68,64	-	-
	16	4,73	95,22	98,38	-	-
Bunny, 69.678 triangles	1	6,51	33,27	33,79	-	-
	2	4,87	33,57	34,41	-	-
	3	4,45	34,68	35,55	-	-
	4	3,99	38,05	39,06	23,9	20,64
	8	3,45	46,57	47,86	-	-
	16	3,14	60,26	62,56	-	-
Dragon, 100.012 triangles	1	9,35	99,27	99,75	-	-
	2	7,26	99,41	100,96	-	-
	3	6,35	104,6	105,61	-	-
	4	5,86	108,35	111,5	57,99	54,91
	8	5,11	132,57	138,06	-	-
	16	4,65	188,55	198,07	-	-

**Table 2:** Test scene properties, number of triangles per BVH leaf node, rendering times for different ray-triangle intersection methods for Path Trace setting with 500 samples and max ray depth of 16. GPU AMD Radeon HD 6870. Image resolution: 800x600.

Scene	# Triangles per BVH node	GPU Scene Size, MB	Moller-Trumbore, seconds (average)	Unit Triangle, seconds (average)	Quad Moller-Trumbore, seconds (average)	Quad Unit Triangle, seconds (average)
Buddha	4	5,89	66,59	67,07	63,87	62,13
Bunny	4	3,99	56,82	57,52	53,18	51,17
Dragon	4	5,86	84,76	85,94	80,37	78,64

**Table 3:** Test scene properties, number of triangles per BVH leaf node, rendering times for different ray-triangle intersection methods for Path Trace setting with 1000 samples and max ray depth of 16. GPU NVIDIA GeForce GTX 560. Image resolution: 800x600.

Scene	# Triangles per BVH node	GPU Scene Size, MB	Moller-Trumbore, seconds (average)	Unit Triangle, seconds (average)	Quad Moller-Trumbore, seconds (average)	Quad Unit Triangle, seconds (average)
Buddha	4	5,89	38,03	39,41	33,43	31,35
Bunny	4	3,99	36,48	37,97	31,64	29,85
Dragon	4	5,86	65,56	67,09	56,72	54,92

**Table 4:** Test scene properties, number of triangles per BVH leaf node, rendering times for different ray-triangle intersection methods for Path Trace setting with 1000 samples and max ray depth of 16. GPU AMD HD7850. Image resolution: 800x600.

## 4. RESULTS

In this section we describe the results for measurement on three different scenes. We used scenes from individual objects courtesy of Stanford scene repository. These scenes are frequently used to test the performance of ray tracing and global illumination algorithms. Images were rendered in 800x600 pixels resolution. All performance results in this paper were measured on 4 different GPUs:

1. NVIDIA GeForce GT 240M (2009), 48 CUDA cores on 1210MHz, 1 GByte of memory with bandwidth of 54.4 GB/sec.
2. AMD HD 6870 (2010), 2 TFLOPs, 1120 Stream Processors on 900MHz, 1 GByte of memory with bandwidth of 134.4 GB/sec.
3. NVIDIA GeForce GTX 560 (2011), 2.1 TFLOPs, 336 CUDA cores on 1620-1900MHz, 1 GByte of memory with bandwidth of 128 GB/s.
4. AMD HD 7850 (2012), 1.76 TFLOPs, 1024 Stream Processors on 860MHz, 2 GByte of memory with bandwidth of 153.6 GB/s.

The static properties of data structures for all three scenes along with average computation time for path-tracing are shown in Tables 1 and 2. Performance results for BVHs build with different count of triangles per leaf node are shown in columns 4 and 5. Those results demonstrate that both the number of triangles per leaf node and the selected intersection method remarkably affect the performance. Results lead us to assumption that 4 triangles per leaf node is the optimal number for BVH traversal.

Moller-Trumbore kernel had proven to be up to 5% faster than Unit Triangle in all tests while Quad Unit Triangle kernel shown to be up to 14% faster than Quad Moller-Trumbore.

Our proposed Quad Unit-Triangle method brings moderate improvements of 5% to 11% on different generations of NVIDIA hardware except for Dragon scene setup on GT 240M GPU where it gain about 18% (tables 1 and 3). The situation is better on VLIW AMD/ATi hardware where it had shown to be up to 2x times faster than Moller-Trumbore (table 2). Furthermore we managed to analyze performance of the latest GPU generation AMD HD 7850 (table 4). As we expected, it showed consistent results in spite of new architecture and showed that Quad Unit-Triangle is approximately 20% faster than Moller-Trumbore method.

Good results are gained for VLIW architecture used in AMD GPUs where more functional units are available and may be scheduled by the compiler or hardware simultaneously. According to description of VLIW architecture, it's possible to perform compute operations along with memory access. We assume that this result is mainly achieved by performing more linear memory access to GPU global memory, avoiding branching by unrolling triangle-intersection loop and taking advantage of denser GPU register usage.

Things are bit different for NVIDIA GPUs like Fermi which internally operate in a SIMD manner by ganging multiple (32) scalar threads together into SIMD warps. If a warp's threads diverge, the warp serially executes both branches, temporarily disabling threads that are not on that path. Thus, ray tracing performance certainly can benefit from loop unrolling and more linear memory access. But the true cause of performance improvement lies much deeper in the GPU architecture and goes beyond the scope of this article.

## 5. CONCLUSION AND FUTURE WORK

We have shown that triangle intersection routines that tend to have good performance when used separately can behave badly

when used together with acceleration structures like BVH's due to incoherent memory access, lack of registers, and so on. So we focus our work on finding the robust combination of triangle intersection method and spatial data structure. For now it's the Quad Triangle intersection method used along with BVH.

As future work, the implementation could be extended by several other data structures, such as Kd-Trees, Uniform Grids along with few different ray-triangle intersection methods that can be efficiently mapped to GPUs and are likely to show unexpected results when used together.

Furthermore, shown results can hardly be called ambiguous as they are pretty much view dependent. Dragon scene showed 18% performance improvement on NVIDIA GT 240M for one angle of view. For different angles performance may vary, showing both improvement and deterioration. So, a part of our future work will be devoted to analysis of more complex and dynamic scenes where view dependency is not that great.

Unfortunately, we didn't manage to make in-depth performance study on latest discreet NVIDIA and integrated Intel GPUs. We would like to complete our research by taking this GPUs into account as a part of future work.

## 6. ACKNOWLEDGMENTS

This work has been supported by the Foundation for Assistance to Small Innovative Enterprises, grant no. 10031p/17006.

## 7. REFERENCES

- [Ail09] Aila, T., and Laine, S.. Understanding the Efficiency of Ray Traversal on GPUs. In Proceedings of High-Performance Graphics 2009, pages 145–150, New York, NY, USA, 2009. ACM.
- [Are88] J. Arenberg, “Ray-Triangle Intersection with Barycentric Coordinates”, in Ray Tracing News, edited by Eric Haines, v 1.1, n. 11, November 4, 1988
- [Bad90] Badouel, F.: An efficient Ray-Polygon intersection, Graphic Gems, Academic Press, pp:390-393, 1990.
- [Car06] Carr, N.A., Hoberock, J., Crane, K., and Hart, J.C. Fast GPU ray tracing of dynamic meshes using geometry images. In GI '06: Proceedings of Graphics Interface 2006, pages 203–209, Toronto, Ont., Canada, Canada, 2006. Canadian Information Processing Society
- [Gün07] Günther, J., Popov, S., Seidel, H.-P., and Slusallek, P. Realtime Ray Tracing on GPU with BVH-based Packet Traversal. In Proceedings of the IEEE/Eurographics Symposium on Interactive Ray Tracing 2007, pages 113–118, September 2007.
- [Hav10] Jiří Havel, Adam Herout, "Yet Faster Ray-Triangle Intersection (Using SSE4)," IEEE Transactions on Visualization and Computer Graphics, vol. 16, no. 3, pp. 434-438, May-June 2010, doi:10.1109/TVCG.2009.73
- [Lau09] Lauterbach, C., Garland, M., Sengupta, S., Luebke, D., and Manocha, D. Fast BVH Construction on GPUs. Computer Graphics Forum, 28(2):375–384, April 2009. (Proceedings of Eurographics 2007).
- [Möl97] Möller, T., Trumbore, B.: Fast, minimum storage ray-triangle intersection, Journal on Graphic Tools, Vol.2, No.1, pp.21-28, 1997.
- [Seg01] Rafael J. Segura and Francisco R. Feito. Algorithms to Test RayTriangle Intersection. Comparative Study. In Vaclav Skala, editor, WSCG 2001 Conference Proceedings, February 2001.
- [She07] M. Shevtsov, A. Soupikov, and A. Kapustin, “Ray-Triangle Intersection Algorithm for Modern CPU Architectures,” in Proceedings of GraphiCon 2007, 2007, pp. 33–39.

[Sny87] Snyder, M., Barr, A.H.: Raytracing complex models containing surface tessellations, Proceedings of the 14th annual conference on Computer Graphics, 1987, Vol.21, No.4, pp.119-128, 1987

[Thr05] Thrane, N., and Simonsen, L.O. A comparison of acceleration structures for GPU assisted ray tracing. M.Sc. Thesis, University of Aarhus, Denmark, 2005

[Tor09] Torres, R., Martin, P.J., and Gavilanes, A. Ray Casting using a Roped BVH with CUDA. In 25<sup>th</sup> Spring Conference on Computer Graphics (SCCG 2009), pages 107–114, Budmerice, Slovakia, April 2009.

[Woo05] S. Woop, J. Schmittler, P. Slusallek, RPU: A Programmable Ray Processing Unit for Realtime Ray Tracing, ACM Transactions Graphics, 24(3), pp. 434-444, 2005

[Wal04] I. Wald, “Realtime ray tracing and interactive global illumination”, PhD thesis, Saarland University, 2004.

[Zho08] Zhou, K., Hou, Q., Wang, R., and Guo, B. Real-time KD-tree construction on graphics hardware. In SIGGRAPH Asia '08: ACM SIGGRAPH Asia 2008 papers, pages 1–11, New York, NY.

## 8. APPENDIX

### 8.1 Casual Moller-Trumbore GPU Ray-Triangle Intersection Routine (HLSL code)

```
float intersect(float3 orig, float3 dir, float3 v0,
float3 v1, float3 v2)
{
    float3 e1 = v1 - v0;
    float3 e2 = v2 - v0;

    float3 normal = normalize(cross(e1, e2));
    float b = dot(normal, ray.dir);

    float3 w0 = ray.orig - v0;
    float a = -dot(normal, w0);
    float t = a / b;

    float3 p = ray.orig + t * ray.dir;
    float uu, uv, vv, wu, wv, inverseD;
    uu = dot(e1, e1);
    uv = dot(e1, e2);
    vv = dot(e2, e2);

    float3 w = p - v0;
    wu = dot(w, e1);
    wv = dot(w, e2);
    inverseD = uv * uv - uu * vv;
    inverseD = 1.0f / inverseD;

    float u = (uv * wv - vv * wu) * inverseD;
    if (u < 0.0f || u > 1.0f)
        return -1.0f;

    float v = (uv * wu - uu * wv) * inverseD;
    if (v < 0.0f || (u + v) > 1.0f)
        return -1.0f;

    UV = float2(u, v);
    return t;
}
```

### 8.2 Quad Moller-Trumbore GPU Triangle-Ray Intersection Routine (HLSL code)

```
float intersect(float3 orig, float3 dir, float3 v01,
float3 v11, float3 v21, float3 v02, float3 v12, float3
v22, float3 v03, float3 v13, float3 v23, float3 v04,
float3 v14, float3 v24)
{
    float3 e11 = v11 - v01;
    float3 e21 = v21 - v01;
    float3 e12 = v12 - v02;
```

```
float3 e22 = v22 - v02;
float3 e13 = v13 - v03;
float3 e23 = v23 - v03;
float3 e14 = v14 - v04;
float3 e24 = v24 - v04;
float4 v0x = float4(v01.x, v02.x, v03.x, v04.x);
float4 v0y = float4(v01.y, v02.y, v03.y, v04.y);
float4 v0z = float4(v01.z, v02.z, v03.z, v04.z);
float4 elx = float4(e11.x, e12.x, e13.x, e14.x);
float4 ely = float4(e11.y, e12.y, e13.y, e14.y);
float4 elz = float4(e11.z, e12.z, e13.z, e14.z);
float4 e2x = float4(e21.x, e22.x, e23.x, e24.x);
float4 e2y = float4(e21.y, e22.y, e23.y, e24.y);
float4 e2z = float4(e21.z, e22.z, e23.z, e24.z);
float4 dir4x = ray.dir.xxxx;
float4 dir4y = ray.dir.yyyy;
float4 dir4z = ray.dir.zzzz;
float4 pvecx = dir4y*e2z - dir4z*e2y;
float4 pvecy = dir4z*e2x - dir4x*e2z;
float4 pvecz = dir4x*e2y - dir4y*e2x;
float4 divisor = pvecx*elx + pvecy*ely + pvecz*elz;
float4 invDivisor = float4(1, 1, 1, 1) / divisor;
float4 orig4x = ray.orig.xxxx;
float4 orig4y = ray.orig.yyyy;
float4 orig4z = ray.orig.zzzz;
float4 tvecx = orig4x - v0x;
float4 tvecy = orig4y - v0y;
float4 tvecz = orig4z - v0z;
float4 u4;
u4 = tvecx*pvecx + tvecy*pvecy + tvecz*pvecz;
u4 = u4 * invDivisor;
float4 qvecx = tvecy*elz - tvecz*ely;
float4 qvecy = tvecz*elx - tvecx*elz;
float4 qvecz = tvecx*ely - tvecy*elx;
float4 v4;
v4 = dir4x*qvecx + dir4y*qvecy + dir4z*qvecz;
v4 = v4 * invDivisor;
float4 t4;
t4 = e2x*qvecx + e2y*qvecy + e2z*qvecz;
t4 = t4 * invDivisor;
float t = -1.0f;

if(t4.x < t && t4.x > 0)
    if(u4.x >= 0 && v4.x >= 0 && u4.x + v4.x <= 1)
        t = t4.x;

if(t4.y < t && t4.y > 0)
    if(u4.y >= 0 && v4.y >= 0 && u4.y + v4.y <= 1)
        t = t4.y;

if(t4.z < t && t4.z > 0)
    if(u4.z >= 0 && v4.z >= 0 && u4.z + v4.z <= 1)
        t = t4.z;

if(t4.w < t && t4.w > 0)
    if(u4.w >= 0 && v4.w >= 0 && u4.w + v4.w <= 1)
        t = t4.w;

return t;
}
```

### 8.3 BVH Traversal Routine (HLSL code)

```
struct BvhCell
{
    float4 vmin; //float3 min + uint children
    float4 vmax; //float3 max + uint count
};

bool RayIntersectScene(Ray ray)
{
    uint stack[64], stackPos = 0, node = 0;
    float t = FLT_MAX;
    bool intersect = false;
    BvhCell cellLeft, cellRight;
    BvhCell current = GetNode(node);

    while(1)
    {
        uint count = GetNodeTriangleCount(current);
        if(count > 0)
        { // Leaf Node
            uint offset = GetNodeTriangleOffset(current);
            intersect = RayTrisTest(ray, t, offset, count);
            if(stackPos > 0)

```

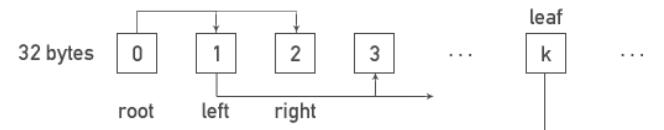
```

{
    node = stack[--stackPos];
    current = LoadNode(node);
}
else return intersected;
}
else
{
    uint leftNode = GetLeftChildID(node);
    uint rightNode = GetRightChildID(node);
    float lMin, rMin;
    cellLeft = GetNode(leftNode);
    cellRight = GetNode(rightNode);
    bool wantLeft = RayAABBTest(ray, cellLeft, lMin);
    bool wantRight = RayAABBTest(ray, cellRight,
        rMin);
    if(wantLeft && wantRight)
    {
        bool firstLeft = leftMin < rightMin;
        if(firstLeft)
        {
            current = cellLeft;
            node = leftNode;
            stack[stackPos++] = rightNode;
        }
        else
        {
            current = cellRight;
            node = rightNode;
            stack[stackPos++] = leftNode;
        }
    }
    else if(wantRight)
    {
        current = cellRight;
        node = rightNode;
    }
    else if(wantLeft)
    {
        current = cellLeft;
        node = leftNode;
    }
    else
    {
        if(stackPos > 0)
        {
            node = stack[--stackPos];
            current = GetNode(node);
        } else return intersected;
    }
}
return intersected;
}

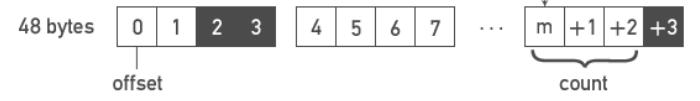
```

## 8.4 BVH Data Layout

### BVH Nodes:



### Node Triangles:



— degenerate triangle

## Robotic aided surgery in full knee replacement

Yuncai Liu

Shanghai Jiao Tong University, P. R. China

### Abstract

Every year, thousands of patients suffer from joint diseases, such as rheumatoid arthritis or osteoarthritis, and need total knee replacement (TKR) surgery to recover their normal functions. Presently, the positioning of prosthetic components in surgeries mainly depends on clinic experiences of doctors and special surgical guiding devices.

To avoid the limitations of jig-based TKR systems, we devoted to develop a robot/computer assisted surgery system, which can obtain a better operation precision and surgical quality. The system that we developed is a CT-free visual navigated robot assisted TKR surgery system called WATO-II. In WATO-II, both the infrared stereo cameras and the cutting tool are fixed on the end-effector of the robot. During the surgery, WATO-II automatically determines the position of cutting planes and establishes the mechanical axis according to the infrared markers, then guides the robot complete the surgery.

In this talk, the methods of system calibration, the surgical planning and navigation are discussed. We proposed an integrated calibration algorithm for the cameras, the stereo rig and the robot hand-eye, the infrared probe calibration algorithm, the tool calibration algorithm and physiological point based femoral and tibia positioning and calculation algorithm. WATO-II makes use of the physiological characteristics of the knee and the stereo navigation system, can get high-accurate locations of the femur and the tibia and achieve high-precision cutting of the bone, which fulfills the requirements of cadaver trials and is very meaningful toward clinic applications.

A lot of experiments, including gesso bones and human bones experiments, are conducted to test the efficiency and accuracy of the system. The experimental results show that WATO-II system can provide much high surgical accuracy and efficiency than manual operations.

### About the author

Yuncai Liu is a professor at Shanghai Jiao Tong University, P. R. China.

## Two Approaches for Noise Filtering in 3D Medical CT-Images

Maria V. Storozhilova, Alexey S. Lukin, Dmitry V. Yurin, Valentin E. Sinitsyn

<sup>1</sup>Laboratory of Mathematical Methods of Image Processing

Department of Computational Mathematics and Cybernetics

Lomonosov Moscow State University, Moscow, Russia

<sup>2</sup>Radiology Department at Federal Center of Medicine and Rehabilitation, Moscow, Russia

<sup>1</sup>[mariastorozhilova@gmail.com](mailto:mariastorozhilova@gmail.com), <sup>1</sup>[lukin@graphics.cs.msu.ru](mailto:lukin@graphics.cs.msu.ru), <sup>1</sup>[yurin@cs.msu.ru](mailto:yurin@cs.msu.ru), <sup>2</sup>[vsini@mail.ru](mailto:vsini@mail.ru)

### Abstract

Noise in 3D computer tomography (CT) images is close to white and becomes large when patient radiation doses are reduced. We propose two methods for noise reduction in CT images: 3D extension of fast rank algorithms (Rank-2.5D) and 3D extension of a non-local means algorithm (NLM-2.5D). We call both our algorithms “2.5D” because the 3-dimensional NLM algorithm is slightly asymmetric by slice axes, while our Rank algorithms, being fully symmetric mathematically and by results, have some implementation asymmetry. A comparison of the methods is presented. It is shown that NLM-2.5D method has the best quality, but is very computationally expensive: its complexity quickly rises as a function of the neighborhood size, while Rank-2.5D only shows a linear growth. Artificial test sequences are used for signal-to-noise performance measurements, while real CT scans are used for visual assessment of results.

**Keywords:** medical imaging, CT, DICOM, filtering, enhancement, noise reduction, denoising, 3D image processing.

### 1. INTRODUCTION

The problem of noise reduction in digital images has a long history. The first algorithms were linear filters [7], such as convolutions with a low-pass window function (rectangular or Gaussian), frequency-domain filtering, Wiener filtering. The problem with linear methods is inevitable loss of quality of the image: loss of sharpness, blurring of edges, ringing effects.

The next wide class of image filtering methods has been introduced in [15], they are called rank algorithms. The most well-known of these methods is the median filter, which has fast computational algorithms [3], [14], [6]. Median filter preserves sharp edges, but rounds the corners in the image. Recently we have proposed fast algorithms for other types of range filters [10], [11]. Before their existence, practical applications of such filters were limited. Unlike median filtering and linear methods, rank algorithms reduce the contrast of edges without blurring or changing their shape. A well-known bilateral filtering algorithm can be roughly considered as being a rank filter too. If the existing Gaussian spatial and range kernels in [12] are replaced with rectangular kernels, a so-called  $\varepsilon_V$  averaging algorithm is produced [15]. The only fast algorithms known for the bilateral filter are approximating algorithms.

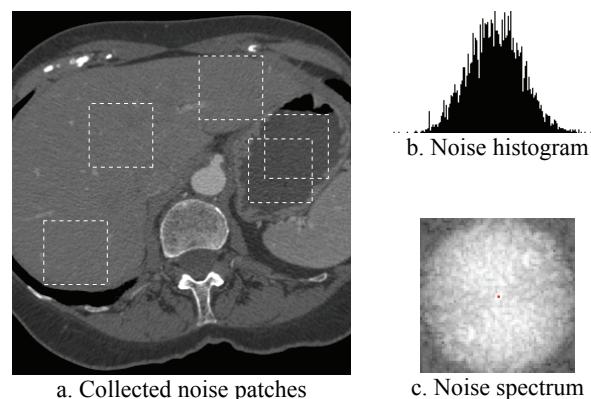
Most recent works in noise reduction show the advantage of methods which average pixels depending on their neighborhood statistics, not just pixel values [1], [2]. The first method, known as non-local means, calculates averaging weights using similarity of pixel neighborhoods (patches). The second one is more complex and consists of two stages. On the first stage a rough frequency-domain filtering is performed to facilitate the search of similar patches. The second stage is the joint filtering of groups of similar patches from the source image. In [2] there

are comparisons of the proposed method with 5 other algorithms.

This paper focuses on noise reduction in computer tomography (CT) scans. CT images capture the density of a sequence of slices of a human body. These slices are obtained with a small fixed stride, which is perpendicular to the slice plane and is comparable with the pixel size in each slice. Together they represent a 3D set of data, so traditional 2D methods of image denoising are less than optimal because they fail to exploit a high degree of data correlation between slices. Independent noise reduction in each slice may cause difference in color and position of edges of objects between slices. Our main objective is to reduce noise and avoid the loss of small low-contrast image areas. Such areas may contain symptoms of an illness and it is important to maintain sharp edges and prevent the loss of information for them. This requirement limits the use of median filtering and other algorithms based on the image blurring as they may shift the edges.

Noise is always present in CT images. Interestingly, as the noise reduction methods advance over time, one can expect even noisier raw data to be coming from the scanner, because it means lower radiation dose for the patient. Specialized algorithms for denoising of CT images are an area of active research [9], [13].

The noise spectrum in our images is close to white: slightly low-pass, but without evident directionality (Figure 1c). The amplitude p.d.f. is close to Gaussian (Figure 1b). This justifies application of standard noise reduction methods, most of which have been formulated for additive white Gaussian noise. Another study of noise in CT images is given in [4], [5].



**Figure 1:** Analysis of noise in our CT images.

This paper proposes two approaches to noise reduction. A higher-quality algorithm is based on a non-local means [1] adapted to a 3-dimensional image data. We are using a fast, but not quite symmetric variant (we call it NLM-2.5D) developed in our previous work for video processing [8]. The second approach (Rank-2.5D) is using the adaptation of fast rank filters which do not blur or shift the edges of the image [10], [11]. The proposed algorithm is fully symmetric in 3 dimensions and

comprises the direct extension of  $\varepsilon_V$  averaging [15] on a 3D space, with an adaptive choice of  $\varepsilon$  in every point of the image. This  $\varepsilon$  is calculated using another rank algorithm. Every slice is processed using information from a *collection* of adjacent slices. The implementation of the algorithm is not symmetric: it is based on sequential processing of 2D slices and 1D filtering across slices. This allows considerable savings in memory, but slows down the computation. The processing time becomes proportional to the number of slices in the collection (neighborhood). A comparison of speed and PSNR quality is carried out on phantom images with an artificially generated noise. Visual results on real medical images are also provided.

## 2. EV - FILTERING

We start with some necessary definitions, following [15]. Consider the current  $v_0$  and some neighborhood S of pixel  $v_0$  that contains  $N$  pixels. Frequently S-neighborhood has a square, round or octagonal shape [15], [6].

**Definition 1.** A rank series  $\{v(r)\}$  is a one-dimensional sequence of  $N$  pixels from S whose elements are sorted in an ascending order with respect to their values:  $\{v(r) \leq v(r+1), r = 0..N-1\}$ .

**Definition 2.** Pixel's  $v_R$  rank  $R$  is the position of the  $v_R$  element in the rank series.  $R = \text{rank}(v_R)$ .

Let's consider some selected pixel  $v_c \in S$ ; for example we can consider  $v_c = v_0$ ,  $v_c = \text{med}\{v(r)\}$ , or  $v_c = \text{mean}\{v(r)\}$  as a pivot. It should be noted that under such definition the pivot pixel do not always satisfy conditions  $v_c \in S$  or  $v_c \in \{v(r)\}$ .

Then

**Definition 3.**  $\varepsilon_V$  (or EV) neighborhood is a subset of pixels set  $\{v(r)\}$  whose values deviate from the value of the pivot pixel at most by a predetermined quantity  $\varepsilon$ :

$$\varepsilon_V(v_c) = \{v(r) : |v(r) - v_c| \leq \varepsilon\} \quad (1)$$

As it can be seen from the definition,  $\varepsilon_V$  neighborhood average can be treated as a simplified bilateral filter [12], where bilateral filter parts that depend both on distance and on pixel brightness, are represented by rectangular functions instead of Gaussians. For this reason  $\varepsilon_V$  filtering should keep the edges of the objects sharp, assuming the parameter  $\varepsilon$  is properly chosen.

## 3. 3D RANK FILTERING ALGORITHM

It is proposed to use a 3-dimensional  $\varepsilon_V$  filtering algorithm with an adaptive search of parameter  $\varepsilon$  for denoising of CT images. Fast rank algorithms [10] are based on multiscale histogram approach. Either fast rank algorithms or a lazy calculations technique remain the same in a 3D case, though some difficulties occur with column histogram maintaining [6], [11]. Thus if a 2-dimensional image with the size of 512x512 pixels requires maintaining of 512+1 multiscale histograms, then in a 3 dimensional case we will have to maintain 512x512+1=262145 multiscale histograms for a sequence of images with the same size. Taking into account the large size of multiscale histograms with the specific additional information [10], we may see that the number above is at the limit of memory size of 32-bit computers. We propose to use a separable approach for reduction of complexity by processing 2D images. At first,  $\varepsilon_V$  neighborhood average will be calculated

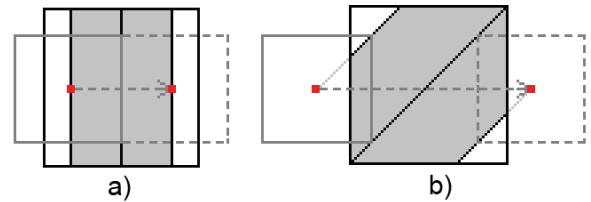
separately with the same pivot for all  $N$  slices that are used for denoising of the current image. Then the 1D variant of  $\varepsilon_V$  filtering will be applied to the result. Really, if the inequality (1) is true for a number of subsets it is true for the union of these subsets too.  $\varepsilon_V$ -averaging means calculation of the ratio of sum of values to number of elements, so it is sufficient to sum up the sums for each subsets and the number elements in each subsets and than calculate their ration once.

For the correct choice of the  $\varepsilon$  parameter we propose to calculate the intensity variance in a square neighborhood of small radius  $R_{\text{disp}}$  for each pixel of the image. This step allows determining the uniformity of an area of a specified radius around the pixel. For a flat area with insignificantly varying color the intensity variance value will provide the noise variance.

The choice of  $R_{\text{disp}}$  is based on the fact that the noise in CT images is almost non-correlated (white). So the neighborhood can have an arbitrary radius. Since our main goal is to preserve small low-contrast objects, it is essential to choose  $R_{\text{disp}}$  in accordance with the size of the object that needs to be preserved. Since the variance will be high on the boundaries of areas with different colors, it is proposed to apply a some kind of minimum filter to the variance image, using the neighborhood of a larger radius. This allows propagating correct values of noise variation to regions with excessive variance.

So the next step is the min-filtering of variance by averaging  $K$  pixels with the smallest rank of the neighborhood of a larger radius. To completely suppress high-variance values on boundaries it is proposed to use the neighborhood of a higher radius  $1.5 \cdot R_{\text{disp}}$ . Let us consider possible cases for location of a boundary within the neighborhood. Since the radius of the neighborhood is small enough, we assume that the boundary between the objects inside the sliding window is a straight line.

In the easiest case (Figure 2a) the amount of pixels that represent variance outside the edge area is about 30% of the bigger neighborhood. In the worst case (Figure 2b) the amount of pixels for the variance of the outside-edge area is about 10%. These values can be easily proved by simple geometrical calculations.



**Figure 2:** The area with high variance value for a) the vertical edge; b) the diagonal edge

Variance minimized in such a way corresponds to adaptive  $\varepsilon$  value. In order to control the degree of noise reduction it is proposed to use a multiplier parameter  $M$ . In general the coefficient  $M$  is user-defined, but it is essential to use the range of [2...6] in order to simultaneously suppress the majority of noise pixels and to avoid blurring of the edges.

Then  $\varepsilon_V$  algorithm with the  $\varepsilon$  value that was obtained on the previous step and the  $R_{\text{disp}}$  neighborhood radius is performed for each slice. It is important to emphasize that we take not the central intensity value of the current slice but the intensity value of a source image with corresponding coordinates as a

central pixel for  $\varepsilon_V$  neighborhood average algorithm. This condition is important because we apply the algorithm not to a 2D neighborhood, but to the volume region.

To extend the algorithm to a 3D space, it is proposed to calculate the total sum and the total number of elements involved in the averaging for the current pixel of the current slice. As a result, for each pixel of each slice we receive the structure that contains the total sum and the amount of the  $\varepsilon_V$  neighborhood elements.

The last step is  $\varepsilon_V$  filtering for slices. It is proposed to use the minimized variance for the source image multiplied by  $M$  as the value of  $\varepsilon$  for each pixel of the source image. The averaging is performed with only one structure from each slice (i.e. if we use  $N$  slices, only  $N$  elements will take part in averaging). For the pixel  $I(x, y)$  of the source image the necessary structures will be located at the same coordinates in the corresponding slice. The number of slices for processing is defined automatically in accordance with the distance between slices.

Let  $N$  be the number of images (slices) for denoising of one image,  $I_{src}$  be the source image,  $I_i$  be the  $i$ -th image from the array of image slices,  $R_{disp}$  be the radius of the variance calculation window,  $D(I_i)$  be the variance of the  $i$ -th image,  $M$  be the variance multiplier,  $Reps$  be the radius for a 2D spatial  $\varepsilon_V$  filtering. The following algorithm is proposed for CT image denoising.

**Algorithm 1.** 2.5D rank denoising algorithm.

```

1.   for  $i := 0, i < N$  do
2.     for each pixel  $I_i(x, y)$  of  $I_i$  do
3.       Compute the variance of  $I_i(x, y)$  with the window radius  $R_{disp}$ ;
4.       Minimize variance  $D(I_i(x, y))$  for  $I_i$  by averaging  $K$  elements with the smallest rank using the window radius of  $1,5 \cdot R_{disp}$ ;
5.       Perform  $\varepsilon_V$  filtering around the current pixel's  $I_i(x, y)$  neighborhood to obtain the total sum and total count of elements for  $I_i(x, y)$ .  $\varepsilon(x, y) = M \cdot D(I_i(x, y))$ , the window radius is  $R_{disp}$  and  $I_{src}(x, y)$  is taken as the central pixel of the neighborhood;
6.     end for;
7.   End for;
8.   for each pixel  $I_{src}(x, y)$  of  $I_{src}$  do
9.     Perform 1D  $\varepsilon_V$  filtering with  $\varepsilon(x, y) = M \cdot D(I_{src}(x, y))$  for  $N$  corresponding elements  $I_i(x, y)$ ,  $i = \overline{1, N}$ ;
10.  End for.

```

#### 4. 3D NON-LOCAL MEANS FILTERING

Bilateral filtering algorithm is well known in image processing for its simplicity and edge-preserving properties [12]. The output pixel value  $I_{out}(x, y)$  is formed as a weighted sum of pixel values from the neighborhood  $\Omega$ :

$$I_{out}(x, y) = \frac{1}{\sum W} \sum_{i, j \in \Omega} W(x, y, j, i) \cdot I(x + j, y + i) \quad (2)$$

The weights  $W$  depend on geometric distance and color difference between pixels  $(x, y)$  and  $(x + j, y + i)$  in order to facilitate averaging of pixels with similar values:

$$W(x, y, j, i) = \exp \frac{j^2 + i^2}{-2\sigma^2} \cdot \exp \frac{(I(x + j, y + i) - I(x, y))^2}{-2\rho^2} \quad (3)$$

Non-local means is a relatively novel method of image filtering that builds upon a bilateral algorithm. A formula (3) for pixel similarity in bilateral filtering considers colors and spatial coordinates of two pixels. In the non-local means algorithm, this formula instead considers the *context* of two pixels [1]. Specifically, instead of comparing values of two pixels, the algorithm compares the content of *image patches*  $v$  around two pixels:

$$W(x, y, j, i) = \exp \frac{\|v(x + j, y + i) - v(x, y)\|_2^2}{-2\rho^2} \quad (4)$$

The squared norm of pixel-wise patch differences in formula (4) ensures that only pixels with a similar surrounding content are averaged together.

The extension of this method on a 3-dimensional space is straightforward. For the standard 2D image processing, the neighborhood  $\Omega$  from formula (2) is a circle or a square around the central pixel. For the 3D filtering, we extend the neighborhood to be a sphere or cube in the 3D image space: it includes several image slices that are adjacent to the processed pixel. A 2-dimensional summation in formula (2) turns into a 3-dimensional summation and calculation of weights in formula (4) is adjusted accordingly: similar patches are searched among the array of several adjacent slices. Since the compared patches are still 2-dimensional, we call this method NLM-2.5D.

For improved speed of calculations we employ an optimization from [8] for sparse update of weights  $W$ .

#### 5. SYNTHETIC PHANTOM GENERATION

For testing the noise reduction capability of our algorithms we have generated a synthetic phantom image. We do not use real phantom images because they always contain noise due to CT-imaging generation process. An attempt to shoot a phantom without noise results in high-dose shooting which is radically different from real medical imaging conditions. Our synthetic phantom construction (see Figure 3) is described below. The phantom body is cylindrical. It has two cylindrical “organs” with different radioparity. These two “organs” are connected by two thin truncated cones (“processes” or “ligaments”), with radioparity smoothly changing from one organ to another. Each “organ” has one truncated conic “vessel” with different radioparity. These truncated cone shapes are used for thin objects in order to detect the extent of damage to the phantom along the slice axis in the process of filtering (if any).

We model all “body”, “organs”, “ligaments”, and “vessels” with color  $c_j$  and density function

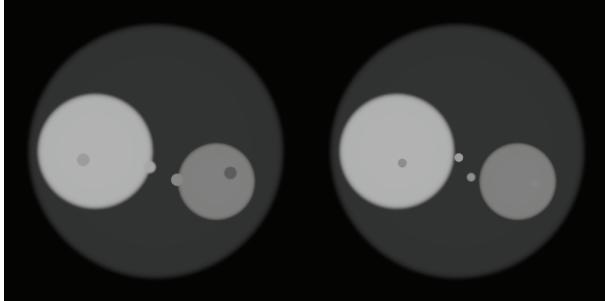
$$\rho(\bar{r}) = \exp \left\{ - \left( \frac{\bar{r} - \bar{r}_0}{R} \right)^{2n} \right\}, \bar{r} = (x, y) \quad (5)$$

which is used smooth object boundaries, mainly to avoid pixelated (aliased) shapes. The center of small objects (“ligaments”

and “vessels”) depends on a slice position  $z : \mathbf{r}_0(z), R(z)$ , so density function depends on all 3 coordinates  $\rho(\mathbf{r}) = \rho(x, y, z)$ . The resulting slice image set is obtained recursively as

$$\begin{aligned} I_0(x, y, z) &= \rho_{body}(\mathbf{r})c_{body} \\ I_j(x, y, z) &= \rho_j(\mathbf{r})c_j + (1 - \rho_j(\mathbf{r}))I_{j-1}(x, y, z) \end{aligned} \quad (6)$$

where large objects are added first. Typical simulated phantom images are presented in Figure 3.

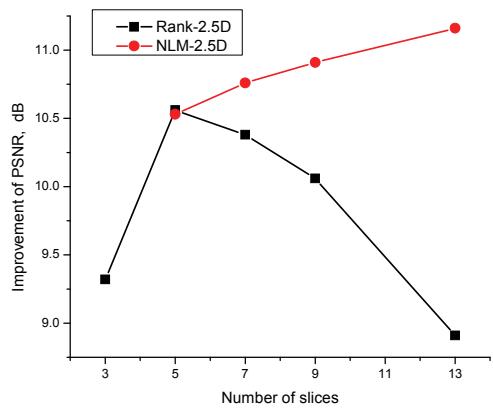


**Figure 3:** Simulated noise-free phantom images

The sharpness of edges is  $n = 30$  for body,  $n = 20$  for “organs”, and  $n = 4 \dots 12$  for different “ligaments” and “vessels”. The size of each slice is  $512 \times 512$  pixels and the total number of slices is 100. Thin object radii  $R(z)$  are from 5 to 12 pixels.

## 6. RESULTS

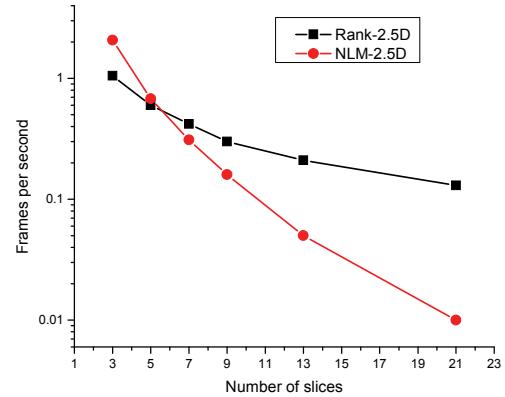
Figure 4 shows the results of a phantom CT image denoising. The Gaussian noise which corresponds to the real noise distribution on CT images was added to the synthesized phantom slices. The measured value is the improvement of PSNR in decibels between processed images and the ground-truth noise-free image. Each algorithm has been run with optimal parameters which maximize PSNR for each size of the neighborhood. The optimized parameter for Rank-2.5D was  $M$ , while the optimized parameter for NLM-2.5D was  $\rho$ . The patch size in NLM has been set to  $8 \times 8$  pixels, while the “pixel” size has been set to  $2 \times 2$  pixels.



**Figure 4:** Improvement of PSNR for the phantom image filtered by different algorithms, depending on the neighborhood size  $N$ .

The maximum in the curve for Rank-2.5D algorithm occurs when the neighborhood size is of order of smallest significant object size (“ligaments” and “vessels”). When the neighborhood becomes larger, small objects are suppressed as noise.

It can be seen that the algorithms are able to exploit high degree of correlation between image slices, which is reflected in PSNR and visual quality.



**Figure 5:** Speed of our C++ implementation of the proposed algorithms on a 2.8-GHz desktop.



**Figure 6:** Original noisy CT image, result of Rank-2.5D algorithm, result of NLM-2.5D algorithm.

Figure 5 compares the speed of the proposed algorithms on a CT scan with  $515 \times 512$ -pixel slices. It shows that the Rank-2.5D

algorithm has a linear complexity growth depending on the neighborhood size, which makes it suitable for future high-resolution CT scanners. The NLM-2.5D algorithm's complexity grows much faster.

Figure 6 shows the result of real CT image denoising with Rank-2.5D and NLM-2.5D algorithms operating on a neighborhood of  $7 \times 7 \times 5$  pixels (the last '5' being the number of slices). The value of parameter M in Rank-2.5D method has been set to 3.5 for good visual results. This is lower than the value of M = 5 which optimized PSNR in our experiments.

Subjectively, NLM-2.5D approach provides more accurate and clear image while Rank-2.5D leaves some blurred patches in high-contrast areas.

## 7. CONCLUSION

Two methods for 3-dimensional noise reduction in CT images have been presented. The evaluation shows that they are able to effectively exploit the existing correlation between CT slices for improvement of the resulting image quality. The presented Rank-2.5D shows only moderate growth in computational complexity depending on the size of neighborhood. It is easy to see from Figure 5 that when the neighborhood size in slices is larger than 6, Rank-2.5D becomes faster than NLM-2.5D. The maximum quality of Rank-2.5D is achieved when the neighborhood size is approximately equal to the minimal useful object size in the image.

## 8. AKNOLEDGMENTS

This work was supported by the Federal Targeted Program "R&D in Priority Fields of the S&T Complex of Russia 2007-2013".

## 9. REFERENCES

- [1] Buades A., Morel J.M. "A Non-Local Algorithm for Image Denoising" // IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, vol. 2, 20–26 June 2005, pp. 60–65.
- [2] Dabov K., Foi A., Katkovnik V., Egiazarian K., "Image denoising by sparse 3D transform-domain collaborative filtering," IEEE Trans. Image Process., vol. 16, no. 8, pp. 2080–2095, August 2007.
- [3] Huang T., Yang G., Tang G. "A Fast Two-Dimensional Median Filtering Algorithm" // IEEE Trans. Acoust., Speech, Signal Proc., 1979, Vol. 27, No. 1, P. 13–1.
- [4] Kijewski M.F., Judy P.F. "The noise power spectrum of CT images" // Phys. Med. Biol., 1987, Vol. 32, No 5, pp. 565–575. Printed in the UK.
- [5] "Radiology of the Skull and Brain. Vol. 5. Technical aspects of computed tomography" Ed. by T. H. Newton, and D. G. Potts, 585 p., 1981 (C. V. Mosby Co, St. Louis), ISBN 0-8016-3662-0.
- [6] Perreault S., Hebert P. "Median Filtering in Constant Time" // IEEE Transactions on Image Processing, 2007, Vol. 16, pp. 2389–2394.
- [7] Pratt W.K. "Digital Image Processing: PIKS Scientific inside" (4th ed.) Wiley-Interscience, John Wiley & Sons, Inc., Los Altos, California, 2007, 782 p.
- [8] Putilin S., Lukin A. "Non-local means method modification for noise suppression in video" (in Russian) // In: 17-th International Conference on Computer Graphics GraphiCon'2007, pp. 257–259.

[9] Reiter M., Zauner G. "Denoising of Computed Tomography Images using Multiresolution Based Methods" // European Conference on Non-Destructive Testing, Berlin, 25–29 Sept. 2006, Poster.

[10] Storozhilova M.V., Yurin D.V. "Fast Rank Algorithms Based on Multiscale Histograms" // In: 21-th International Conference on Computer Graphics GraphiCon'2011. Moscow, Russia, 2011, pp. 132–135.

[11] Storozhilova M.V., Yurin D.V. "Fast Rank Algorithms with Multiscale Histograms Lazy Updating" // 8th Open German-Russian Workshop "Pattern Recognition and Image Understanding" (OGRW-8-2011), November 2011, pp. 380–383.

[12] Tomasi C., Manduchi R. "Bilateral Filtering for Gray and Color Images" // Proceedings of the IEEE Sixth International Conference on Computer Vision (ICCV'98), January 1998, pp. 839–846.

[13] Trinh D.H., Luong M., Rocchisani J.-M., Pham C.D., Pham H.D., Dibos F. "An Optimal Weight Method for CT Image Denoising" // Journal of Electronic Science and Technology, June 2012, V.10, No.2, pp. 124–129.

[14] Weiss B. "Fast median and bilateral filtering", ACM Transactions on Graphics (TOG), July 2006, Vol. 25 n.3, pp. 519–526.

[15] Yaroslavsky L.P., Kim V. "Rank Algorithms for Picture Processing" // Computer Vision, Graphics and Image Processing, 1986, Vol. 35, pp. 234–258.

## About the authors



Maria V. Storozhilova, is a student at Chair of Mathematical Physics, Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University, Russia. Her contact email is [mariastorozhilova@gmail.com](mailto:mariastorozhilova@gmail.com)



Alexey S. 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](mailto:lukin@graphics.cs.msu.ru)



Dmitry V. Yurin (Ph.D.) is a senior researcher at laboratory of Mathematical Methods of Image Processing, Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University, Russia. His contact email is [yurin\\_d@inbox.ru](mailto:yurin_d@inbox.ru)



Valentin E. Sinitsyn (MD) is a Head of Radiology, Federal Center of Medicine and Rehabilitation, Moscow. His contact email is [vsini@mail.ru](mailto:vsini@mail.ru)

# Border Extraction of Epidermises, Derma and Subcutaneous Fat in High-Frequency Ultrasonography

Alexandr Semashko<sup>1</sup>, Artem Yatchenko<sup>1</sup>, Andrey Krylov<sup>1</sup>, Artur Bezugly<sup>2</sup>, Natalia Makhneva<sup>2</sup>, Nikolay Potekaev<sup>2</sup>

<sup>1</sup>Laboratory of Mathematical Methods of Image Processing, Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University, Moscow, Russia

<sup>2</sup> Moscow Scientific-Practical Center of Dermatology and Cosmetology, Moscow, Russia

## Abstract

Skin image segmentation semi-automatic algorithm for high-frequency ultrasound probes with 75 MHz has been designed. The method includes a specific image filtering procedure followed by multiframe active contour algorithm to detect borders between the epidermis and the dermis and between the dermis and subcutaneous fat tissue.

**Keywords:** Skin, Epidermis, Dermis, Border, High-frequency ultrasound, Snakes.

## 1. INTRODUCTION

Last decade showed a rapid increase in the number of new treatments in the field of dermatology and aesthetic medicine. However, the number of objective skin diagnostical evaluation methods is not too big.



Figure 1: High-frequency ultrasonography study process.

High-frequency ultrasound probes with 22, 30, 50, 75 and 100 MHz is one of noninvasive evaluation methods of the skin [1] (see an example of the clinical procedure in Figure 1). The resolution of this method lies in the range 80-16 microns, which allows visualization of the skin and its layers. In addition to visualization, high-frequency ultrasound imaging allows to obtain quantitative data on the size of the observed objects, such as the thickness of the epidermis and dermis, as well as their acoustic density, which increases the reliability of clinical data. High-frequency ultrasound examination of the skin provides valuable diagnostic information when monitoring skin tumors, in assessing the development and regression of primary and secondary elements of skin lesions; also it has been successfully used for monitoring of the effects of different treatments – topical therapy, pharmacological therapy, physiotherapy and surgery. In the aesthetic medicine high-frequency ultrasound is used before and after fillers injections.

Recently, ultrasonography of the skin became quite widespread. It is used both in research and in clinical practice [2–4]. Application of high-frequency ultrasound under the existing time limits for the

examination of the patient makes very actual the problem of automating the processing of ultrasound images. Particularly it is necessary to automate calculation of dermis thickness, as well as to detect borders between the epidermis and the dermis and between the dermis and subcutaneous fat tissue.

The structure of the article is as follows: The statement of the high-frequency ultrasound probes images analysis is given in Section 2; The necessary filtering procedure for the high-frequency ultrasound images is described in Section 3; Section 4 presents the use of the proposed multiframe active contour algorithm for experimental images segmentation; An example of the obtained processing results by our method is given in Section 5.

## 2. CLINICAL INVESTIGATION

We studied the skin of the forearm in healthy caucasian women 35-37 years old. The study was carried out on a high-frequency ultrasound skin imaging system DUB (tpm production, Germany).

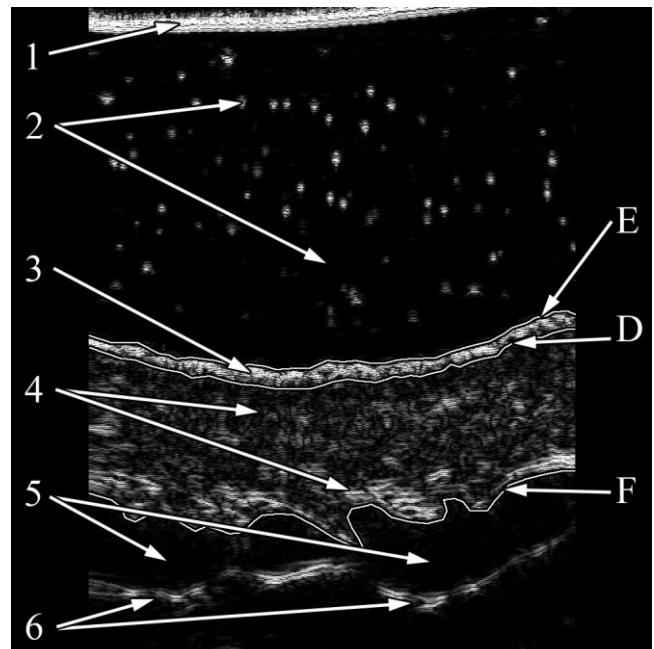


Figure 2: Skin structure visualization. 1) is a membrane delimited water chamber of the ultrasonic transducer, 2) is a contact gel between the membrane and the patient's skin, 3) is the epidermis (bright hyperechoic band), 4) is the derma (heterogeneous area, light and dark pixels), 5) is hypoechoic subcutaneous fat, 6) is superficial fascia. Lines: E is surface between epidermis and gel, D is surface between dermis and epidermis, F is surface between hypoechoic subcutaneous fat and dermis.

We used 75 MHz linear transducer for ultrasound with scanning depth 4 mm and the width of the scanned area of 12.9 mm/ Axial resolution was 21 microns. The epidermis, dermis and subcutaneous fat, were visualized (Figure 2) and the thickness of the epidermis and dermis were measured, and the level of echogenicity were determined.

### 3. SKIN IMAGE FILTERING

The input noisy data have a distinct anisotropic structure (see example of input data in Figure 3a). To suppress this artifact 1D-vertical Gaussian blurring and median filtering are used. Median filter preserve the brightness of image regions but small details (especially in dark areas) can be lost (Figure 3b). Gauss blurring with small radius (in our calculations we used value  $\sigma = 2$ ) preserve small details in image. However, the entire image becomes blurred and the overall brightness of the image can be lost (especially for the bright areas) (see Figure 3c). For image filtering a combination of median filter with 3x3 kernel and vertical Gauss blurring is used:

$$F(i_{x,y}) = \max\{ M(G(i_{x,y})), G(M(i_{x,y})) \},$$

where  $i$  is the processed image,  $M(i_{x,y})$  is the result of applying the median filter to the image  $i$  in  $\{x,y\}$ ,  $G(i_{x,y})$  is the result of the vertical Gauss blurring with  $\sigma = 2$  to the image  $i$  in  $\{x,y\}$ .

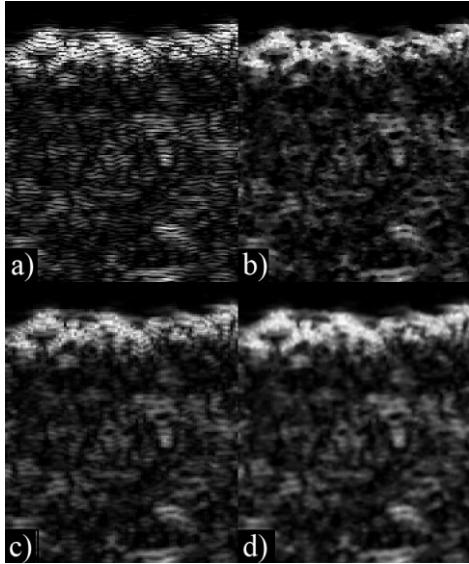


Figure 3: Results of filters application. a) input image, b) median filter with 3x3 kernel, c) vertical Gauss blurring with  $\sigma = 2$ , d) the proposed filtering method.

### 4. CONTOURS

Our goal is to measure derma thickness in a sequence of ultrasound images. One image from the sequence is to be marked up manually by the doctor. The necessary markup consists of three lines roughly representing boundaries of epidermis and derma (see lines E, D, F in Figure 2).

We track these lines through the sequence using active contours. Each line has its own set of external forces that control the active contour movement. See sections 4.1-4.3 for details. One force common to all the lines is a modified gradient vector flow force described in section 4.2. All forces act along the contour normal

(this means that the GVF force, for example, is projected onto the normal when applied at a control point). In our model magnitude of all forces is less than or equal to 1.

#### 4.1 Capturing outer boundaries of skin tissues

The outer boundaries of skin tissues separate the dense internal region (mostly white) from exterior which is transparent for the ultrasound and appears dark on the image. The gradient vector flow force will, of course, generally push the contours towards these boundaries, but other edges may be strong enough locally in some regions. We introduce a local adjusting force that allows the contours to overcome these local attractors.

For each segment connecting a pair of control points two adjacent regions are examined (see Figure 4a). We calculate average intensity in these regions. If the region on the outer side has high average intensity, we need to push the contour towards the exterior. Likewise, if the region on the inner side is too dark, the contour must be pulled back. Magnitudes of the two forces that provide such a behavior depend linearly (but are bounded) on the average intensities (see Figure 4b). The local adjusting force is the sum of these forces.

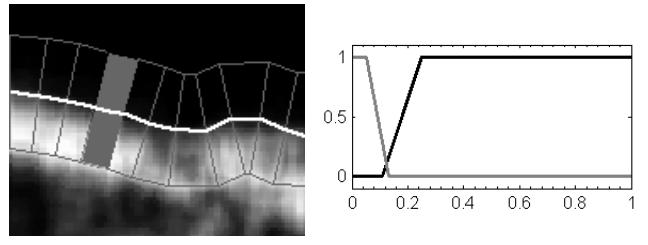


Figure 4. Calculation of boundary adjusting forces.

a) Patch analysis example. White line is the contour being adjusted, the patches corresponding to the 4<sup>th</sup> segment are shown in solid gray. Other patches are outlined in gray.

b) Plot of forces magnitude versus average patch intensity (assumed pixel values are in range [0, 1]). Gray line describes the force that pull the contour down (depends on the patch below the contour), the black line describes the opposite force.

#### 4.2 Separating the epidermis

The boundary D between derma and epidermis is sometimes hard to detect. First, there are regions without any significant change in image intensity at the boundary. And there may be regions where the epidermis is extremely thin. In the first case, the segmentation cannot be reliably done using only an edge-attracting external force like GVF [5]. If the second case takes place (or, ultimately, both of them), the segmentation will likely fail, detecting the outer boundary, E, instead of D.

In order to detect the line D correctly in all cases, we employ a modified gradient vector flow force, and add a thickness constraint to the active contours that track lines E and D.

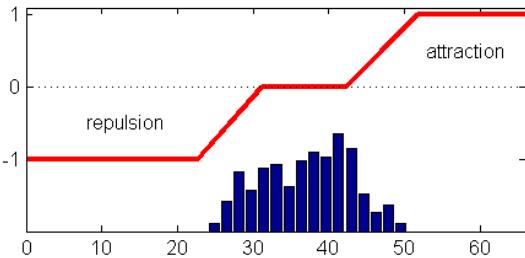
The GVF force may tend to attract contour tracking the line D to the outer boundary if the epidermis is rather thin. We propose a modification of GVF that suppresses gradient flow through regions with high intensity. This effectively stops attracting the line D contour towards line E, as there is bright epidermis region between them. From the mathematical point of view, this is achieved by adding an additional coefficient that weakens the smoothness constraint imposed onto the force components, so the modified force field minimizes the following functional:

$$\varepsilon = \iint \left( \mu \cdot W \cdot (\nabla p)^2 + (\nabla q)^2 + |\nabla M|^2 \cdot |\mathbf{F} - \nabla M|^2 \right) dx dy,$$

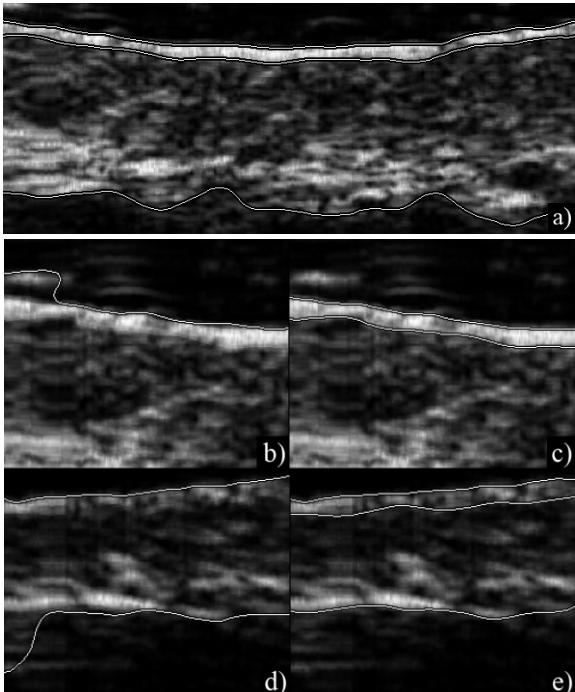
where  $F = (p, q)^\top$  is the force field,  $M$  is the edge map,  $\mu$  is a regularization coefficient, and  $W$  is the smoothness weakening coefficient. It is defined as follows ( $I$  denotes image intensity):

$$W = \begin{cases} 1 - 2I, & 1 - 2I \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Prior to evaluating active contours, we measure the epidermis thickness from the manual markup. We add a force that maintains the distance between active contours tracking lines E and D so that it corresponds to the epidermis thickness from manual markup. For each control point on one of the contours, we detect the nearest point on the other contour. The distance between these points dictates whether a repulsing or attracting force should be applied to the control point. An example that shows how the force depends on this distance is shown in Figure 5.



**Figure 5.** Plot of the distance maintaining force against the distance between contours in pixels. The histogram of distances in the markup is given below the plot.



**Figure 6:** a) example of the epidermis and the dermis segmentation result, b),d) are fragments of the result of single-frame snakes algorithm segmentation, c),e) are fragments of the result of our multi-frame snakes segmentation.

#### 4.3 Cross-frame links

Since we process a sequence of frames, we expect the resulting contours to move smoothly in time (i.e. between frames). To achieve this, we evaluate all the contours for all frames

simultaneously, with an additional force that links a contour tracking a line, say line E, in one frame, to the contours that track the same line E in the previous and the next frames. For each control point we detect the nearest point of the corresponding contour from a neighboring frame. Magnitude of the force that pulls the control point towards its nearest point on a neighboring frame is proportional to the distance between points. The forces acting on a single control point are summed together and, like the other forces, bounded to be not greater than 1 in magnitude.

## 5. RESULTS

Typical example of an image series segmentation result is shown in Figure 6a (only one frame is presented). Obtained derma thickness value for this series is  $306.25\mu\text{m}$  with standard deviation  $27.5\mu\text{m}$ . A comparison of multiframe and single-frame active contour results for border detection is presented in Figure 6b–6e. Single-frame snakes often produce inaccurate results (see Figure 6 b, d for examples). The proposed method is much more robust due to the use of constraints that limit contour changes in neighboring frames and control epidermis thickness (see Figure 6 c, e).

The work was supported by federal target program “Scientific and scientific-pedagogical personnel of innovative Russia in 2009–2013”.

## 6. REFERENCES

- [1] Raju B.I., Swindells K.J., Gonzalez S., Srinivasan M.A., *Quantitative ultrasonic methods for characterization of skin lesions in vivo* // *Ultrasound in Medicine & Biology*, v. 29, no. 6, 2003, pp. 825–838.
- [2] Konig K., Speicher M., Kohler M.J. et al., *Clinical application of multiphoton tomography in combination with high-frequency ultrasound for evaluation of skin diseases* // *J. Biophotonics*, v. 3, no. 12, 2010, pp. 759–773.
- [3] Moghimi S., Miran Baygi M.H., Torkaman G., Mahloojifar A., *Quantitative assessment of pressure sore generation and healing through numerical analysis of high-frequency ultrasound images* // *J. Rehabil. Res. & Devel.*, v. 47, no. 2, 2010, pp. 99–108.
- [4] Dańczak-Pazdrowska A., Polańska A., Silny W., et al., *Seemingly healthy skin in atop dermatitis: observations with the use of high-frequency ultrasonography, preliminary study* // *Skin Research and Technology*, v. 18, no. 2, 2012, pp. 162–167.
- [5] Xu C., Prince J., *Snakes, Shapes, and Gradient Vector Flow* // *IEEE Trans. on Image Proc.*, v. 7, no. 3, 1998, pp. 359–369.

## About the author

- Alexander Semashko is a Ph.D. student of the Faculty of Computational Mathematics and Cybernetics of Lomonosov Moscow State University (CMC MSU). His contact email is [alex.semashko@gmail.com](mailto:alex.semashko@gmail.com)
- Artem Yatchenko is a Ph.D. student of CMC MSU. His contact email is [artem@yatchenko.com.ua](mailto:artem@yatchenko.com.ua)
- Andrey Krylov is a professor, head of Laboratory of Mathematical Methods of Image Processing, CMC MSU. His contact email is [kryl@cs.msu.ru](mailto:kryl@cs.msu.ru)
- Artur Bezugly is a senior scientist at Moscow Scientific-Practical Center of Dermatology and Cosmetology (MSPCDC). His contact email is [arturbezugly@gmail.com](mailto:arturbezugly@gmail.com)
- Natalia Makhneva is a professor, science vice director of MSPCDC,
- Nikolay Poteaev is a professor, director of MSPCDC.

# Deblurring in fundus images

Alexandra A. Chernomorets, Andrey V. Nasonov\*

Department of Computational Mathematics and Cybernetics

Lomonosov Moscow State University, Moscow, Russia

{chernomorets, nasonov}@cs.msu.ru

## Abstract

New method for blur detection in retinal images has been suggested. The procedure includes original algorithm of edge width estimation. A method of image deblurring with ringing control is proposed.

**Keywords:** *Retinal images, blur detection, edge width, ringing control.*

## 1. INTRODUCTION

Automatic quality analysis of medical images has become an important research direction. However, the automatic blur detection in fundus images has not received enough attention. Fundus images are acquired at different sites, using different cameras operated by people with varying levels of experience. This results in a variety of images with different quality, and in some of them pathologies cannot be clearly detected or are artificially introduced. These low quality images should be specially examined by an ophthalmologist and reacquired if needed.

Current approaches for retinal image quality determination are based on global image intensity histogram analysis [1] or on the analysis of the global edge histogram combined with localized image intensity histograms [2]. In both these approaches a small set of excellent quality images was used to construct a mean histogram. The difference between the mean histogram and a histogram of a given image then indicated the image quality. Nevertheless these methods cannot be used as general fundus image quality classifiers because images of poor quality that match the method's characteristics of acceptable quality can be easily presented.

In [3] a correlation between image blurring and visibility of the vessels was pointed out. By running a vessel segmentation algorithm and measuring the area of detected vessels over the entire image, the authors estimate if the image is good enough to be used for screening. The main drawback is that the classification between good and poor quality needs a thresholding.

In [4] a quality class classification of the images is proposed. Vasculature is analyzed in a circular area around the macula. The presence of small vessels in this area is used as an indicator of image quality. The presented results are good, but the method requires a good quality segmentation of the vasculature and other major anatomical structures to find the region of interest around the fovea.

This paper presents a novel approach for blur detection in fundus images based on estimation of blur level of automatically segmented vasculature. The method does not require the segmentation of optic disk or macula, and only a rough segmentation of blood vessels is necessary. This allows us to perform the procedure of vasculature segmentation using downsampled images and significantly decrease computational time. The method can be also applied to noisy images or images with non-uniform luminosity.

## 2. EDGE WIDTH ESTIMATION

### 2.1 Edge model

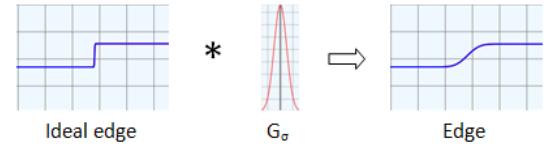
We consider general edge model as a result of convolution of the ideal step edge of unit height and a Gaussian kernel with some dispersion  $\sigma$ . Such assumption gives us a unique correspondence between the edge and a numeric value, i.e. the dispersion  $\sigma$  of the Gaussian kernel, which we take as the value of the edge width.

We define the ideal step edge function of unit height as:

$$H(x) = \begin{cases} 1, & x \geq 0, \\ 0, & x < 0. \end{cases} \quad (1)$$

The edge  $E_\sigma(x)$  (see fig. 1) is defined as

$$E_\sigma(x) = [H * G_\sigma](x). \quad (2)$$



**Figure 1:** Edge model

Note that the function  $E_\sigma(x)$  holds

$$E_\sigma(x) = E_{\sigma'}\left(\frac{\sigma'}{\sigma}x\right). \quad (3)$$

### 2.2 Edge width

For the edge width estimation we use the unsharp masking approach.

Let  $U_{\sigma,\alpha}[E_{\sigma_0}](x)$  be the result of unsharp masking applied to the edge  $E_{\sigma_0}(x)$ :

$$\begin{aligned} U_{\sigma,\alpha}[E_{\sigma_0}](x) &= \\ &= (1 + \alpha)E_{\sigma_0}(x) - \alpha E_{\sigma_0} * G_\sigma = \\ &= (1 + \alpha)E_{\sigma_0}(x) - \alpha E_{\sqrt{\sigma_0^2 + \sigma^2}}(x). \end{aligned} \quad (4)$$

Using (3) and supposing  $\sigma = \sigma_0 = \sigma_1$ , (4) holds

$$\begin{aligned} U_{\sigma_1,\alpha}[E_{\sigma_1}](x) &= \\ &= (1 + \alpha)E_{\sigma_1}(x) - \alpha E_{\sqrt{2}\sigma_1}(x) = \\ &= (1 + \alpha)E_{\sigma_1}(x) - \alpha E_{\sqrt{2}\sigma_2}\left(\frac{\sqrt{2}\sigma_2}{\sqrt{2}\sigma_1}x\right) = \\ &= U_{\sigma_2,\alpha}[E_{\sigma_2}]\left(\frac{\sigma_2}{\sigma_1}x\right). \end{aligned} \quad (5)$$

\*The work was supported by Federal Targeted Programme "R&D in Priority Fields of the S&T Complex of Russia 2007–2013".

The unsharp masking approach (4), due to (5), holds that the intensity values of corresponding extrema of  $U_{\sigma,\alpha}[E_\sigma](x)$  at  $x_{max_\sigma}$  and  $x_{min_\sigma}$  are the same for all  $\sigma > 0$  with fixed  $\alpha$ . Thus, taking into account the monotonicity of  $U_{\sigma,\alpha}[E_\sigma](x)$  as a function of  $\sigma$  due to (5) and the properties of Gaussian functions, fixing the value of  $\alpha$  and  $U_E = \max_x U_{\sigma_E,\alpha}[E_\sigma](x)$  for some  $\sigma_E$

$$\begin{aligned} U_{\sigma,\alpha}[E_{\sigma_0}](x_{max_\sigma}) &\geq U_E, \sigma < \sigma_0 \\ U_{\sigma,\alpha}[E_{\sigma_0}](x_{max_\sigma}) &\leq U_E, \sigma > \sigma_0. \end{aligned} \quad (6)$$

### 2.3 The edge width estimation algorithm

The final edge width estimation algorithm looks as follows:

1. Given values:  $\alpha$ ,  $U_E$ , 1-dimensional edge profile  $E_{\sigma_0}(x)$ .

2. for  $\sigma = \sigma_{min}$  to  $\sigma_{max} : \sigma_{step}$

    compute  $U_{\sigma,\alpha}[E_{\sigma_0}](x)$ ,  
     find local maxima  $x_{max_\sigma}$  of  $U_{\sigma,\alpha}[E_{\sigma_0}](x)$ ,  
     if  $U_{\sigma,\alpha}[E_{\sigma_0}](x_{max_\sigma}) \geq U_E$   
         result =  $\sigma$ ,  
         stop cycle.

3. Output: result.

### 2.4 Automatization of the parameters of the edge width estimation algorithm

The algorithm requires the values of  $\alpha$ ,  $\sigma_{min}$ ,  $\sigma_{max}$ ,  $\sigma_{step}$  and  $U_E$  that is dependent on  $\alpha$ . The research shows that the best value for  $\alpha$  is 4 (see Table 1).

The value of  $U_E$  for  $\alpha = 4$  is equal to 1.24.

The value for  $\sigma_{min}$  is fixed and is equal to 0.5. This value is the smallest possible value for the edge blur due to the digitization of the image. The value of  $\sigma_{step}$  is also fixed to 0.1 as this is an acceptable accuracy for the task.

The value of  $\sigma_{max}$  depends on the size  $l$  of the edge profile sample. In order to reach the best approximation of the edge width estimation the size of the Gaussian filter with dispersion  $\sigma$  is taken as  $8\sigma$ . Thus the value of  $\sigma_{max}$  is set to  $l/8$ .

$\sigma \downarrow \alpha \rightarrow$	2	2.5	3	3.5	4	4.5	5
0.5	0.6	0.6	0.6	0.6	0.6	0.6	0.6
1.5	1.5	1.5	1.5	1.5	1.5	1.6	1.6
2.5	2.5	2.6	2.6	2.6	2.5	2.5	2.5
3.5	3.6	3.5	3.5	3.5	3.5	3.5	3.6
4.5	4.5	4.5	4.5	4.6	4.5	4.5	4.5
5.5	5.5	5.6	5.5	5.5	5.5	5.5	5.6
6.5	6.5	6.5	6.5	6.6	6.5	6.5	6.5
7.5	7.6	7.5	7.5	7.5	7.5	7.5	7.5
8.5	8.5	8.5	8.5	8.5	8.5	8.5	8.5
9.5	9.5	9.5	9.5	9.5	9.5	9.5	9.5
10.5	10.5	10.5	10.5	10.5	10.5	10.5	10.5
11.5	11.5	11.5	11.5	11.5	11.5	11.5	11.5
12.5	12.5	12.5	12.5	12.5	12.5	12.5	12.5
13.5	13.5	13.5	13.5	13.5	13.5	13.5	13.5
14.5	14.5	14.5	14.5	14.5	14.5	14.5	14.5
15.5	15.5	15.5	15.5	15.5	15.5	15.5	15.5
16.5	16.5	16.5	16.5	16.5	16.5	16.5	16.5
17.5	17.5	17.5	17.5	17.5	17.5	17.5	17.5
18.5	18.5	18.5	18.5	18.5	18.5	18.5	18.5
19.5	19.5	19.5	19.5	19.5	19.5	19.5	19.5

**Table 1:** The results of the edge width estimation algorithm for different values of parameter  $\alpha$

### 3. APPLICATION OF THE EDGE WIDTH ESTIMATION ALGORITHM TO BLUR DETECTION IN FUNDUS IMAGES

In order to compute the blur value of the image we use the following algorithm:

- Extract vessels,
- Preliminary blur the image in order to suppress noise,
- Extract vessel edge profiles,
- Compute the image blur value taking into account the edge heights.

#### 3.1 Vessel detection

The algorithm for vessel segmentation was previously described in [5]. We perform vessel detection on downsampled fundus images in order to fusten the algorithm.

#### 3.2 Preliminary blurring

Due to the way of acquisition the target images have some noise. In the case of sharp images the noise almost does not affect the results of the edge width estimation algorithm, but in the general case some denoising is necessary for the correct use of the algorithm.

The algorithm for determination of the Gaussian kernel dispersion for preliminary blurring is as follows:

- Find a homogenous area  $A(x, y)$  on the image that is considered constant without noise.
- Compute the mean  $M$  on the found area.
- Compute the percentage  $P$  of maximum absolute deviation  $D$  from mean in the area:

$$P = D / (\max A(x, y) - \min A(x, y))$$

- Set the dispersion  $\sigma_B$  to value

$$\sigma_B = (P + 1) * P * 100.$$

#### 3.3 Extraction of the vessel profiles

We analyze the profiles of the boundaries of the widest vessels. In order to obtain these vessels' segments, we use the following algorithm:

- Skeletonize the vessel mask,
- Take only segments with the length greater than double maximum vessel width,
- Sort the segments list in the descending order of the segment width,
- Take 50 of the widest segments,
- For every segment find its center and direction. The edge profile is taken as the cross-section of the segment.

#### 3.4 Blur value estimation

In order to obtain an adequate result of two fundus images comparison we should take into account not only the average edge widths, but also amplitudes of the edges. The median of the weighted edge widths is taken as the blur value of the image. The weights are taken as inverse amplitude of the edge  $10/A$ .

So the algorithm for computing the blur value of the image is as follows:

- Compute the edge amplitude  $A_i$  and normalize every edge profile so that the values belong to the interval from 0 to 1,

2. For every normalized profile compute the edge width  $\sigma$  according to the algorithm described in section 2.3, and then compute the value  $W_i$ , taking into account preliminary blur:

$$W_i = \sqrt{\sigma^2 - \sigma_B^2},$$

3. Scale the obtained values by the inverse original edges amplitudes and obtain the value  $10 \frac{W_i}{A_i}$  that characterizes the edge,

4. Compute the median value of the weighted edge widths of the image.

### 3.5 Results

Most of the publicly available databases contain images only of a good quality. As an example the proposed algorithm was tested on retinal images from the DRIVE database [6]. The average blur value for the images from this database was found as 0.29.

The method was also tested on real images from ophthalmological practice of different quality.

The results for images of different quality are shown in fig. 2, 3.

The results show that the images with blur value less than 1 are good enough for pathology analysis.

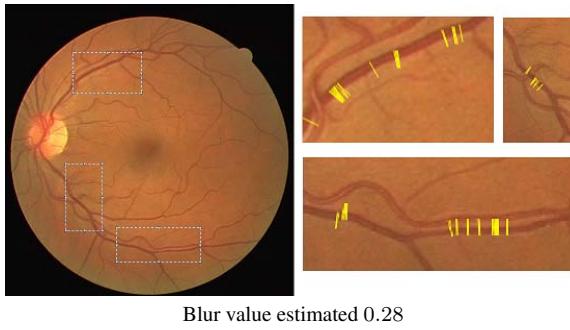


Figure 2: Blur value estimation for DRIVE image 01\_test.

## 4. DEBLURRING OF RETINAL IMAGES

After blur value estimation, we perform deblurring of retinal images. Unsharp masking is used for image sharpening. One of the problems of the image sharpening is introducing a ringing effect when the sharpening level is too high. We choose the parameter  $\sigma$  of the unsharp masking in accordance to estimated edge width and image ringing level.

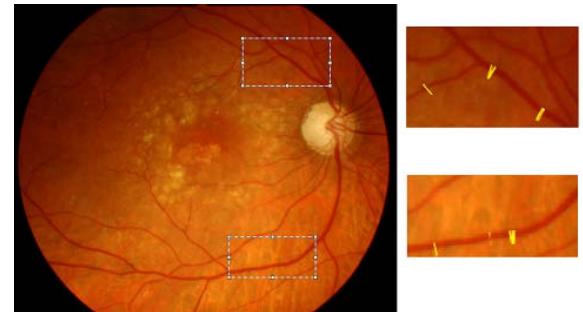
### 4.1 Ringing level estimation

We use the integral approach from [7] based on Total Variation analysis to estimate the ringing effect after applying the unsharp mask.

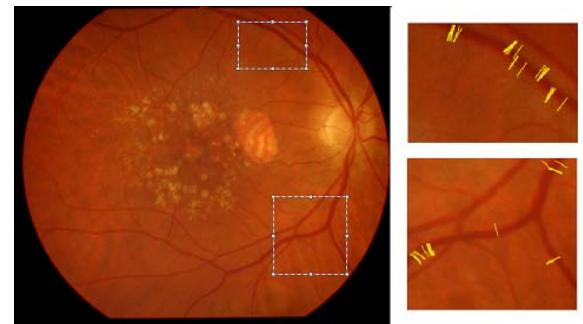
Ringing oscillations are located tangential to the edge producing them. The ratio between the average modulus of directional partial derivatives with normal and tangential directions  $nV/tV$  to the nearest edge is a good indicator of the presence of the ringing effect.

The ringing effect is the best noticeable near basic edges — sharp edges distant from other edges [8]. An example of basic edges detection is shown in Fig. 4.

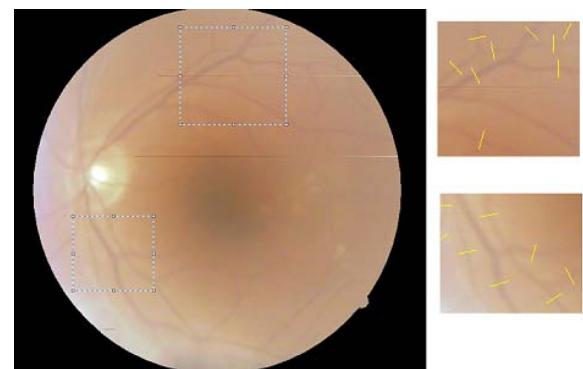
We calculate  $nV$  and  $tV$  in the subset of basic edge area with the distance to the nearest basic edge between  $d$  and  $ad$  where  $a$  controls the size of ringing effect. For unsharp mask deblurring, we use  $d = \sigma$  and  $a = 2$ . Then we calculate the ratio  $R_V = nV/tV$ . We use this value as the indicator of presence of ringing effect. If the ratio is close to 1, we assume that there is no ringing effect. If this value is significantly greater than 1, we assume



Blur value estimated 0.63



Blur value estimated 0.82



Blur value estimated 1.09

Figure 3: Blur value estimation for real fundus images.

that there is ringing effect. To reduce the influence of noise, we prefilter the image by Gauss filter with the radius  $\sigma/2$ .

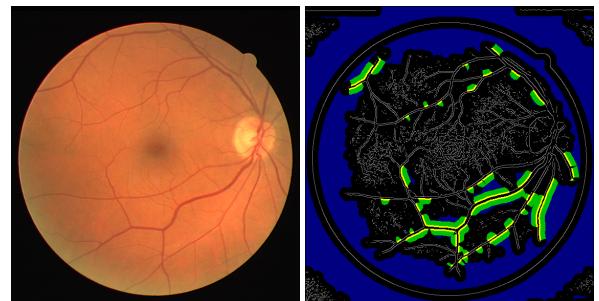


Figure 4: The results of basic edges detection. Left image: the original image. Right image: the result of basic edges detection. Green area is the area where we analyze the ringing effect.

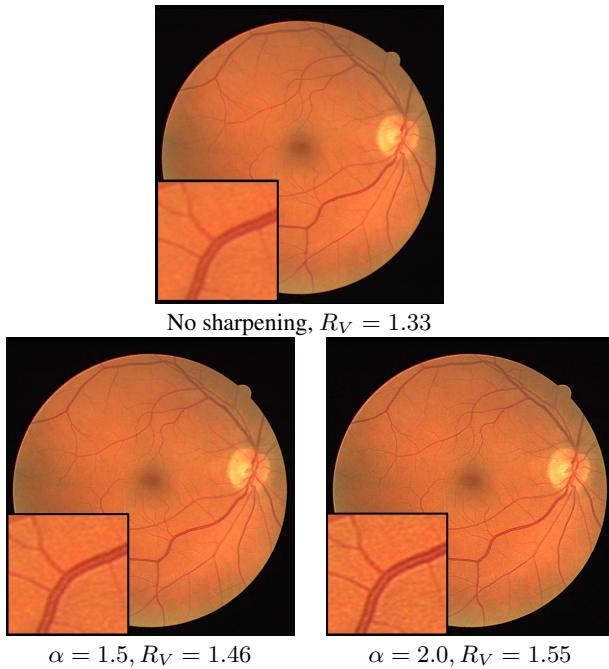
## 4.2 Unsharp masking

We use unsharp mask filter

$$I_\alpha = I * G_\sigma + \alpha(I - I * G_\sigma)$$

with the value  $\sigma$  equals to estimated edge width.

We use the maximum parameter  $\alpha$ , which results in ringing value  $R_V$  less than 1.5. Example results are shown in Fig. 5.



**Figure 5:** The results of retinal images sharpening using unsharp mask with ringing control, estimated  $\sigma = 1.23$ .

## 5. CONCLUSION

The paper presents a solution to the problem of the detection of blur in retinal images.

The application of the proposed algorithm to the classification of real retinal images showed good results. The algorithm can be used during the acquisition of fundus images and for the preliminary control of input data for retinal image CAD systems.

## 6. REFERENCES

- [1] S.Lee and Y.Wang, "Automatic retinal image quality assessment and enhancement," in *Proceedings of SPIE Image Processing*, 1999, vol. 3661, pp. 1581–1590.
- [2] M.Lalonde, L.Gagnon, and M.C.Boucher, "Automatic visual quality assessment in optical fundus images," in *Proceedings of Vision Interface*, 2001, pp. 259–264.
- [3] D.B.Usher, M.Himaga, and M.J.Dumskyj, "Automated assessment of digital fundus image quality using detected vessel area," in *Proceedings of Medical Image Understanding and Analysis*, 2003, pp. 81–84.
- [4] A.D.Fleming, S.Philip, K.A.Goatman, J.A.Olson, and P.F.Sharp, "Automated assessment of diabetic retinal image quality based on clarity and field definition," *Investigative Ophthalmology & Visual Science*, vol. 47, no. 3, pp. 1120–1125, 2006.
- [5] A.A.Chernomorets, A.S.Krylov, A.V.Nasonov, A.S.Semashko, V.V.Sergeev, V.S.Akopyan, A.S.Rodin,

and N.S.Semenova, "Automated processing of retinal images," in *21-th International Conference on Computer Graphics GraphiCon'2011*, September 2011, pp. 78–81.

- [6] A.Can, H.Shen, J.N.Turner, H.L.Tanenbaum, and B.Roysam, "Rapid automated tracing and feature extraction from retinal fundus images using direct exploratory algorithms," in *IEEE Transactions on Information Technology in Biomedicine*, 1999, vol. 3, pp. 125–138.
- [7] A. M. Yatchenko, A. S. Krylov, and A. V. Nasonov, "MRI medical image ringing detection and suppression," in *8th Open German-Russian Workshop "Pattern Recognition and Image Understanding" (OGRW-8-2011)*, Lobachevsky State University of Nizhny Novgorod, November 2011, pp. 259–262.
- [8] A. V. Nasonov and A. S. Krylov, "Finding areas of typical artifacts of image enhancement methods," *Pattern Recognition and Image Analysis*, vol. 21, no. 2, pp. 316–318, 2011.

## ABOUT THE AUTHOR

Alexandra Chernomorets is a PhD student of the Faculty of Computational Mathematics and Cybernetics (CMC), Lomonosov Moscow State University (MSU).

Andrey Nasonov is a member of scientific staff of CMC MSU.

# A Hash based Approach for Secure Keyless Steganography in Lossless RGB Images

Ankit Chaudhary<sup>1</sup>, J. Vasavada<sup>1</sup>, J.L. Raheja<sup>2</sup>, Sandeep Kumar<sup>3</sup>, Manmohan Sharma<sup>3</sup>

<sup>1</sup>BITS Pilani, RJ India  
ankitc.bitspilani@gmail.com, h2010451@bits-pilani.ac.in

<sup>2</sup>CEERI/CSIR Pilani, RJ India  
jagdish@ceeri.ernet.in

<sup>3</sup>Pilani, RJ India  
{kumarsandeep.bits, manmohan.manu}@gmail.com

## Abstract

This paper proposes an improved steganography approach for hiding text messages in lossless RGB images. The objective of this work is to increase the security level and to improve the storage capacity with compression techniques. The security level is increased by randomly distributing the text message over the entire image instead of clustering within specific image portions. Storage capacity is increased by utilizing all the color channels for storing information and providing the source text message compression. The degradation of the images can be minimized by changing only one least significant bit per color channel for hiding the message, incurring a very little change in the original image. Using steganography alone with simple LSB has a potential problem that the secret message is easily detectable from the histogram analysis method. To improve the security as well as the image embedding capacity indirectly, a compression based scheme is introduced. Various tests have been done to check the storage capacity and message distribution. These testes show the superiority of the proposed approach with respect to other existing approaches.

**Keywords-** Keyless Steganography, Lossless RGB Images, Hash Based Embedding, Information Hiding

## 1. INTRODUCTION

Steganography is a technique used to transmit a secret message from a sender to a receiver in a way such that a potential intruder does not suspect the existence of the message. Generally this is done by embedding the secret message within another digital medium such as text, image, audio or video. The terms cryptography and steganography are often used synonymously although they are essentially distinct. In cryptography, a plain message is encrypted into cipher text and might look like a meaningless jumble of characters whereas in case of steganography, the plain message is hidden inside a medium that looks quite normal and does not provide any reason for suspecting the existence

of a hidden message. Using steganography, we need to take care about the resource utilization, the space requirements and the level of security. In addition, the quality of the cover object i.e. digital medium is important after using it for steganography.

Electronic steganography approaches use digital processing techniques for hiding and detecting embedded information. In case of image steganography, the secret message is transmitted embedded within a digital image called a *cover* image. Once the message is embedded within the image, it is referred to as a *stego* image. In a keyless steganography approach the sender includes only the information and does not include any cryptography algorithm. Therefore, the reliability and the security are totally dependent on the efficiency of the steganography algorithm itself. In this paper a keyless steganography algorithm applicable for lossless image formats like BMP, PNG or TIF, is proposed. Attempts are made to improve the storage capacity while incurring minimal quality degradation of the image. Security is enhanced by distributing the message throughout the image. The organization of the paper is as follows: section 2 discusses the related works, section 3 describes the proposed algorithm, section 4 provides details about the experimentations and results while section 5 discusses the overall conclusion and future scopes. This work is extension of our previous work [3].

## 2. RELATED WORK

The most widely used technique to hide data is the usage of the LSB. Although there are several disadvantages to this approach, it is relatively one of the easiest to implement. This method uses bits of each pixel in the image, it is necessary to use a lossless compression format otherwise the hidden information will get lost in the transformations of a lossy compression algorithm. Least Significant Bit replacement embeds fixed-length secret bits into the least significant bits of pixels by directly replacing the Least Significant Bits of each byte of the cover image with the secret message bits. Some stego analysis methods will identify the pixel difference in

the host image very easily [6]. For a 24 bit image, 3 bits can be stored in each pixel. To the human eye, the resulting stego image looks almost identical to the cover image. A random LSB insertion method in which secret message is spread out among the image data in a seemingly random manner [11]. This is an efficient approach but changes to the MSB bits can degrade the image quality substantially.

One of the best keyless steganography approaches is the Pixel Indicator Technique (PIT) algorithm proposed by A. Gutub [4] where a color channel is used as a pixel indicator and the other two channels are used as for message embedding. The main drawback is that one of the channels cannot be used to store the actual message. To increase the capacity, usage of 2, 3 or 4 are proposed in [7]. A heuristic based approach for information embedding in the form of multimedia objects or text using steganography is proposed in [1]. In [10], the least significant bit (LSB) of each pixel is modified sequentially. In [2] the authors have proposed a steganographic technique by mixing with it cryptography to increase the security layer. There are other approaches available using Discrete Cosine Transformation (DCT) and Wavelet Transform (WT) [5].

### 3. PROPOSED APPROACH

The proposed approach is described in different sections as follows.

#### 3.1 Motivation and Overview

The main motivation of the proposed work is to overcome the shortcomings of the techniques suggested in [9], which is an improvement of Pixel Indicator Technique (PIT) proposed in [4]. The process can be made efficient by considering only a single function i.e. randomization, instead of calculating K1 and K2. There is another shortcoming with the approach suggested in [9] that K1 and K2 calculations are dependent upon image size and message length. Our approach also takes care of distribution of the message over the cover image, as an improvement of not storing the message bits within the cover image in contiguous pixels only in the upper portion of the image [4]. Additionally the proposed work also attempts to increase the security level by introducing a text message compression at the first level, which indirectly increases the embedding capacity of the image.

#### 3.2 Indicator Values

The first step towards the random distribution of the message in image is using indicator values. In the current work, we use MSB bits of Red, Green and Blue channel as pixel indicator values instead of utilizing an entire channel as in [4], which is suggested in [9]. The MSBs indicate in what sequence the message is hidden using the LSBs. In addition to this, this scheme is applied after applying compression to the original message. Therefore it would be make it extremely difficult to break, even after suspicion of the message within an image. In this scheme the MSB remains unchanged when an LSB of a byte is utilized for storing a message. The sequences of LSB bits containing the message are indicated in Table 1. For example if the MSB code of channels is 001, then the message hiding LSB sequence is RGB, but if the MSB is 100, then the

message hiding sequence becomes BGR. This scheme enables us to fully utilize all the LSBs of every channel of the cover image to store the hidden message and hence improve its capacity. Moreover the varying indicator values introduce a security aspect as it becomes increasingly difficult to decode the message even if its presence is suspected. We are not changing the scheme suggested for Indicator values in [9].

**Table 1.** Indicator Values

MSB bits of Red, Green and Blue channel sequentially	Sequence of channel's LSB bits where the message bits needs to be Hidden
000	Red, Green and Blue (RGB)
001	Red, Green and Blue (RGB)
010	Red, Blue and Green (RBG)
011	Blue, Red and Green (BRG)
100	Blue, Green and Red (BGR)
101	Green, Red and Blue (GRB)
110	Green, Blue and Red (GBR)
111	Green, Blue and Red (GBR)

#### 3.3 Message Distribution

Instead of storing the message only in the upper portion of the image in contiguous pixels, as in [4], there is a technique proposed to distribute the message over the entire image in [4]. There is a key-number generated which indicate the gap value between two pixels containing the message to be hidden. This key-number K2 is computed based on *MessageLength* and image dimension i.e. *ImageWidth* and *ImageHeight* of the cover image. K2 is an integer which indicates the gap between pixels which contain the actual hidden information i.e. the information is inserted after every K2 pixels. The potential drawback with this is: the message distribution within the image is fixed in order. i.e. information is inserted after every K2 no. of pixels.

The proposed approach is based on randomization and hashing with respect to the MSBs of the channels, to skip R numbers of bytes. R is a generated random numbers based on the seed value S given to a random numbers algorithm. To illustrate the procedure, a XOR operation between the LSB bits of the cover image and the stego image indicate the pixels which have changed. The message is seen clustered towards the upper portion of the stego image, while for the scheme indicated in [9], the results for the proposed approach where the message is seen to be distributed with skipping K2 bytes after storing information once [9]. The result for the proposed approach, where the message is distributed randomly (with respect to MSBs) after storing information once, is better than the above two approaches.

#### 3.4 Image Quality

While using steganography, image quality is one of the most important issues because, degradation in image quality suggests directly towards ‘something embedded in the image’. The quality of an image is degraded by changing the bits of an image to store information. As the number of LSB bits used to store the message increases, the quality of the stego image

correspondingly degrades more. To limit this degradation the proposed approach uses stego-1 bit LSB [7] which implies that we are storing 1 message bit/channel, however we can use maximum of four Least Significant bits of each channel from an image. This improves the quality compared to the algorithm proposed in [4] where two LSB bits are used for the purpose. The other improvement in the image quality is due to randomization, i.e. not all the image bytes are used for the embedding purpose.

### 3.5 Message Encoding Process

The message encoding process is summarized below:

Input: Cover Image, Secret Message File.

Step1: Take as input the cover image, message/file to be hidden.

Step2: Compress the original secret file. (Output: .zip file)

Step3: Store the compressed file within the cover image (using the indicator values specified in Table 1).

Step4: Calculate the random number R and skip R no. of bytes.

Step5: Repeat steps 3 and 4 till the message embedding is not over.

Output: Stego Image.

### 3.6 Message Retrieving Process

The message retrieving process is summarized below:

Input: Stego Image.

Step1: Take the stego image as an input.

Step2: Retrieve the message from the image by generating random number R and skipping R bytes every time, using the table indicator values.

Step3: The output would be a compressed text file (.zip). By uncompressing the file, we would get the original text message file.

Output: Secret Message.

## 4. EXPERIMENTAL RESULTS

We resort to a number of statistical measures, to verify the superiority of this approach. Different sizes of messages inserted to different cover images. Table 2 shows the analysis with respect to embedding capacities of different algorithms. Figure 1 shows the original image and the Figure 2 is after inserting the secret text. The sample image contains more than 1 KB data. It is clear that, the change in image after embedding the data is not visible in the image, by bared eyes. Here, 'Y' in Table 2 represents '*embedding is possible*' and 'N' represents '*embedding is not possible*'. It can be seen from the Table 2 that the embedding capacity of the suggested approach is comparable to the approach suggested in [9]. In our approach, although randomization tries to distribute message more, compression in the first step would compress the original message, increasing the embedding capacity of the image indirectly. One important thing to note here is, the cover image size should not be affected after embedding.

However, padding based approach increases the image size by message size after embedding.



Figure 1: Original Image



Figure 2: Stego Image

## 5. CONCLUSIONS

In this paper we have proposed an improved steganography approach for hiding text messages within lossless RGB images. The objective of this work is to increase the security level and improve the storage capacity while incurring minimal quality degradation. As future improvements we want to compress the message and then store the data to increase the capacity further. We also want to insert any message both text and image at the same time (might be in PDF format or WORD format). We would also like to improve our algorithm by keeping in mind the robustness issue.

## 6. ACKNOWLEDGEMENT

This research is being carried out at Central Electronics Engineering Research Institute (CEERI), Pilani. Authors would like to thank Director, CEERI for his active encouragement and support.

**Table 2.** Embedding Capacity

Algo \ MsgSize	10 B	1 KB	100 KB	1 MB	3 MB	5 MB	7 MB
Simple LSB	Y	Y	Y	Y	Y	Y	N
Padding Based Approach <sup>#</sup>	Y	Y	Y	Y	Y	Y	Y
PIT	Y	Y	Y	N	N	N	N
Ref-10 (max = 2.2MB)	Y	Y	Y	Y	N	N	N
Suggested Approach (max = 3.8MB)	Y	Y	Y	Y	Y	N	N

\*considering imagesize = 800\*600

## 7. REFERENCES

- [1] Bandyopadhyay, S.K., Bhattacharyya, D., Das, P., Mukherjee, S., Ganguly, D., A secure scheme for image transformation, In Proceedings of 9th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, Phuket, Thailand, Aug. 06-08, 2008, pp. 490-493.
- [2] Bhattacharyya, D., Das, P., Bandyopadhyay, S.K., Kim, T., Text steganography: a novel approach. International Journal of Advanced Science and Technology, Vol. 3., Feb 2009, pp. 79-86.
- [3] Chaudhary, A. and et al., "An Algorithmic Approach to Intrusion Detection", Proceedings of 4th IEEE International Conference on Advanced Computing and Communication Technologies (ICACCT 2010), Panipat, India, 30th Oct, 2010, pp. 742-748.
- [4] Gutub A. A., Pixel indicator technique for RGB image steganography, Journal of Emerging Technologies in Web Intelligence, Vol. 2, Issue 1, Feb 2010, pp. 56-64.
- [5] Khare, A., Kunari, M., Khare P., Efficient Algorithm For Digital Image Steganography, Journal of Information Science, Knowledge and Research in Computer Science and Application, Oct 2010, pp. 1-5.
- [6] Johnson, N.F., and Jajodia, S., Exploring steganography: Seeing the unseen, IEEE Computer Magazine, Vol. 31, Issue 2, Feb 1998, pp. 26-34.
- [7] Neeta, D., Snehal, K., Jacobs, D., Implementation of LSB steganography and its evaluation for various bits, In Proceedings of 1<sup>st</sup> International Conference on Digital Information Management, Bangalore, India, Dec 6, 2006, pp. 173-178.
- [8] Raheja, J.L., Manasa M.B.L., Chaudhary, A., Raheja, S., ABHIVYAKTI: Hand Gesture Recognition using Orientation Histogram in different light conditions, Proceedings of the 5th Indian International Conference on Artificial Intelligence (IICAI-11), Tumkur, India, 14-16 Dec, 2011, pp. 1687-1698.
- [9] Roy, S. and Parekh, R., A Secure Keyless Image Steganography Approach for Lossless RGB Images, In Proceedings of ACM ICCCS, Rourkela, Odhisha, India, Feb 11-12, 2011, pp. 573-576.
- [10] Singh, M., Singh, S.B. and Singh L.S.S., Hiding encrypted message in the features of images, International Journal of Computer Science and Network Security, Vol. 7, Issue 4, April 2007, pp. 302-307.
- [11] Sutaone, M.S. and Khandare, M.V., Image based steganography using LSB insertion technique. In Proceedings of International Conference on Wireless, Mobile and Multimedia Networks, Mumbai, India, Jan 11-12, 2008, pp. 146-151.

# Cost-effective multiframe demosaicking based on bilateral filtering

K. Gorokhovskiy

Aerocosmos Scientific Center for Aerospace Monitoring, Moscow, Russia  
gorokhovskiy@gmail.com

## Abstract

A simple but effective multiframe demosaicking method is proposed. Its primary goal is to replace more expensive mechanical motion compensation systems. Therefore, it is designed to be easily implemented in hardware for consumer devices. The described multiframe demosaicking algorithm is suitable for mass production devices such as mobile phones or digital cameras. It is compared to a multiframe noise reduction of similar complexity. The comparison is based on computer-based simulation of a camera being (unintentionally) shaken by a human operator. The following error measurements were taken: Mean Squared Error (MSE), Peak Signal to Noise Ratio (PSNR) and Normalized Color Difference (NCD) errors measurements were taken.

**Keywords:** demosaicking, noise reduction, bilateral filtering, multiframe processing

## 1. INTRODUCTION

Digital cameras and so-called camera-phones are now widely spread. Although, image quality from them has improved drastically in recent years, still, it is not comparable to human vision capabilities especially under low light conditions. One of the main problems is sensor noise.

Current cameras perform at their physical limits and photon noise is dominant. On a physical level, this type of noise can be reduced by increasing the number of photons detected by each cell on a sensor. Usually, the solutions are: increasing the optical efficiency of a lens system or increasing exposure times.

Improving optical efficiency is expensive as the complexity of the lens grows disproportionately relative to its quality, not to mention that the camera often needs to be small in its application (e.g. a camera-phone). Longer exposures, in turn, produce motion blur which can be compensated mechanically or electronically.

Taking into account the generally falling cost of electronic components electronic motion compensation becomes more and more attractive in terms of quality per unit cost.

Both frame-based demosaicking and multiframe noise reduction are well developed areas in their own right. The combination of these two methods only recently received a proper attention [1]. However, there is still a lack of simple but effective methods which can be implemented in existing devices.

In this paper a new method of multiframe demosaicking is proposed and compared to combination of simple frame-based demosaicking and multi-frame noise reduction. The comparison is carried out using computer-based simulation of a series of shots which are shifted and rotated, then mosaicked. After that, Poisson noise is added to simulate the photon noise of a photo sensor. This algorithm is an extension of the work described in [2].

## 2. ALGORITHM REQUIREMENTS

The original prerequisite for the proposed algorithm is that it can be put into a camera image processing pipeline without a significant increase in cost. This leads to the following requirements:

- (a) The method should not consume too much memory (not more than 4 image frames) even if the technique involves merging many more frames.
- (b) It should be real-time or, in other words, the user should receive the result just after the shot (no time-consuming post processing is allowed).

It is clear from the requirements that algorithm should be stream based and data should be accumulated and processed “on the fly”.

Having many images of the same scene it is possible to use a wide variety of super-resolution algorithms. However, the requirements for memory and computational power restrict application of those methods inside digital still cameras and mobile phones.

## 3. TEMPORAL BILATERAL DEMOSAICKING

Having the classical bilateral filtering equation for image  $f(\mathbf{x})$  as in [3]:

$$h(\mathbf{x}) = k_d^{-1} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(\xi) c(\xi - \mathbf{x}) s(f(\xi) - f(\mathbf{x})) d\xi \quad (1)$$

and normalization coefficient  $k_d$ :

$$k_d = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} c(\xi - \mathbf{x}) s(f(\xi) - f(\mathbf{x})) d\xi \quad (2)$$

where  $c(\xi - \mathbf{x})$  is the *geometric* closeness between the neighborhood centre  $\mathbf{x}$  and a nearby point  $\xi$ ,  $s(f(\xi) - f(\mathbf{x}))$  measures the *photometric* similarity between the pixel at the neighborhood centre  $\mathbf{x}$  and that of a nearby point  $\xi$ .

For the task of multiframe demosaicking it is possible to introduce an additional pixel weight coefficient responsible for trustworthiness of a pixel  $w(\xi)$ . In the situation when several frames are merged together some pixels may contain more pixels of a particular color. The greater the number of values in a given pixel position, the better the accuracy. Thus equations (1) and (2) become:

$$h(\mathbf{x}) = k_d^{-1} \sum_{t=1}^T \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \hat{f}(\xi) w(\xi) c(\xi - \mathbf{x}) s(f(\xi) - f(\mathbf{x})) d\xi \quad (3)$$

and

$$k_d = \sum_{t=1}^T \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} w(\xi) c(\xi - \mathbf{x}) s(f(\xi) - f(\mathbf{x})) d\xi \quad (4)$$

where  $\hat{f}(\xi)$  are the resultant mean values of the colors in the given locations,  $t$  is the index of the frame in a sequence,  $T$  is total number of frames available for fusion. By converting the equation in discrete space we have

$$h(x, y) = k_d^{-1} \sum_{t=1}^T \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} \hat{f}(i, j) w(i, j) c(i - x, j - y) s(f(i, j) - f(x, y)) \quad (5)$$

and

$$k_d = \sum_{t=1}^T \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} w(i, j) c(i - x, j - y) s(f(i, j) - f(x, y)) \quad (6)$$

Usually the *photometric* similarity function is defined as:

$$s(f(i, j) - f(x, y)) = \exp \left\{ - \left( \frac{\delta(f(i, j) - f(x, y))}{2\sigma} \right)^2 \right\} \quad (7)$$

where

$$\delta(f(i, j) - f(x, y)) = \|f(i, j) - f(x, y)\| \quad (8)$$

Usually,  $\|f(i, j) - f(x, y)\|$  is selected as a suitable measure of distance between the two color values. In the scalar case, this may be simply the absolute difference of the pixel values or, since the photon noise increases with intensity, an intensity dependant version of it. It is possible to determine what the *photometric* similarity function should be in case of Poisson noise.

In order to simplify the formulae and minimize the amount of computations the image samples can be converted to a space where the noise has a normal distribution (i.e. Gaussian). Also, it is assumed for simplicity that image under consideration has only one channel. The equations below can be easily extended for multi-channel images. The probability function relating to the difference of two image samples which are close in space is:

$$p(a - b) = \frac{1}{\sqrt{2\pi(\sigma_a^2 + \sigma_b^2)}} \exp \left\{ - \frac{((a - b) - (\mu_a - \mu_b))^2}{2(\sigma_a^2 + \sigma_b^2)} \right\} \quad (9)$$

where  $\sigma^2$  is variance of a random variable and  $\mu$  is the mean or its expected value. The probability that these two samples have the same value is

$$p(a - b)_{\mu_a = \mu_b} = \frac{1}{\sqrt{2\pi(\sigma_a^2 + \sigma_b^2)}} \exp \left\{ - \frac{(a - b)^2}{2(\sigma_a^2 + \sigma_b^2)} \right\} \quad (10)$$

It can also be demonstrated that the standard deviation of both variables are the same

$$\sigma_a = \sigma_b = \sigma \quad (11)$$

Thus

$$p = \frac{1}{\sqrt{4\pi\sigma^2}} \exp \left\{ - \frac{(a - b)^2}{4\sigma^2} \right\} \quad (12)$$

which is similar to equation (7). For the experiment the following conversion to the space with a normal distribution was used

$$f_n(x, y) = \sqrt{f_p(x, y)} \quad (13)$$

where  $f_p$  are the samples with a Poisson distribution and  $f_n$  are the samples with a normal distribution.

In order to reduce the amount of computations used a simplified spatial penalty function was used:

$$c(i, j) = \begin{cases} 1, & -N \leq i \leq N, -M \leq j \leq M \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

The multiframe bilateral demosaicking algorithm was applied on a  $5 \times 5$  neighborhood with  $N = M = 2$ .

The described equations can be optimized in order to minimize memory usage on a computational device. Equations (5) and (6) allow accumulation of the intermediate results frame by frame.

By storing numerator and denominator  $k_d$  of equation (5) as two separate frames in memory it is possible accumulate data frame by frame. When no more frames are expected in a sequence the final result of computation can be achieved by dividing accumulated numerator

$$\sum_{t=1}^T \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} \hat{f}(i, j) w(i, j) c(i - x, j - y) s(f(i, j) - f(x, y))$$

by denominator  $k_d$  from equation (6).

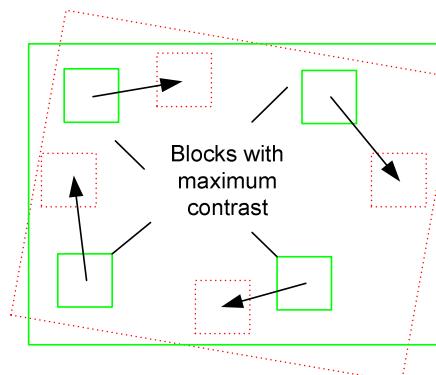
#### 4. GLOBAL MOTION ESTIMATION ON MOSAICKED IMAGES

As the global motion estimation was not an essential part of the comparison and the simplest exhaustive search was taken as a basis.

It is important to stress out that there was no novelty introduced for global motion estimation in this paper. Motion estimation was not a goal of this research. Any state of the art research results on global motion estimation could be used here. Therefore the comparison of accuracy of used global motion estimation (ME) algorithm was outside of the scope of this paper.

The same global motion estimation coefficients were used for both compared methods and therefore the *relative* accuracy of demosaicking methods should be unaffected by the accuracy of the global motion estimation. However, the algorithm for motion estimation is explained below for the reproducibility of the results.

It was assumed that global motion of the frame can be described as an affine transform with a relatively small number of coefficients so that for small area of image it can simply be defined as a shift in two dimensions (see Figure 1).



**Figure 1:** Global motion estimation using block matching. The proposed model assumes that if rotation is small ( $0^\circ - 2^\circ$ ) it can be neglected for motion blocks (32x32 pixels). Only shifts are taken into account.

Equation (15) and condition (16) define this:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (15)$$

where

$$|a_{21}| \ll 1, |a_{12}| \ll 1 \quad (16)$$

A limited number of blocks with maximal contrast were selected. The target is to find shifts in these blocks and to calculate the global motion using linear regression [4] or robust fitting. Figure 1 illustrates this.

The exhaustive search block matching algorithm was adapted from [5]. It was modified to introduce a penalty term for large motion vectors. In cases where two vectors exist with an equal cost the shortest will be selected.

The precision of motion estimation can be optimized further with a priori knowledge that the transform contains only rotation and shift. This condition can be described as follows:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha & a_{13} \\ -\sin \alpha & \cos \alpha & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (17)$$

As the rotation is small, the  $\cos \alpha$  component can be replaced by 1. Let  $c$  be:

$$c = \sin \alpha \quad (18)$$

Hence:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & c & a_{13} \\ -c & 1 & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (19)$$

The number of transform coefficients is thereby reduced from 6 to 3 making linear regression methods more effective.

The difference between classical motion estimation techniques and the proposed approach is that it is required to work with mosaicked images. The mosaicked images can be demosaicked before motion estimation but this is not the most precise or most computationally efficient way. The proposed method uses mosaicked (RAW) images for motion estimation. It will be shown that it is possible to obtain pixel precise motion estimate vectors using mosaicked data.

The basic operation in motion estimation is a measure of similarity between two regions of images. In our case the mask of existence of the given color in a given position is available, which simplifies the task. Let us assume that the penalty is the absolute difference between the two colors at a given pixel. Then it is possible to say that having no particular color in the mask should not add a penalty. This can be formalized as follows

$$p(x_1, y_1, x_2, y_2) = \sum_{c=1}^C |s_1(x_1, y_1, c)m_2(x_2, y_2, c) - s_2(x_2, y_2, c)m_1(x_1, y_1, c)| \quad (20)$$

where  $C$  is number of colors used in a sensor (usually 3 or 4),  $p(x_1, y_1, x_2, y_2)$  is the penalty term for pixels in locations  $x_1, y_1$  and  $x_2, y_2$  respectively. Also,  $s_1$  denotes a sample of the first

image,  $m_1$  is a mask value for the first image,  $s_2$  and  $m_2$  are defined similarly for the second image.

For some offsets there will be situations when all pixels between two block of image are unmatched according to color masks. In such conditions the penalty will be zero no matter what are the contents of the image.

In order to avoid such conditions the values of pixels are blurred spatially (separately by color planes) together with corresponding color masks.

The experiments carried out by the author show that Gaussian filtering with a small  $3 \times 3$  kernel of one of the images (including mask) improves the accuracy of the motion estimation for Bayer pattern.

Then, the penalty or difference measure  $P(x_1, y_1, x_2, y_2)$  for the block of pixels with dimensions  $N$  and  $M$  will be

$$P(x_1, y_1, x_2, y_2) = \sum_{i=1}^N \sum_{j=1}^M p(x_1 + i, y_1 + j, x_2 + i, y_2 + j) \quad (21)$$

Square blocks of pixels with dimensions  $M = N = 32$  were used for this research.

The resultant motion estimation algorithm used in this research can be described as following steps:

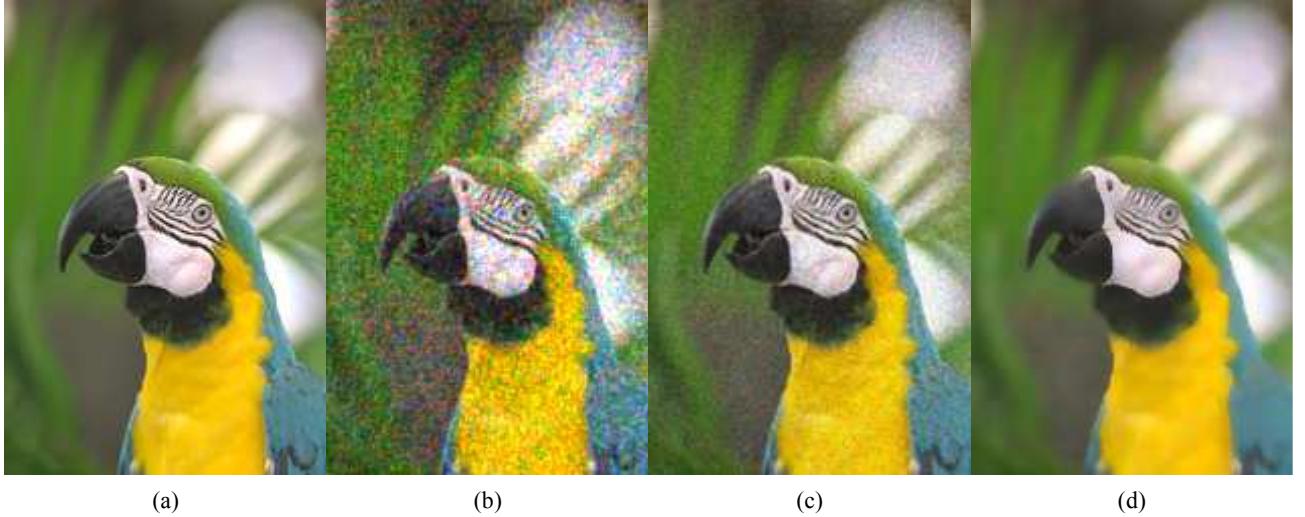
1. Split the first frame in the sequence into blocks 32 by 32
2. Select 50% of these blocks with maximal contrast
3. Let us assign the index  $k$  for each block having  $k \in [1 \dots K]$ , where  $K$  is total number of selected blocks
4. Store integer coordinates of the centers of the blocks as  $(x_k, y_k)$
5. For each block with coordinates of the  $(x_k, y_k)$  find the corresponding block on a given frame (different from first one) with integer coordinates  $(x'_k, y'_k)$  which minimizes  $P(x_k, y_k, x'_k, y'_k)$
6. Using multivariate linear regression algorithm [4] on initial and resultant sets of coordinates  $(x_k, y_k)$  and  $(x'_k, y'_k)$  find coefficients  $c$ ,  $a_{13}$ , and  $a_{23}$  for this frame

## 5. COMPARRISON OF THE RESULTS USING SIMULATION OF NOISE AND SHAKE

There methods were compared using raw images generated from "Kodak Image Set" [6]. Images were downsampled in order to reduce simulation time. The aim was to reproduce the image sequence from the real camera. Using real image sequence it is difficult to evaluate the accuracy of described methods as it is impossible to get the original reference image. By contrast when using a simulation the reference image is known in advance.

The following assumptions were used for the simulation process:

1. Overall exposure time of a sequence of shots is less than  $\frac{1}{4}$  second
2. There is only rotation and shift of the image taking place (no scaling or second order distortions)
3. Rotation is no more than 5 degrees between any two images in a sequence



**Figure 2:** From left to right: (a) original image, (b) one of the noisy images from the sequence of 9 images, (c) result of temporal variable number of gradients demosaicking using 9 images, (d) result of proposed method using 9 images.

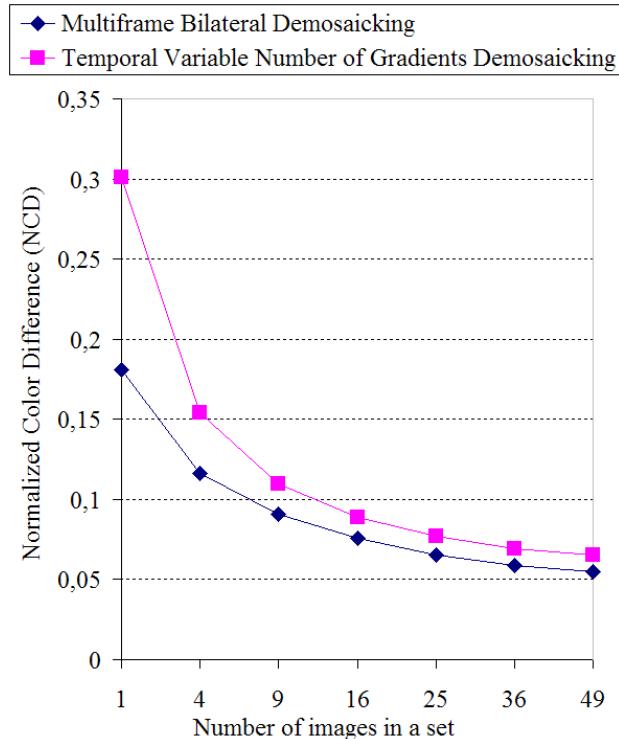
For error measurements MSE, PSNR, NCD formulae were used. MSE and PSNR are:

$$MSE = \frac{1}{W \cdot H \cdot C} \sum_{y=1}^W \sum_{x=1}^H \sum_{c=1}^3 \|O(x, y, c) - R(x, y, c)\|^2 \quad (22)$$

$$PSNR = 10 \cdot \log_{10} \left( \frac{1}{MSE} \right) \quad (23)$$

It is assumed that color values are within the range  $[0, 1]$ .

NCD stands for Normalized Color Difference. It was used previously to quantify the perceptual color difference and is defined as follows:



**Figure 3:** Dependence of normalized color difference from number of images available for fusing.

$$NCD = \frac{\sum_{x,y} \sqrt{(L_o - L_r)^2 + (u_o - u_r)^2 + (v_o - v_r)^2}}{\sum_{x,y} \sqrt{L_o^2 + u_o^2 + v_o^2}} \quad (24)$$

where  $L_r$ ,  $u_r$ ,  $v_r$  are lightness and chrominance components of the result image in CIELUV color space [7] at the pixel's location  $(x, y)$ ,  $L_o$ ,  $u_o$ ,  $v_o$  are the same color components that were in the original image.

The simulation was performed for different number of frames in a sequence varying from 1 to 49. The results are shown in Table I. The Multiframe Bilateral Demosaicking shows best results for MSE, PSNR and NCD measures.

Variable Number of Gradients Demosaicking [8] is one of the best non-iterative algorithms described in scientific publications. The operational neighborhood for both these methods is  $5 \times 5$  pixels. However, the computational efficiency of Multiframe Temporal Demosaicking is better than for Temporal Variable Number of Gradients Demosaicking.

Reference, noisy, and processed images are shown in Figure 2.

TABLE I: ACCURACY OF THE RESULTS OF TEMPORAL VARIABLE NUMBER OF GRADIENTS DEMOSAICKING COMPARED TO MULTIFRAME BILATERAL DEMOSAICKING ON THE KODAK IMAGE SET USING MSE, PSNR, AND NCD ERROR MEASUREMENTS

Number of images in a sequence	Temporal Variable Number of Gradients			Multiframe Bilateral Demosaicking		
	MSE	PSNR	NCD	MSE	PSNR	NCD
1	0.00966	20.22	0.301	0.00414	24.19	0.181
4	0.00262	25.90	0.154	0.00236	26.57	0.116
9	0.00151	28.38	0.110	0.00151	28.50	0.091
16	0.00111	29.80	0.089	0.00108	29.97	0.076
25	0.00091	30.76	0.077	0.00079	31.29	0.065
36	0.00083	31.27	0.069	0.00066	32.12	0.059
49	0.00077	31.67	0.065	0.00057	32.81	0.055

As can be seen from Table 2 and also from Figure 2, multiframe demosaicking is more effective than temporal noise reduction for a small number of frames. As the number of frames increases these two methods show comparable performance. On the other hand, multiframe temporal demosaicking is at least marginally better for each case in the simulation.

As can also be seen from the graph in Figure 3 both methods become more effective as the number of frames in the set increases.

It is important to note that the proposed method of Multiframe Bilateral Demosaicking is not based on the specific structure of a classic Bayer filter layout and can be easily adapted for alternative filter patterns.

## 6. CONCLUSION

The proposed method of multiframe demosaicking has shown an advantage over temporal noise reduction on sequences with number of frames varying from 1 to 49. It is also simple to implement in the hardware of modern digital camera or a mobile phone. To get better results with a small number of images in a set, multiframe demosaicking can be improved in an adaptive way such as a spatial filtering kernel for uniform surfaces and temporal filtering for edges. It should be mentioned that one of the disadvantages of the proposed method is absence of local motion estimation. The method can be significantly improved by detecting the areas of local motion between the frames. Thus, by matching the moved areas it would be possible to reduce the noise without introducing motion blur.

## 7. REFERENCES

- [1] S. Farsiu, M. Elad, and P. Milanfar, “Multiframe Demosaicking and Super-Resolution of Colour Images”, *IEEE Trans. Image Process.*, vol. 15, no. 1, pp. 141-159, Jan. 2006.
- [2] K. Gorokhovskiy; J.A. Flint, S. Datta, and N. Glushnev, “Cost Effective Multiframe Demosaicking for Noise Reduction,” *15th International Conference on Digital Signal Processing*, Cardiff, UK, pp. 407–410, July 2007.
- [3] C. Tomasi, R. Manduchi, “Bilateral Filtering for Gray and Colour Images,” *Proceedings of the 1998 IEEE International Conference on Computer Vision*, Bombay, India, 1998
- [4] P.W. Holland, R.E. Welsch, “Robust Regression Using Iteratively Reweighted Least-Squares,” *Communications in Statistics: Theory and Methods*, A6, pp. 813 827, 1977.
- [5] A. Barjatya, “Matching Algorithms for Motion Estimation,” 2004, <http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=8761>, last accessed December 2007.
- [6] 24 scanned images, “Eastman Kodak © photographic color image database”, 1993
- [7] M.D. Fairchild, “Color Appearance Models,” *Wiley-IS&T series in imaging science and technology*, Chichester, West Sussex, England: J. Wiley, pp. 194–201, 2005.
- [8] C E. Chang, C. Shiufun, and D. Pan, “Color filter array recovery using a threshold-based variable number of gradients,” *Proceedings of SPIE*, vol. 3650, pp. 36–43, 1999.

# Detection and correction of unnatural purple colors in images of refractive objects taken by digital still camera

Mikhail Matrosov, Alexey Ignatenko

Department of Computational Mathematics and Cybernetics  
Lomonosov Moscow State University, Moscow, Russia  
[{matrosov, ignatenko}@graphics.cs.msu.ru](mailto:{matrosov, ignatenko}@graphics.cs.msu.ru)

Sergey Sivovolenko

OctoNus Software Ltd.  
[sivovolenko@octonus.com](mailto:sivovolenko@octonus.com)

## Abstract

We discovered significant error in color in images produced by a digital still camera used to capture scenes with a special setup of illumination and an object. The object is small, transparent, color-less and has light-refractive properties. Illumination is produced by a number of LEDs with known spectra. Due to light dispersion in the object, vivid monochromatic colored flares can be seen and captured. Images from a digital still camera captured in the described setup occasionally exhibit bright purple (almost pink) colors, which do not correspond to any monochromatic color.

In this paper, we analyze the origins of this effect by examining different properties of the setup and analyzing RAW images. We propose a simple and efficient algorithm for correction of those unnatural purple colors by using only final JPEG image produced by the camera. In this algorithm, we develop a continuous transform which maps all unnatural colors to the natural ones in a perceptually uniform color space. Then, a 3D look-up table is constructed using this transform, which in turn is used for further correction of images captured by the camera.

**Keywords:** color management, color calibration, color correction, monochromatic colors, RAW-processing, perceptually uniform color spaces, light dispersion, digital still camera.

## 1 BACKGROUND

Consumer digital still cameras are very powerful tools for capturing real world images. They are broadly available, well-studied and intensively developed, that's why they are frequently used not only by photographers, but in many science and engineering applications. The latter require precise, repeatable and calibrated results.

Camera color calibration is not a trivial task since camera "sees" world in a model-specific color space. In order to be able to process and correctly display such an image one should be able to convert it to a certain conventional color space, like CIE XYZ [1]. The whole chain of color processing in digital cameras including this issue is briefly and clearly described by Adams et al. in [2]. They describe this conversion to be handled by a  $3 \times 3$  matrix converting camera-specific RGB response to universal XYZ values. The tricky part is that camera spectral sensitivities cannot usually be represented as linear combinations of CIE color matching functions forming XYZ values.

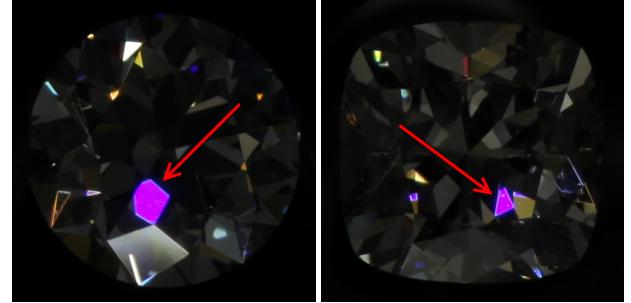
So this conversion matrix is usually designed to minimize averaged error for a specific set of colorants. Spaulding et al. [3] used Macbeth Color Checker [4] as a target set and an RMS error of CIELAB  $\Delta E_{ab}^*$  color-difference measure to find an optimal matrix:

$$\Delta E_{RMS}^* = \sqrt{\sum_{i=0}^N (\Delta E_i^*)^2},$$

where  $N$  is the number of color patches and

$$\Delta E_i^* = \sqrt{(L_{si}^* - L_{di}^*)^2 + (a_{si}^* - a_{di}^*)^2 + (b_{si}^* - b_{di}^*)^2},$$

where  $L_{si}^*$ ,  $a_{si}^*$  and  $b_{si}^*$  are the CIELAB scene color values for the



**Figure 1:** Images with regions of unnatural purple color for two different objects.

$i$ -th color path and  $L_{di}^*$ ,  $a_{di}^*$  and  $b_{di}^*$  are the CIELAB reproduced color values for the  $i$ -th color path.

Hong et al. [5] used a broader collection of colorants: an ANSI IT8.7/2 [6] chart on Kodak Ektacolor Professional Paper and the textile samples selected from The Professional Colour Communicator [7] using reactive dyes on cotton. They also performed a polynomial regression with least-squares fitting to minimize the color-reproduction error.

Thus, there are many available techniques to perform conversion of camera RGB response to XYZ values which can generally include some non-linear transformations or multidimensional look-up-tables. And we do not know precisely how a particular camera model handles this conversion since most of camera firmwares are proprietary and closed.

However, most of these techniques focus on reproduction of colors usually observable in natural scenes, but not all the physically available spectra. In specific engineering tasks, a certain spectra can be encountered, which a camera would not be able to handle properly. That is the case discussed in this paper.

In our task, we analyze optical properties of a transparent colorless object shaped as a polyhedron. Its refractive index is high enough to make it a dispersive media. Thereby, when illuminated by a white light, such an object appears to have colored faces. Since colors are induced by light dispersion and the object's faces are small enough, color spectra of a single face is nearly constant and virtually monochromatic. We used a consumer digital still camera to capture images of the described scene and discovered vivid saturated purple colors appearing under certain conditions. Examples of such images are shown in figure 1.

However, such vivid purple colors do not correspond to any monochromatic spectra and we weren't able to witness the same purple faces with naked eyes observation. Thus we have decided that we've encountered the mentioned above case of a camera being unable to properly represent captured color. We analyzed this issue and created simple a correction algorithm that works well with our setup. While the algorithm is very specific and aimed at our particular task, the conducted research is extensive and general enough.

## 2 INTRODUCTION

Detailed description of the used setup, including notes on illumination, objects' properties and camera model, is given in section 3. This section also contains information on how images were obtained and how a collection of analyzed images was formed.

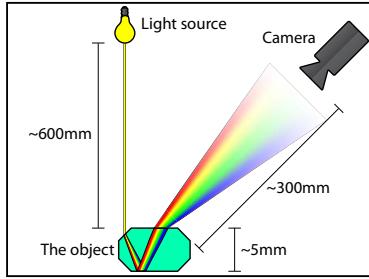
Extensive issue examination is done in section 4. We analyze the behavior of RGB response function in both output camera JPEG image and primary camera RAW image and show the evidence of error in color management process done by the camera in our specific case. We did not analyze the spectra of the faces which produced the unnatural purple colors since they are too small (about 100 microns in diameter) to be measured by any of the consumer spectrometers and appropriate measurement devices were unavailable for us. We do not consider unnatural colors to be induced by non-visible parts of spectra since we've done additional tests with UV and IR filters and the difference was barely noticeable.

Though we show that a consumer camera cannot properly handle discussed colors, we want to be able to use our setup for further research of described objects. So in section 5 we propose a simple correction algorithm which can be applied to JPEG camera output images to replace unnatural vivid purple colors with ones of a more bluish hue, which can be observed as a result of light dispersion. This algorithm continuously shifts specific hues in a perceptually uniform color space within boundaries estimated from captured collection of images. Examples of images corrected with the proposed algorithm are shown in section 6.

Conclusions and acknowledgements are given in sections 7 and 8 respectively.

## 3 SETUP AND PHOTOS

There are three essential components of the setup: illumination, an object and a camera. All of these are enclosed in a closed box with illumination mounted on the top, an object mounted at the bottom and a camera placed at the front and directed at the object. See figure 2 for schematic illustration<sup>1</sup>.



**Figure 2:** Schematic illustration of the setup used. Relative sizes of the objects and relative distances in the scene are not preserved for illustrative purposes.

Illumination consists of a number of bright LEDs with wide warm spectra. Camera white-balance was adjusted automatically prior to the shooting of any images. A sample paper patch with a neutral color was used for this purpose. There were about 50 LEDs, each of which is small enough and is supposed to approximate a point light source.

An object has a shape of a polyhedron with 50-70 faces and is 4-6 millimeters in diameter. It is made of a transparent colorless material with refractive index about 2.41, hence it introduces strong light dispersion and its faces appear to be colored when observed under appropriate illumination from a suitable point of view. The object is fixed on a motorized holder allowing rotation along two perpendicular axes situated in plane orthogonal to

<sup>1</sup>Camera icon designed by Go Squared Ltd.

camera's optical axis. Controlling this holder, one may adjust the position of the object in which a face with a color of interest will be observable by the camera.

The camera is mounted in front of the box and is pointed at the object. It is plugged into and is operated by a computer, so one can capture images of the object without touching the camera, which can lead to undesired vibrations of the box and break down the current dispersive pattern. In our tests we used a Canon EOS 5D Mark II digital still camera with a Canon EF 100mm f/2.8 Macro USM lens and a Kenko Teleplus PRO 300 DGX 1.4x AF teleconverter. However, as we will show later, explored effect poorly depends on a specific model of the digital camera.

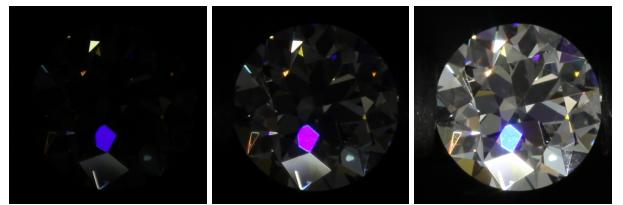
With the given setup, the linear size of an object on captured images came out to be 400-600 pixels. To obtain images of an object with purple faces (like the ones shown in figure 1) we simply rotated the holder slightly in an arbitrary direction and made a shot with the camera. We then studied acquired image for purple faces and optionally suggested a direction for further rotation. The effect is not quite rare, so usually it was enough to make 3-5 shots of the object to detect a new purple face and additional 2-3 shots to select an appropriate exposure.

Once a purple face was detected and an appropriate central exposure was selected, we made 11-15 shots of the same scene with different exposures using a  $\frac{2}{3}$  E.V. step. In other words, by making 15 shots we captured a number of images taken with exposures from  $4\frac{2}{3}$  to  $+4\frac{2}{3}$  E.V. relative to the central exposure.

All in all, we captured 11-15 exposures for each of the 3-5 positions of 5 objects resulting in 254 images.

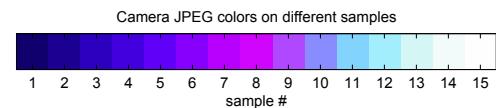
## 4 EXAMINATION

In this section we analyze separate sets with a number of images of the same scene taken with different exposures. Let us examine one particular set with 15 exposures; images in this set are numbered from 1 to 15 in order of the increasing exposure. Images #4, #8 and #12 are shown in figure 3. These are conventional output images of the camera stored as jpeg-files in the standard sRGB color space and we will refer to them as JPEG images (in contrast to RAW images, which will be discussed later).



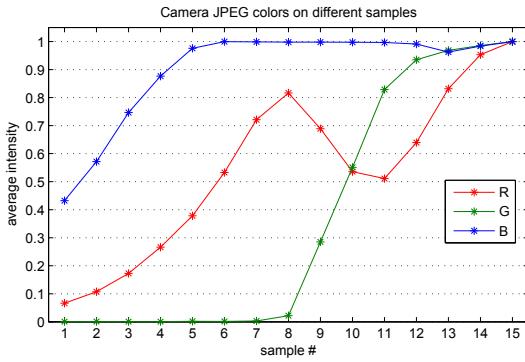
**Figure 3:** Three out of fifteen images in the examined set.

We begin analyzing the set by manually masking the region of the purple face on the images. Since all of the images are perfectly aligned, the same mask is used for all the images. We then average colors inside the mask for each of the images. In case there are several purple faces, we only mask the biggest one for simplicity and consistent averaging. Colors computed in such a way for all images in the set are shown in figure 4. We will refer to them as JPEG colors.



**Figure 4:** Averaged colors of a masked purple face for different images.

Since all these colors are from the same scene taken with different exposures, one can mention strange behavior of the color



**Figure 5:** Per-channel values of averaged colors of purple face for different JPEG images in the set.

transition with the increase of exposure. Blue color found in samples 1-4 changes to purple color in samples 5-9 and then to teal in samples 10-13, finally overexposing to white in samples 14-15. Only the last transition to white is expected, while others should not emerge in properly calibrated color-capturing device.

Intensity values of JPEG colors in the sRGB color space are plotted in figure 5. The first thing to mention is the non-monotonous R channel. Second is the different dynamic of growth of R and B channels in samples 1-5. Apparently, these facts indicate an unidentified error during capturing process.

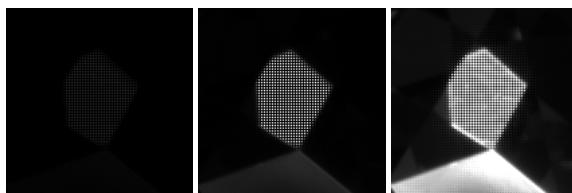
#### 4.1 Analyzing RAW

To dig into the origins of the problem, we analyze low-level responses of camera sensors stored in camera RAW images. Unfortunately, Canon EOS 5D Mark II provides RAW in CR2 format which is proprietary and not open. Moreover, none of the publicly available official Canon utilities like Digital Photo Professional or RAW Codec provide functionality to extract non-interpolated Bayer mosaic with primary sensor response. This is also true for a number of well-known RAW-processing utilities including Adobe Camera RAW, Adobe Lightroom, Apple Aperture and others.

Considering this fact, we used a free open-source utility ddraw (v9.12) with command line arguments “-d -4 -w” to obtain a non-interpolated Bayer mosaic from CR2 files. According to ddraw manual, the meaning of these arguments is as follows:

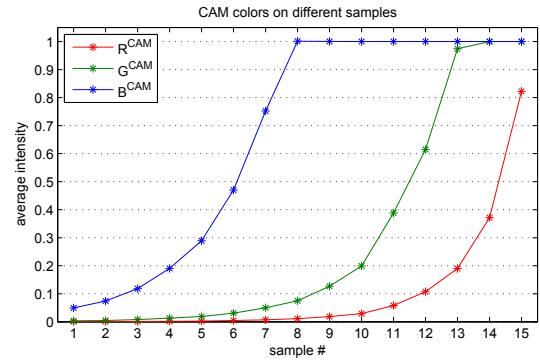
- d Document mode (no color, no interpolation)
- 4 Linear 16-bit
- w Use camera white balance, if possible

In this way we obtain linear responses from camera sensors in the form of pgm-files (see [8] for notes on PGM format) with 16-bit color depth and with white balance corrected using values estimated by the camera. Obtained colors are encoded in a camera-specific color-space, let us call it CAM. Likewise, we will refer to these images as CAM images. Cropped regions of a purple face from CAM images are shown in figure 6. They correspond to samples shown in figure 3.

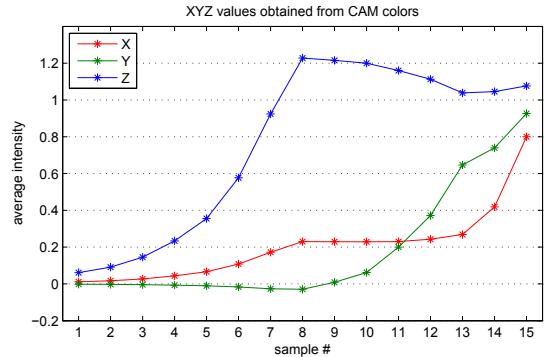


**Figure 6:** Linear Bayer mosaic extracted from RAW images.

Intensity values of CAM colors are plotted in figure 7. Averaging is done by splitting grayscale mosaic according to Bayer pattern



**Figure 7:** Per-channel values of averaged CAM colors.



**Figure 8:** XYZ values obtained with ddraw conversion matrix.

RGGB. The twin Bayer green components are treated equally. Analyzing this plot one can state the following observations:

- Values are linear within [0; 1] range.
- Values exceeding [0; 1] range are cropped.
- Values of channel B<sup>CAM</sup> are much greater than values of other channels.
- Channel B<sup>CAM</sup> saturates at image #8; channel G<sup>CAM</sup> saturates at image #14.

Matching these observations with mentioned color transitions in JPEG images we can conclude the following:

- Color is changing from blue to purple until saturation of B<sup>CAM</sup> channel.
- Once B<sup>CAM</sup> channel is saturated, color is changing towards teal until saturation of G<sup>CAM</sup> channel occurs.
- After that color is changing towards white.

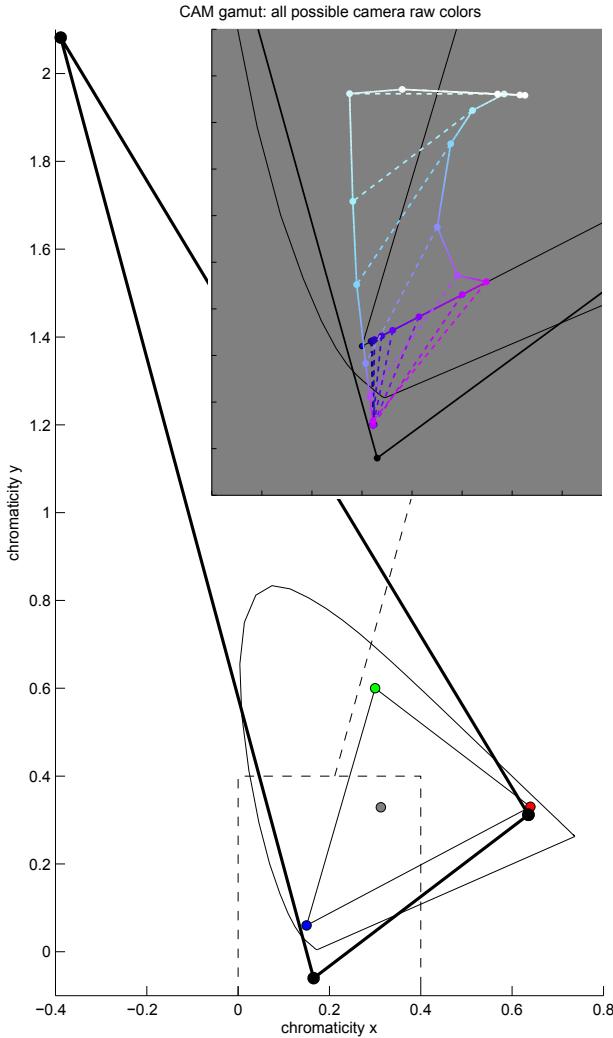
To be done with basic RAW analyzing, we make the conclusion that all unexpected color transitions correspond to the subsequent saturation of CAM color channels.

#### 4.2 Processing RAW

Since the camera converts a primary RAW image to a final JPEG image using proprietary closed firmware, we cannot precisely reproduce this process. However, we do know that it performs demosaicing to interpolate Bayer pattern and it uses a certain conversion matrix to obtain XYZ values from CAM values. Such a matrix is represented in ddraw source codes. Apparently, it was acquired by some reverse-engineering. We can use this matrix to convert CAM values to XYZ values, which are given in figure 8.

By analyzing this plot we conclude the following:

- XYZ values are linear while CAM values lay within [0; 1] range.
- When CAM values exceed [0; 1] range, XYZ values begin to change nearly unpredictably.



**Figure 9:** Gamut of CAM colors depicted on xyY chromaticity diagram against sRGB triangle. Zoomed part shows pairs of JPEG colors (inside sRGB gamut) and corresponding CAM colors (outside sRGB gamut).

- Y value is negative while CAM values lay within  $[0; 1]$  range.

The latter point is crucial. Y value of an XYZ triplet cannot be negative by definition. Thus we consider CAM  $\rightarrow$  XYZ conversion unreliable. The conversion is optimized to minimize errors in reproduction of normally observable colors and fails to properly handle values induced by purple faces. Applying this conversion to all possible CAM colors we derive its gamut, which is shown in figure 9. Zoomed part of the figure shows pairs of JPEG colors against CAM colors (background is grayed out to make luminous colors visible). Dashed lines connect corresponding elements of a pair. Solid lines connect colors from adjacent images for both CAM and JPEG series, thus creating two tracks.

It may be seen that camera's rendering intent tends to make sRGB colors from given XYZ values more purple while it should be probably mapped to blue. This intent is the final reason why we see those unnatural colors on JPEG images. However, rendering intent is not of a big concern since XYZ values themselves are obtained using an unreliable CAM  $\rightarrow$  XYZ conversion.

## 5 CORRECTION

By speaking of "correction of purple colors" we mean a scheme which allows to obtain conventional image of the scene without mentioned unnatural colors. Generally, it can include some kind

of RAW-processing, installation of color filters on a camera or certain manual adjustments. But we want this scheme to satisfy two requirements.

Firstly, a JPEG image produced by a camera has satisfactory quality, so we demand the corrected image to be as close to JPEG image as possible (except for regions of purple faces). Secondly, we want this scheme to be fast, meaning that the correction should only take a small fraction of time required to perform a shot.

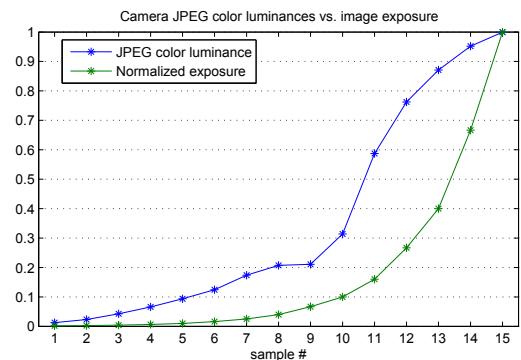
RAW-processing done by camera consists of a number of proprietary hardware-specific algorithms such as demosaicing, sharpening and postprocessing. It is unlikely that we can reproduce all of them in a simple and efficient enough RAW-processing line. So, in our scheme we propose to correct the resulting camera JPEG image itself.

Since we don't reliably know the XYZ values of the purple faces' colors, we can chose the corrected color somewhat arbitrarily. We suggest to simply shift the color hue of purple faces towards blue. The exact boundaries and directions of the shift are discussed further.

### 5.1 Proposed algorithm

Since in our task the only meaningful information in the camera JPEG image is the picture of an object, any purple color in any region of the image can be considered unnatural. Thus, we can apply a global correction based only on values of the particular pixel. Moreover, we can generate a 3D look-up-table to map values of input pixels to values of corrected pixels. This LUT transformation is very fast and since LUT itself can be pre-computed, we don't have to worry about the performance of the transformation algorithm.

Now let's deal with the properties of the color shift. Figure 10 illustrates luminance of JPEG colors plotted against normalized exposure for every image in the set. It may be seen that luminance is not linearly dependent on exposure. However, luminance plot is monotonous and its dynamics is similar to the dynamics of the exposure plot. So we will preserve the luminance value during the correction shift.



**Figure 10:** Luminance of JPEG colors against exposure.

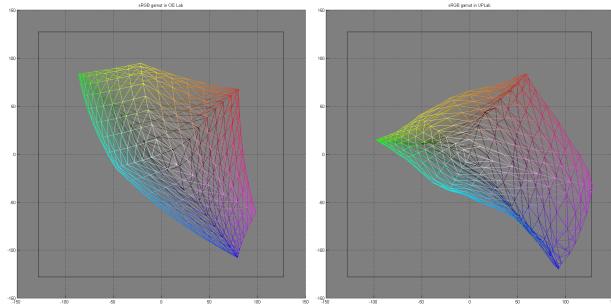
Since it is impossible (within the sRGB color space) to preserve both luminance and saturation while varying hue, we will have to compress the color saturation during the shift. Compressing saturation of the blue color is related to a known issue called the "blue turns purple" problem [9, The "Blue Turns Purple" Problem<sup>2</sup>]. We don't want to shift purple towards blue only for it to become purple again after the compression of the saturation. In order to avoid this issue, we use the Perceptually Uniform Lab color space [9, Uniform Perceptual Lab<sup>3</sup>] for all of our color manipulations. Moreover, the exact rules controlling the hue shift

<sup>2</sup><http://brucelindbloom.com/MunsellCalcHelp.html#BluePurple>

<sup>3</sup><http://brucelindbloom.com/UPLab.html>

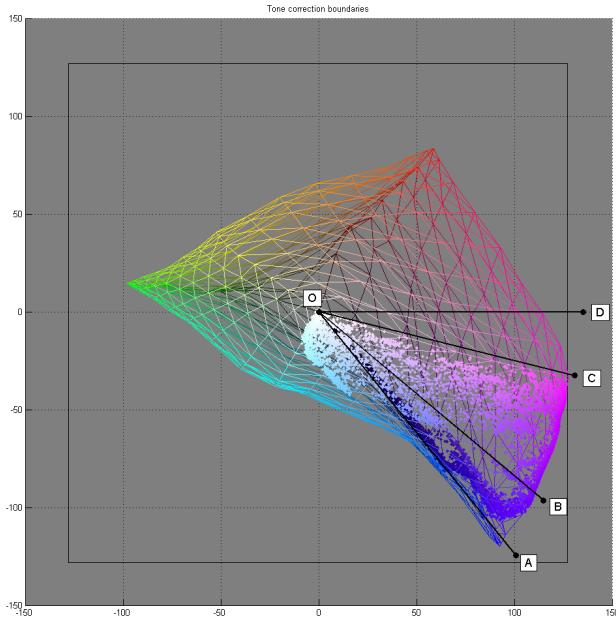
are much simpler in the UPLab color space than in the CIE Lab color space.

Rotation of the color radius-vector in UPLab within the *ab* plane corresponds to the Munsell hue [10] shift. Alteration of the radius-vector length corresponds to the shift in the Munsell chroma. We move pixels within the *ab* plane in the UPLab space preserving the *L* coordinate to keep luminance unchanged. Gamuts of the sRGB color space in CIELab and UPLab are shown in figure 11.



**Figure 11:** sRGB gamut in CIELab (left) and UPLab (right) color spaces (projection to *ab* plane with *L* axis directed up perpendicular to the image).

We divide the *ab* plane with four rays outgoing from the origin to four sectors of hues. These rays are shown in figure 12 as *OA*, *OB*, *OC* and *OD*. Colors of the purple faces collected through the entire collected base of images are also shown in this figure.



**Figure 12:** Color correction boundaries in UPLab. The points depict all the colors of purple faces from all the images from the entire base.

Let the color hue be in the range *XY* if its *ab* coordinates lie inside a sector formed by the *OX* ray moving to the *OY* ray counter-clockwise. Then hues of all possible colors are in one of the ranges *AB*, *BC*, *CD* or *DA*.

For a given pixel, correction shift depends on its hue:

- Hues in the *DA* range are not changed.
- Hues in the *AC* range are shrank to the *AB* range.
- Hues in the *CD* range are expanded to the *BD* range.

By shrinking/expanding one range to another we mean the following. Let *P* be a point in the *ab* plane in the UPLab space

Russia, Moscow, October 01–05, 2012

corresponding to the given color (with a persistent *L* coordinate). Let the prolongation of the *OP* ray intersect the sRGB gamut at point *R*. Let the correction move *P* to *P'* with prolongation of *OP'* intersecting the sRGB gamut at point *R'*. Then, if the color hue in the *XY* range is shrank/expanded to the *X'Y'* range:

$$\frac{\angle XOP}{\angle X'Y'} = \frac{\angle X'OP'}{\angle X'Y'} \quad \text{and} \quad \frac{|OR|}{|OP|} = \frac{|OR'|}{|OP'|}.$$

The *A* boundary passes near a distinctive cluster of blue colors on the sRGB gamut. The *B* boundary specifies the strength of correction and passes near the “most purple” observable monochromatic color. It may be varied to achieve the most pleasing result. The *C* boundary is selected so that all colors of purple faces collected through the entire image base lie within the *AC* range. The *D* boundary is selected somewhat arbitrarily to space the ranges evenly and to lie within the purple line of colors so that its hue would not represent any monochromatic color.

The proposed correction algorithm includes intersection of rays with 3D gamuts and conversion from the sRGB color space to the UPLab color space and backwards. These tasks require significant time to be performed, but since we precompute the LUT, the do not affect overall efficiency of the correction algorithm. It took us about 5 minutes to construct a LUT for all possible 16,777,216 sRGB colors. We used CGAL AABB Trees [11, 3D Fast Intersection and Distance Computation (AABB Tree)<sup>4</sup>] to compute intersections of rays with gamuts. The size of full LUT is 48MiB, but it can be easily compressed to  $\approx 7$  MiB using a run-length-encoding technique since most of the colors are unaffected by the correction and it can still be efficiently accessed in the compressed form.

## 6 RESULTS

To test the proposed correction algorithm, we prepared three correction LUTs differing in strength, i.e. with different locations of the *B* boundary. The feasibility of the algorithm was estimated manually by analyzing results for different samples from the base. Results for two objects under two different exposures are shown in figure 13 with close-up on purple faces. Results for other images from the base are available in the internet<sup>5</sup>.

Note how the rest of the image remain unchanged by correction and how purple faces preserve small details, which are of particular interest since they can be used to estimate clarity of the object. It is also visible, that the color of the faces in the images taken with lower exposure remains unaffected by the correction since it has the natural blue hue.

## 7 CONCLUSION

During our engineering work we have stumbled upon limitations of applicability of digital still cameras, where specific colors present in our scene could not be reproduced properly. We have analyzed sources of this issue and proposed a very efficient algorithm for its correction.

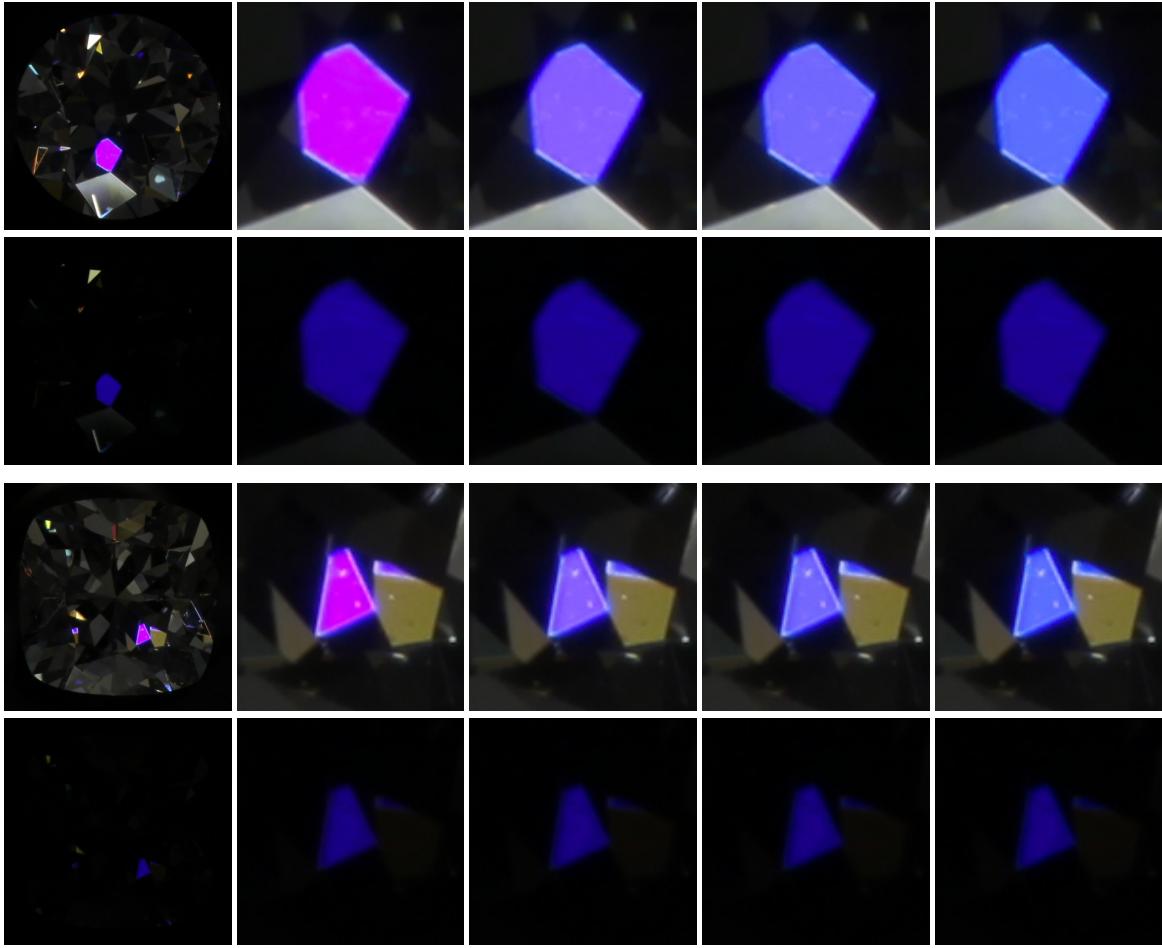
The study of the issue was conducted without access to the spectra of the color of interest because of a small physical size of an examined object. While proposed algorithm can be improved for more accurate color correction and developed to handle more general scenes, it suits our needs well and allows us to address the camera’s limitations.

## 8 ACKNOWLEDGEMENTS

This work was done in cooperation with and with financial and technical support of OctoNus Software Ltd.

<sup>4</sup>[http://www.cgal.org/Manual/latest/doc.html/cgal\\_manual/AABB\\_tree/Chapter\\_main.html](http://www.cgal.org/Manual/latest/doc.html/cgal_manual/AABB_tree/Chapter_main.html)

<sup>5</sup><ftp://graphics.cs.msu.ru/projects/PurpleFires/2012-04-21 - MSS/all.html>



**Figure 13:** Examples of color correction done by proposed algorithm for two different objects, two rows per object. Odd rows contain images of the higher exposure (#8 in set) with the clearly visible purple face, even rows contain corresponding images with the lower exposure (#2 in set) with the same face of the natural blue color. The leftmost image in each row is the initial JPEG image taken by the camera, the second image to the right is a close-up of the purple face in this image. The remaining three images are close-ups of the same face in images corrected with different strength settings, i.e. with different locations of the  $B$  boundary. Correction strength increases from left to right.

## 9 REFERENCES

- [1] T. Smith and J. Guild, “The CIE colorimetric standards and their use,” *Transactions of the Optical Society*, vol. 33, pp. 73, 1931.
- [2] J. Adams, K. Parulski, and K. Spaulding, “Color processing in digital cameras,” *Micro, IEEE*, vol. 18, no. 6, pp. 20–30, 1998.
- [3] K.E. Spaulding, R.M. Vogel, and J.R. Szczepanski, “Method and apparatus for color-correcting multi-channel signals of a digital camera,” Sept. 8 1998, US Patent 5,805,213.
- [4] C.S. McCamy, H. Marcus, and JG Davidson, “A color-rendition chart,” *J. App. Photog. Eng*, vol. 2, no. 3, pp. 95–99, 1976.
- [5] G. Hong, M.R. Luo, and P.A. Rhodes, “A study of digital camera colorimetric characterisation based on polynomial modelling,” 2001.
- [6] IT ANSI, “7/2-1993 (ISO 12641),” *Graphic Technology-Color Reflection Target for Input Scanner Calibration*.
- [7] J. Park and K. Park, “Professional colour communicator-the definitive colour selector,” *Journal of the Society of Dyers and Colourists*, vol. 111, no. 3, pp. 56–57, 1995.
- [8] “PGM format specification,” <http://netpbm.sourceforge.net/doc/pgm.html>, Accessed: 27/05/2012.

- [9] “Bruce lindbloom’s web site,” <http://brucelindbloom.com>, Accessed: 27/05/2012.
- [10] S.M. Newhall, D. Nickerson, and D.B. Judd, “Final report of the OSA subcommittee on the spacing of the munsell colors,” *JOSA*, vol. 33, no. 7, pp. 385–411, 1943.
- [11] “CGAL - computational geometry algorithms library,” <http://www.cgal.org>, Accessed: 27/05/2012.

## ABOUT THE AUTHORS

Mikhail Matrosov is a Ph.D. student at Moscow State University, Department of Computational Mathematics and Cybernetics. His research interests include interactive image enhancement, 3D reconstruction, HDR tone mapping, color theory and adjacent fields. His contact email is matrosov@graphics.cs.msu.ru.

Alexey Ignatenko is a researcher at Computational Mathematics and Cybernetics department of Moscow State University. His research interests include photorealistic 3D rendering, 3D modelling and reconstruction, image-based rendering and adjacent fields. His contact e-mail is ignatenko@graphics.cs.msu.ru.

Sergey Sivovolenko is the CEO of OctoNus Software Ltd. His contact e-mail is sivovolenko@octonus.com.

# Learning Graph Laplacian for Image Segmentation

Sergey Milyaev

Radiophysics Department

Voronezh State University, Voronezh, Russia

sergey.milyaev@gmail.com

Olga Barinova

Department of Computational Mathematics and Cybernetics

Lomonosov Moscow State University, Moscow, Russia

obarinova@graphics.cs.msu.ru

## Abstract

In this paper we formulate the task of semantic image segmentation as a manifold embedding problem and solve it using graph Laplacian approximation. This allows for unsupervised learning of graph Laplacian parameters individually for each image without using any prior information. We perform experiments on GrabCut, Graz and Pascal datasets. At a low computational cost proposed learning method shows comparable performance to choosing the parameters on the test set. Our framework for semantic image segmentation shows better performance than the standard discrete CRF with graph-cut inference.

**Keywords:** semantic image segmentation, unsupervised learning, manifold embedding.

## 1. INTRODUCTION

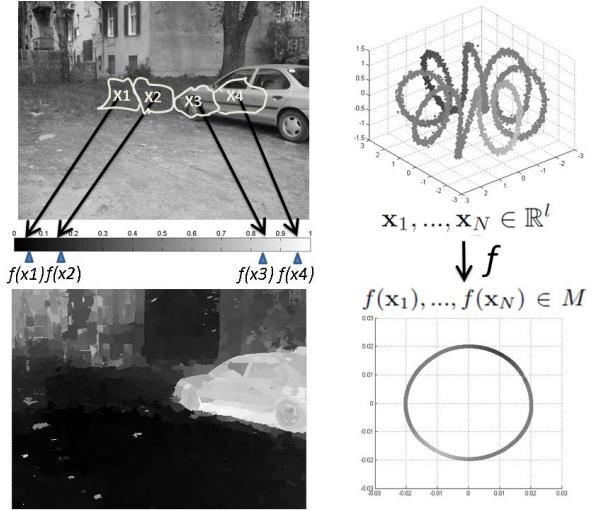
We consider the task of semantic image segmentation that implies assigning a label from a given set to each image pixel. Various discrete CRF models have been proposed for this task [1] [2], [3]. It was shown that learning the parameters of CRF improves its performance [4], [5]. In this work we propose an alternative view on semantic image segmentation.

Methods based on graph Laplacians show state-of-the-art results for interactive image segmentation [6] and image matting [7]. They require just a few local computations and solving one sparse linear system, which can be done very efficiently. In this work we propose a formulation of image segmentation task in terms of manifold embedding and discretize the problem using graph Laplacian approximation.

Graph Laplacian methods have the parameters very similar to those of discrete CRFs. While a remarkable progress has been done in the direction of learning the parameters of discrete CRFs (see e.g. [4], [5]). However the methods for learning parameters of discrete CRF are not applicable to graph Laplacian. Thus the parameters of graph Laplacian are usually chosen by validation on hold-out dataset. The use of validation limits the potential number of parameters used. Moreover, the optimal values of parameters can vary significantly from one image to another, therefore choosing the parameters individually for each image is desirable.

Our formulation of image segmentation problem leads to a novel method for unsupervised learning of graph Laplacian parameters, which is the main contribution of this paper. Our method is designed specifically for the task of semantic image segmentation and provides the values of parameters individually for each test image without using any kind of supervision. Proposed method is computationally efficient and achieves performance comparable to choosing the parameters on the test set, which eliminates the need of using hold-out set or cross-validation. In experimental comparison on Graz and Pascal datasets shows proposed method shows better performance than the standard discrete CRF with graph-cut inference.

The remainder of the paper is organized as follows. We start by discussing related work. In section 3 we describe the image segmentation framework proposed in this paper. In section 4 we present our method for unsupervised learning of graph Laplacian parameters. We proceed to the experimental evaluation of the



**Figure 1:** We formulate semantic image segmentation task as one-dimensional manifold embedding problem. This allows for unsupervised learning of graph Laplacian parameters individually for each image.

proposed method.

## 2. RELATED WORK

The task of semantic image segmentation implies assigning a label from a given set to each image pixel. Various discrete CRF models have been proposed for this task [1] [2], [3]. Learning the parameters of CRF can improve performance of semantic image segmentation [4], [5]. In this work we use an alternative formulation of semantic image segmentation problem that leads to using graph Laplacian instead of discrete CRF.

Methods based on graph Laplacians have emerged recently and proved very efficient for interactive image segmentation [6] and image matting [7]. Graph Laplacian methods allow interpretation in terms of MAP estimation in real-valued CRF [8]. A few other interpretations of graph Laplacian methods have been suggested in the literature.

In [6] Grady suggested explanation of using Laplacians for interactive segmentation in terms of random walks. In [9] the use of graph Laplacian for interactive image segmentation was explained in terms of transductive inference. Hein et al. [10] showed that graph Laplacian provides a good approximation for  $s$ -weighted Laplace operator. Therefore, graph Laplacians propose a discrete alternative to the problem of finding a smooth function such that its values in seed pixels are close to the associated labels and it is allowed to vary only on low-density regions of the input space.

In contrast to these works we derive the graph Laplacian by viewing image segmentation as a manifold embedding task. In contrast to [11] we use manifold embedding for semantic image segmentation and not the unsupervised image segmentation. This formulation of semantic image segmentation allows for unsupervised learning of graph Laplacian parameters.

The idea of our learning method is based on the properties of graph Laplacian approximation of Laplace-Beltrami operator studied in [12]. Coifman et al. in [13] proposed a method for automatic selection the kernel bandwidth of graph Laplacian for the problem of optimal manifold embedding. In contrast to [13] we design the method specifically for the task of semantic image segmentation. While the method proposed in [13] aims at choosing one parameter (kernel bandwidth), our method can handle multiple parameters.

### 3. ONE-DIMENSIONAL MANIFOLD EMBEDDING FOR SEMANTIC IMAGE SEGMENTATION

First we discuss the task of manifold embedding and then explain our formulation of image segmentation problem.

#### 3.1 Manifold embedding.

Suppose we have a set of input points  $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^l$ . Let  $d : \mathbb{R}^l \times \mathbb{R}^l \rightarrow \mathbb{R}$  be a symmetrical function giving the distance in  $\mathbb{R}^l$ . The optimal manifold embedding task is to find a smooth differentiable function  $f$  that maps the input space  $\mathbb{R}^l$  on the embedded Riemanian manifold  $M$  of dimensionality  $m$  ( $m < l$ ) (see Figure 1, left column). The function  $f$  should preserve distances between the points, such that if  $d(\mathbf{x}_i, \mathbf{x}_j)$  is small, then  $\|f(\mathbf{x}_i) - f(\mathbf{x}_j)\|$  should be small.

Let us focus on the case when the dimension of manifold  $M$  equals one ( $M = \mathbb{R}$ ). Consider two points  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^l$ . They are mapped to  $f(\mathbf{x}), f(\mathbf{y}) \in \mathbb{R}$  respectively. It can be shown [12] that

$$|f(\mathbf{x}) - f(\mathbf{y})| \leq d(\mathbf{x}, \mathbf{y}) \|\nabla f(\mathbf{x})\| + o(d(\mathbf{x}, \mathbf{y})), \quad (1)$$

where  $\nabla f(\mathbf{x})$  is the gradient of function  $f(\mathbf{x})$ . Thus we see that  $\nabla f(\mathbf{x})$  provides us with the measure of how far apart  $f$  maps nearby points.

We consider the problem of *initialized one-dimensional manifold embedding* when 1)  $M = \mathbb{R}$  and 2) initial estimates  $y_1, \dots, y_N$  of  $f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)$  in  $\mathbb{R}$  are given. Suppose we know confidences  $c_i \geq 0, i = 1, \dots, N$  that reflect our belief in initial estimates of  $f(\mathbf{x}_i), i = 1, \dots, N$ . Using (1) the problem of initialized one-dimensional manifold embedding can be formulated as minimization the following energy with respect to  $f$

$$E(f) = \sum_i c_i (y_i - f(\mathbf{x}_i))^2 + \int_M \|\nabla f\|^2 dV, \quad (2)$$

where the integral is taken with respect to a standard measure on a Riemanian manifold. The first term in (2) guarantees that corresponding one-dimensional vectors  $f(\mathbf{x}_i)$  are close to their initial estimates  $y_i$ . The second term guarantees that if the points  $\mathbf{x}_i, \mathbf{x}_j$  are close in the input space then their images  $f(\mathbf{x}_i)$  and  $f(\mathbf{x}_j)$  are close in  $M$ .

It follows from the Stokes' theorem that  $\int_M \|\nabla f\|^2 dV = \int_M \Delta_M(f) f dV$ , where  $\Delta_M(f)$  is the Laplace-Beltrami operator. It is a second order differential operator defined as the divergence of the gradient of a function defined on  $M$ .

In many cases finding the mapping  $f$  explicitly is not required. The goal then is to find a set of points  $f(\mathbf{x}_1), \dots, f(\mathbf{x}_N) \in M$  such that represent  $\mathbf{x}_1, \dots, \mathbf{x}_N$ .

#### 3.2 Image segmentation as the manifold embedding problem.

For the sake of clarity first we consider the task of object/background image segmentation. We aim to find real-valued alpha-matting coefficients for each image pixel, the segmentation is then done by thresholding the result. Below we formulate image segmentation problem as the problem of initialized one-dimensional manifold embedding.

Suppose each image pixel is mapped in a feature space  $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^l$ . For example the features can include the spatial coordinates and color of the pixels. Suppose we have defined a

distance function between two pixels  $d : \mathbb{R}^l \times \mathbb{R}^l \rightarrow \mathbb{R}$  that tells how likely it is that both pixels belong to object/background.

Suppose that for each image pixel we know the output of some local model  $0 \leq y_i \leq 1, i = 1, \dots, N$  that tells how likely it is that the pixel is a part of the object. Suppose also that we know confidences  $c_i \geq 0, i = 1, \dots, N$  that indicate how much belief we put in the local model.

Our goal is to find real-valued  $f_1, \dots, f_N$  that refine the outputs of local model  $y_1, \dots, y_N$ . We require that  $f_1, \dots, f_N$  lie in the optimally embedded one-dimensional manifold  $M$  and each  $f_i$  corresponds to  $\mathbf{x}_i$ . Therefore the problem of image segmentation reduces to minimization of energy (2).

#### 3.3 Approximation of Laplace-Beltrami operator.

We will now define a graph Laplacian that is an approximation of Laplace-Beltrami operator. Denote weight matrix by  $W : W_{ij} = \exp(-d(\mathbf{x}_i, \mathbf{x}_j)^2)$  (in this work we consider only Gaussian kernel). Let  $g_i = \sum_j w_{ij}$  stand for a sum of  $W$  along the  $i$ -th row. Denote diagonal matrix with values  $g_i$  on diagonal by  $D$ . Graph Laplacian is defined as a matrix  $L = W - D$ .

Belkin et al. [14] showed that graph Laplacian  $L$  converges to Laplace-Beltrami operator in the limit  $N \rightarrow \infty$ . In this sense, the graph Laplacian is a numerical machinery for approximating a specific operator on the underlying manifold, by using a finite subset of points.

#### 3.4 Discretization of the problem with graph Laplacian.

Using approximation of Laplace-Beltrami operator by graph Laplacian the problem (2) reduces to minimization of the following energy function with respect to vector  $\mathbf{f} = (f_1, \dots, f_N)$ :

$$E(\mathbf{f}) = \sum_i c_i (f_i - y_i)^2 + \sum_{i,j} w_{ij} (f_i - f_j)^2. \quad (3)$$

The first term in (3) repeats the first term in (2) and the second term in (3) is a discrete approximation of the second term in (2) according to [14]. Minimization of the energy  $E(\mathbf{f})$  can also be interpreted as MAP inference in a real-valued CRF, which are given by the real-valued outputs  $f_1, \dots, f_N$ .

In the matrix form (3) takes the following form:

$$E(\mathbf{f}) = (\mathbf{f} - \mathbf{y})^T C (\mathbf{f} - \mathbf{y}) + \mathbf{f}^T L \mathbf{f}, \quad (4)$$

where  $C$  denotes a square diagonal matrix with  $c_i$  on diagonal and  $\mathbf{y}$  denotes an  $N$ -dimensional vector of initial likelihood scores  $y_i$ . This optimization problem reduces to solving a sparse linear system:

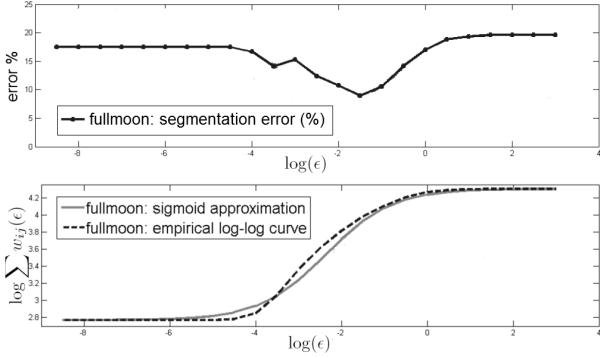
$$(L + C)\mathbf{f} = C\mathbf{y}. \quad (5)$$

The object/background segmentation algorithm then consists in: 1) computing graph Laplacian matrix  $L$ ; 2) solving the sparse linear system (5); 3) thresholding the output.

Described formulation fits both in interactive segmentation scenario and in semantic image segmentation scenario. In case of interactive segmentation confidence values  $c_i$  are infinite for pre-labelled seed points, and 0 for a test points,  $y_i = 1$  for seeds marked as object and equals 0 for background seeds. For semantic segmentation we assume that initial estimates  $y_i$  and confidences  $c_i$  are provided by local models (e.g. appearance model of a specific category).

We notice that instead of pixels we can use image superpixels without making any changes in the algorithm. In the case then superpixels intersect we can average results of (5) for all superpixels that cover image pixel to obtain pixel-wise result.

This framework can be extended to a multi-class segmentation. Let  $K$  denote the number of labels corresponding to object categories. If we solve (5) for each label  $l$  vs all other labels



**Figure 2:** Top: segmentation errors for the "fullmoon" image from GrabCut database with respect to  $\log \epsilon$  ( $\alpha$  is fixed). Bottom: Dashed line - logarithmic plot for the "fullmoon" image with respect to  $\log \epsilon$  ( $\alpha$  is fixed). The optimal value of  $\epsilon$  is chosen in the point of maximum derivative of the logarithmic plot; Solid line - sigmoid fit of the logarithmic plot.

$1, \dots, l-1, l+1, \dots, K$  and obtain the values  $y_i^{(l)}$  for all image pixels; at the end, an  $i$ -th image pixel is assigned to the label  $l_{max}$ , where  $l_{max} = \arg \max_{l=1, \dots, K} y_i^{(l)}$ .

#### 4. UNSUPERVISED LEARNING OF GRAPH LAPLACIAN PARAMETERS

Suppose that the distance function  $d$  is represented as a weighted sum of metrics  $d_i : \mathbb{R} \times \mathbb{R} \rightarrow R^+; i = 1, \dots, K$ :

$$d(\mathbf{x}_i, \mathbf{x}_j)^2 = \frac{1}{\epsilon} \sum_{k=1}^K \alpha_k d_k(\mathbf{x}_i, \mathbf{x}_j)^2, \quad (6)$$

with fixed  $\alpha_1 = 1$ . Therefore the parameters of graph Laplacian  $\alpha_i, i = 2, \dots, l$  are the weights of features  $\mathbf{x}^k, i = 2, \dots, l$  and the kernel bandwidth  $\epsilon$ . Below we show that optimal value of  $\epsilon$  is determined by the values of  $\alpha_i, i = 2, \dots, l$ .

##### 4.1 Kernel bandwidth $\epsilon$ selection with fixed $\alpha$ .

We start by fixing the parameters  $\alpha_i, i = 2, \dots, l$ . As shown in [13], if we assume that  $L$  provides a good approximation of Laplace-Beltrami operator then the following condition holds:

$$\log \sum_{i,j} w_{ij}(\epsilon) \approx m/2 \log(\epsilon) + \log \left( \frac{N^2 (2\pi)^{m/2}}{\text{vol}(M)} \right), \quad (7)$$

where  $m$  is a dimensionality of corresponding manifold  $M$  and  $w_{ij}$  are the elements of the weight matrix  $W$ .

Consider the *logarithmic plot* of  $\log \sum_{i,j} w_{ij}$  with respect to  $\log \epsilon$ . Figure (2) shows the plot of  $\log \sum_{i,j} w_{ij}$  with respect to  $\log \epsilon$  and  $\log \alpha$  for one image from GrabCut dataset. According to (7) if the approximation is good then the slope of this plot  $\epsilon$  should be about the half dimensionality of corresponding manifold. In the limit  $\epsilon \rightarrow \infty$ ,  $w_{ij} \rightarrow 1$ , so  $\sum_{i,j} w_{ij} \rightarrow N^2$ . On the other hand, as  $\epsilon \rightarrow 0$ ,  $w_{ij} \rightarrow \delta_{ij}$ , so  $\sum_{i,j} w_{ij} \rightarrow N$ . These two limiting values set two asymptotes of the plot and assert that logarithmic plot cannot be linear for all values of  $\epsilon$ .

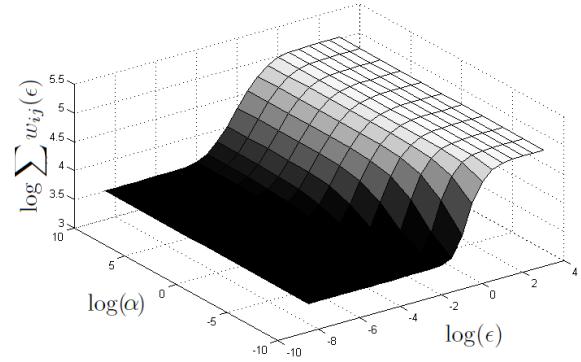
Therefore in order to get better approximation of Laplace-Beltrami operator with  $\alpha_1, \dots, \alpha_K$  fixed we have to choose the value of  $\epsilon$  from the linear region of logarithmic plot. We use the point of maximum derivative as the point of maximum linearity.

##### 4.2 Implementation details

We use the distance function from [9]:

$$\tilde{d}^2(\mathbf{x}_i, \mathbf{x}_j) = \frac{\|r_i - r_j\|^2}{\sigma_r^2} + \frac{\|x_i - x_j\|^2}{\sigma_g^2}, \quad (8)$$

where  $r$  encodes mean RGB color in the superpixel,  $x$  encodes coordinates of the center of the superpixel, parameters of the



**Figure 3:** The plot of  $\log \sum_{i,j} w_{ij}$  with respect to  $\log \epsilon$  and  $\log \alpha$ . The plot shown is shown in (2, bottom) corresponds to the 2-d slice of this 3-d plot for fixed  $\alpha$ . Note that the slope of linear region are not constant for all values of  $\alpha$ . We seek for  $\alpha$  such that the slope in the linear region equals 0.5.

method  $\sigma_r > 0$  and  $\sigma_g > 0$  are the scales of chromatic and geometric neighbourhoods respectively.

This distance function (8) can be rewritten in the form of (6) as:

$$\tilde{d}^2(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\epsilon} (\|r_i - r_j\|^2 + \alpha \|x_i - x_j\|^2), \quad (9)$$

where  $\epsilon = 0.5\sigma_r^2$  and  $\alpha = \sigma_r^2/\sigma_g^2$ . Therefore, the distance function has two parameters  $\epsilon$  and  $\alpha$ .

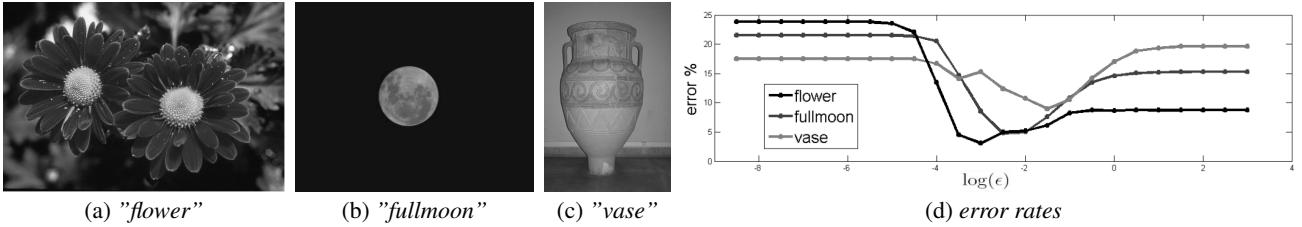
As follows from (7) the slope of the logarithmic curve near optimal value of  $\epsilon$  has to be close to  $m/2$ , where  $m$  is the dimensionality of manifold  $M$ . In our case  $m = 1$ , therefore the slope of the logarithmic plot has to be 0.5. If the plot has different slope in the linear region, this indicates that the second term in (7) is large.

So in the first step of our learning method we should find such  $\alpha$  that the slope of logarithmic plot of  $\log \sum_{i,j} w_{ij}(\epsilon)$  from  $\epsilon$  is equal to 0.5. In the second step we use the sigmoid fit of the logarithmic plot. The shape of logarithmic plot can be approximated with a sigmoid function:  $T(x) = A/(B + \exp(Cx + D)) + E$ . Since the asymptotes of the sigmoid are set by (7) and the slope in the linear region of the sigmoid should be 0.5 the sigmoid has only one free parameter that controls the shift of the sigmoid along horizontal axis. Figure 2 illustrates the choice of  $\epsilon$  according to sigmoid approximation. In our experiments the values of  $\epsilon$  take values as degrees of 10 and the values of  $\alpha$  take values as degrees of 2.

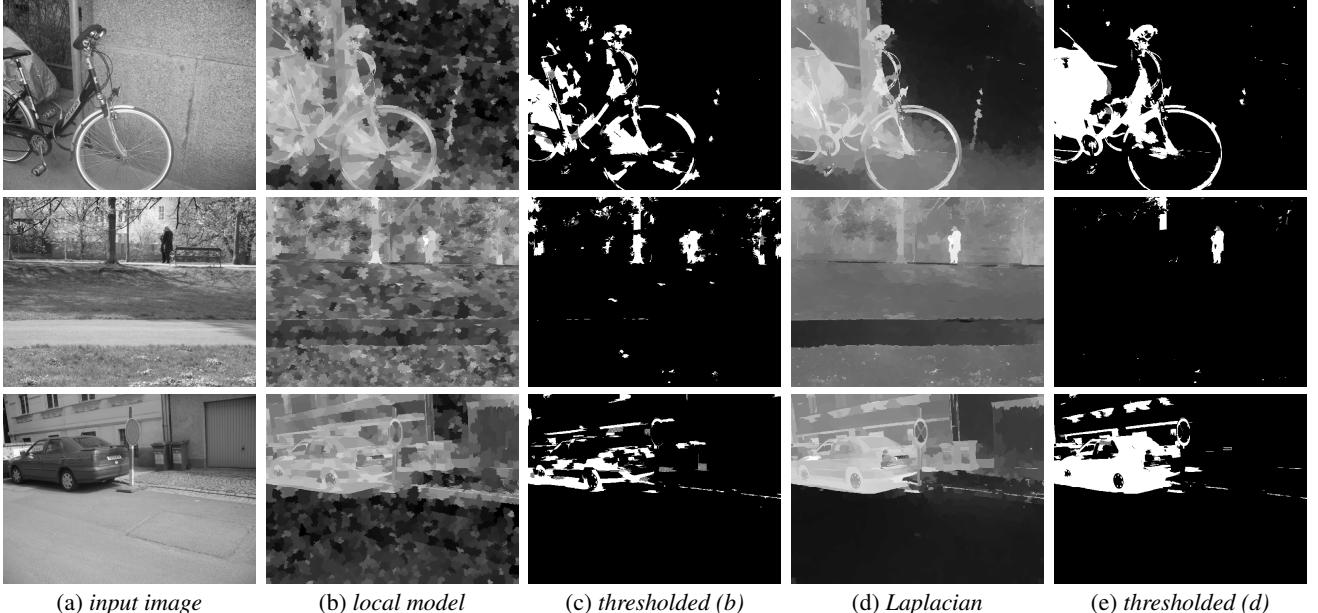
We found empirically that usually the slope of the logarithmic plot is greater than 0.5 for large  $\alpha$  and is less than 0.5 for small  $\alpha$ . In most cases the slope of the logarithmic plot  $S(\alpha)$  is monotonic function of  $\alpha$ . One of the possible explanations of this fact can be the following. Small  $\alpha$  correspond to using only spatial information. This implies that the dimension of manifold where the data lives is 2 and it is difficult to reduce dimensionality further. By decreasing  $\alpha$  we decrease the weight of spatial information in the distance function therefore it gets easier to find the corresponding one-dimensional manifold. On the other hand large  $\alpha$  corresponds to increased weight of color information. Infinite  $\alpha$  corresponds to using color information alone. As long as the color space is three-dimensional and the color distribution of object and background is complex it is difficult to embed one-dimensional manifold in the input points. Example of the logarithmic plot with respect to both  $\epsilon$  and  $\alpha$  is shown in Figure 3.

## 5. EXPERIMENTS

For the experiments we used GrabCut, Graz and Pascal 2007 datasets. In all experiments graph Laplacian operated with superpixels produced by image over-segmentation methods. Each



**Figure 4:** Segmentation errors depending on graph Laplacian parameter  $\epsilon$  on three images from GrabCut dataset. Note that minimal error is achieved on different values of  $\epsilon$  for different images.



**Figure 5:** Results of SVM and graph Laplacian method for images from Graz dataset. (a) - input images of "bike", "person" and "cars" classes; (b) - real-valued output from local SVM model; (c) - results of thresholding the SVM outputs; (d) - real-valued output of graph Laplacian using SVM as a local model with the parameters learnt by our method; (e) - thresholded output of our method. Note how graph Laplacian refines the output from SVM. It doesn't oversmooth the result and preserves fine details like the wheel of the bike and the small figure of the person.

superpixel was linked with a fixed number of its nearest neighbours, and the distances to other superpixels were assumed infinite. For all experiments we used confidences that are a linear function of the outputs of local appearance models  $c_i = 0.5(1 - |p_i - 0.5|)$ .

## 5.1 GrabCut image database.

GrabCut image database contains 50 images of different objects<sup>1</sup>. In the experiments we used the set of superpixels which is the union of oversegmentations provided by Colour Structure Code and Watershed segmentation methods.

Figure 4 (d) shows the error on 3 different images from GrabCut database with respect to  $\log \epsilon$  ( $\alpha$  is fixed). Depending on the choice of  $\epsilon$  one can get different values of errors and for each image, and the optimal values of  $\epsilon$  are different for different images.

We measured performance on GrabCut dataset according to standard metric [1]. We compared three versions of graph Laplacian. First, we chose the best parameters for each image individually by validation on the same image in order to obtain the top bound on performance of graph Laplacian. The resulting error rate is 8.7%. In the second experiment we chose single set of parameters for the whole dataset by validation on the test dataset. This corresponds to upper bound on performance of the method with

fixed parameters. The resulting error rate is  $9.9 \pm 4.3\%$ . Finally we evaluated the performance of graph Laplacian with the parameters learnt individually for each image by our method. The resulting error rate is  $10.2 \pm 4.0\%$ . The results of graph Laplacian with learnt parameters is very close to the upper bound of performance of graph Laplacian with fixed parameters. Notably, the standard deviation of errors obtained with the parameters learnt individually for each image is smaller than of Laplacian with fixed parameters.

The learning phase took from 0.5 seconds to 3 seconds, solving linear system 5 took from 0.05 second to 0.5 seconds depending on the number of superpixels in the image (the total number of superpixels varied from 500 to 30000).

## 5.2 Graz image dataset.

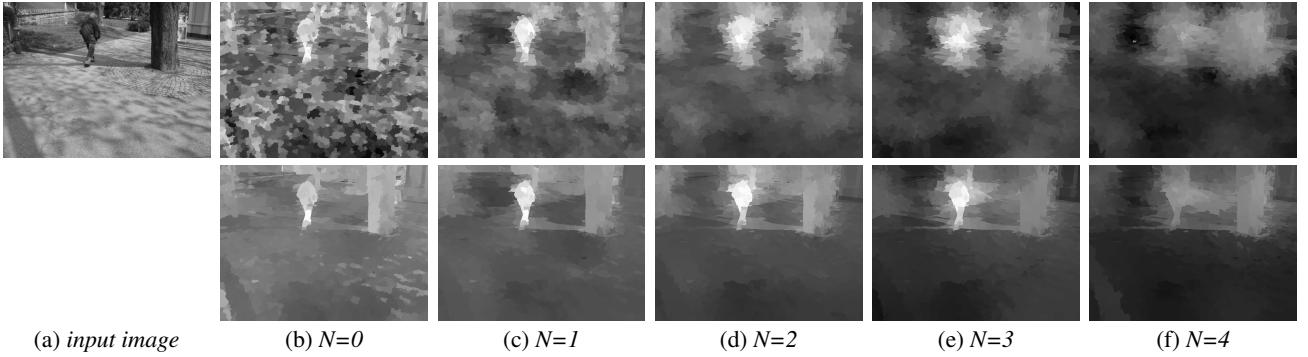
Graz dataset<sup>2</sup> contains 1096 images of three classes: "person", "bike" and "car". In our experiments we solved a separate binary segmentation problem for each category. To measure the quality of segmentation we used a standard metric - percent of incorrectly classified pixels in the image.

In our experiments we used an open-source VIBlocks toolbox<sup>3</sup>, which implements the method described in [15]. We chose it for comparison for the following reasons. First, it allows using dif-

<sup>1</sup>available at <http://research.microsoft.com/en-us/um/cambridge/projects/visionimagevideoediting/segmentation/grabcut.htm>

<sup>2</sup>available at <http://www.emt.tugraz.at>

<sup>3</sup>code available at <http://viblocks.org/index.html>



**Figure 6:** Results of using different local models. The first row shows real-valued output of local appearance models. The second row shows results of our method. Parameter  $N$  sets the size of superpixel neighborhood in the local model. The effect of using graph Laplacian is better visible for smaller  $N$ .

	N=0			N=1			N=2			N=3			N=4		
	cars	bike	pers	cars	bike	pers	cars	bike	pers	cars	bike	pers	cars	bike	pers
<i>SVM</i>	41.9	56.5	49.4	59.6	66.9	63.6	68.0	69.2	66.6	69.4	70.7	65.2	66.5	71.9	63.6
+ <i>CRF</i>	43.0	57.7	49.3	60.2	67.1	63.9	70.1	70.2	66.9	70.7	71.0	65.4	68.8	72.2	64.2
+ <i>Laplacian (valid.GrabCut)</i>	50.0	60.1	56.0	65.5	68.7	68.5	71.6	70.8	70.8	72.2	72.0	69.5	70.0	<u>73.2</u>	67.3
+ <i>Laplacian (valid.test set)</i>	<b>56.6</b>	<b>63.3</b>	<b>59.1</b>	66.3	68.4	68.8	71.9	70.4	70.4	72.6	71.2	69.4	70.8	72.2	68.0
+ <i>Laplacian (learnt)</i>	54.2	60.9	58.5	65.1	66.8	<b>69.4</b>	<b>72.0</b>	69.5	<b>71.3</b>	<b>73.3</b>	70.3	<b>70.2</b>	<b>71.4</b>	71.5	<b>68.9</b>

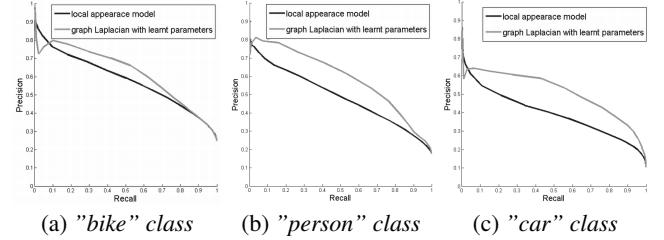
**Table 1:** Performance on Graz dataset at equal precision and recall rates for "cars", "bike" and "person" classes. First row: local appearance model (from VIBlocks toolbox). Second row: result of applying discrete CRF with graph cut inference (from VIBlocks toolbox). Third row: graph Laplacian with parameters validated on GrabCut dataset. Fourth row: graph Laplacian with parameters validated on the test set. Fifth row: graph Laplacian with parameters learnt individually for each image. For each appearance model used in our experiments (we varied the number of neighboring regions as in [15]) the best result is shown in **bold font**. Underlined are the best overall results.

ferent local appearance models. The method has a parameter  $N$  meaning number of neighbouring superpixels which features are used for classification of each particular superpixel. So we report performance metrics for different values of  $N$  to illustrate the performance of proposed graph Laplacian framework applied to different local models. Second, the toolbox includes implementation of discrete CRF with graph-cut inference, which we use for comparison. Note, this CRF model uses similar types of features (color and spatial coordinates of superpixels) to those used in our graph Laplacian. We used the same over-segmentation and the same local appearance model based on SVM as [15]. To obtain initial estimates  $y_i$  for graph Laplacian framework we scaled SVM outputs to  $[0, 1]$  interval for each image.

In the first experiment the parameters  $\epsilon$  and  $\alpha$  were validated on the GrabCut dataset. In the second experiment we validated the parameters on the test set. In the third experiment we used our unsupervised learning method for choosing the parameters individually for each image. We also compared with VIBlocks implementation of CRF with graph-cut inference. The strategy for choosing internal parameters of CRF was the same as in [15].

Table 1 contains results of the comparison. Our unsupervised learning gives results comparable to upper bound on performance of graph Laplacian with fixed parameters from the second experiment. The value of performance gain compared to local appearance model differs for different values of parameter  $N$ . The smaller  $N$  is the smaller neighborhood is considered by low-level model, and the more significant is the gain in performance attained by both CRF and graph Laplacian. The gain in performance of graph Laplacian is almost uniformly higher than the performance gain obtained by discrete CRF. Note that the gain achieved by graph Laplacian is several times higher than the one achieved by discrete CRF for  $N = 0$ . Figure 7 shows precision-recall curves for local appearance models and for our method to illustrate the gain in performance due to graph Laplacian.

Figure 5 shows results provided by local appearance model (SVM) and corresponding results of using graph Laplacian with



**Figure 7:** Precision-recall curves for "bike", "person" and "car" classes of Graz dataset. *Blue curves* - local appearance model ( $N=0$ ); *Green curves* - graph Laplacian with learnt parameters.

learnt parameters. Figure 6 shows how the results vary for different local models.

The running time on Graz dataset is the following: learning phase takes about 0.2 seconds on average, solving of linear system 5 takes about 0.02 seconds on average.

### 5.3 Pascal 2007 image dataset.

Pascal 2007 dataset <sup>4</sup> contains 21 classes. Again in this experiment we use local models from VIBlocks toolbox trained with parameters as in [15]. We compare our graph Laplacian method with unsupervised learning to the discrete CRF implemented in VIBlocks toolbox. The training and testing split is defined in the challenge. We train local model on the train set and choose the parameters of discrete CRF on the validation set. We do not use the validation set in the experiment with our graph Laplacian method and use our unsupervised learning method for choosing Laplacian parameters for each particular image.

The table 5.2 shows the results of comparison to using local

<sup>4</sup>available at <http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/>

	background	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	diningtable	dog	horse	motorbike	person	pottedplant	sheep	sofa	train	tv monitor	Average	Pixels %
Pantofaru <i>et al.</i> [16]	59	27	1	8	2	1	32	14	14	4	8	32	9	24	15	81	11	<b>26</b>	1	28	17	20	-
Shotton <i>et al.</i> [17]	33	46	5	14	11	14	34	8	6	3	10	39	40	28	23	32	19	19	8	24	9	20	-
Fulkerson <i>et al.</i> [15]	56	26	<b>29</b>	<b>19</b>	16	3	<b>42</b>	44	<b>56</b>	23	6	11	62	16	68	46	16	10	21	52	<b>40</b>	<b>32</b>	51
local model ( $N=0$ )	16	17	8	9	13	8	9	9	29	15	9	12	12	7	13	5	16	16	26	14	21	14	15
+discrete CRF ( $N=0$ )	17	18	8	9	<b>21</b>	8	9	8	28	14	9	13	12	7	13	5	16	16	26	14	22	14	16
+Ours(learnt) ( $N=0$ )	23	20	26	12	18	12	17	14	49	17	1	23	16	11	50	4	<b>42</b>	20	<b>44</b>	30	33	23	22
local model ( $N=1$ )	24	13	18	13	9	12	14	24	35	16	9	11	28	15	36	23	14	21	20	35	27	21	38
+discrete CRF ( $N=1$ )	42	6	16	9	6	5	11	14	56	19	4	11	16	16	55	36	24	16	8	56	21	21	38
+Ours(learnt) ( $N=1$ )	33	11	18	14	10	13	16	26	51	16	7	9	35	<b>22</b>	67	29	31	25	16	<b>60</b>	33	26	32
local model ( $N=2$ )	39	10	22	15	11	12	18	36	44	23	8	11	33	15	53	37	17	17	16	36	28	24	36
+discrete CRF ( $N=2$ )	58	9	25	14	6	3	20	38	54	<b>27</b>	12	10	31	7	59	44	12	17	13	43	27	25	50
+Ours(learnt) ( $N=2$ )	52	9	24	17	8	12	19	41	55	21	10	10	<b>37</b>	14	68	42	14	12	14	51	32	27	46
local model ( $N=3$ )	52	13	14	18	8	5	23	38	45	17	7	10	30	21	63	50	17	20	19	43	23	25	46
+discrete CRF ( $N=3$ )	65	10	14	15	5	2	24	40	60	13	6	8	24	19	68	55	18	19	16	46	26	26	56
+Ours(learnt) ( $N=3$ )	63	11	15	17	7	2	22	40	60	13	5	10	32	20	<b>71</b>	57	13	17	14	48	26	27	55
local model ( $N=4$ )	59	6	15	18	4	0	25	44	46	17	3	4	24	20	62	56	15	14	13	38	33	25	51
+discrete CRF ( $N=4$ )	63	8	15	18	4	0	26	<b>46</b>	48	17	2	4	25	20	64	58	12	14	12	38	34	25	54
+Ours(learnt) ( $N=4$ )	69	5	11	<b>19</b>	4	0	28	45	55	13	2	3	18	18	69	<b>60</b>	7	13	7	44	36	25	<b>59</b>

**Table 2:** Results on Pascal 2007 dataset. Best result for each category is shown in bold.

model alone, discrete CRF and our graph Laplacian method with unsupervised learning. For comparison we also reproduce results from [17], [16] and [15]. On this dataset, adding graph Laplacian improves the results significantly compared to using local model alone, and provides a consistent boost for the accuracy as well.

## 6. CONCLUSION

We presented an semantic segmentation framework based on graph Laplacian approximation of manifold embedding problem. The main contribution of this work is a method for choosing internal parameters of graph Laplacian in a fully unsupervised manner individually for each test image. Proposed unsupervised learning method has a low computational cost and shows better performance compared to discrete CRF with graph-cut inference.

## 7. REFERENCES

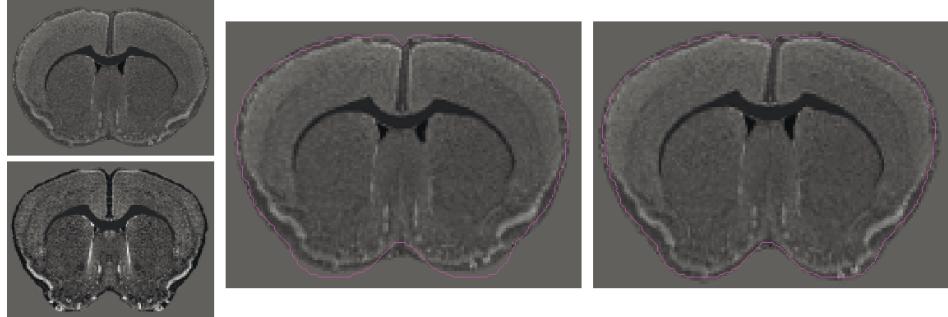
- [1] Yuri Boykov and Marie-Pierre Jolly, “Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images,” in *ICCV*, 2001, vol. 1, pp. 105–112.
- [2] Pushmeet Kohli, Lubor Ladicky, and Philip Torr, “Robust higher order potentials for enforcing label consistency,” in *CVPR*, 2008.
- [3] Lubor Ladicky, Chris Russell, Pushmeet Kohli, and Philip Torr, “Graph cut based inference with co-occurrence statistics,” in *ECCV*, 2010.
- [4] Martin Szummer, Pushmeet Kohli, and Derek Hoiem, “Learning crfs using graph cuts,” in *ECCV*, 2008.
- [5] Sebastian Nowozin, Peter V. Gehler, and Christoph H. Lampert, “On parameter learning in crf-based approaches to object class image segmentation,” in *Proceedings of the 11th European conference on Computer vision: Part VI*, 2010, *ECCV’10*, pp. 98–111.
- [6] Leo Grady, “Random walks for image segmentation,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768–1783, 2006.
- [7] Anat Levin, Dani Lischinski, and Yair Weiss, “A closed form solution to natural image matting,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2008.
- [8] Dheeraj Singaraju, Leo Grady, and Ren Vidal, “P-brush: Continuous valued mrfs with normed pairwise distributions for image segmentation,” in *CVPR*, 2009.
- [9] Olivier Duchenne, Jean-Yves Audibert, Renaud Keriven, Jean Ponce, and Florent Segonne, “Segmentation by transduction,” in *CVPR*, 2008.
- [10] M. Hein, J.-Y. Audibert, and U. von Luxburg, “From graphs to manifolds - weak and strong pointwise consistency of graph laplacians,” *ArXiv Preprint, Journal of Machine Learning Research, forthcoming*, 2006.
- [11] H.B. Zhou and Q.A. Cheng, “O(n) implicit subspace embedding for unsupervised multi-scale image segmentation,” 2011, pp. 2209–2215.
- [12] Mikhail Belkin and Partha Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural Computation*, vol. 15, pp. 1373–1396, 2003.
- [13] Ronald R. Coifman, Yoel Shkolnisky, Fred J. Sigworth, and Amit Singer, “Graph laplacian tomography from unknown random projections,” *IEEE Trans. on Image Processing*.
- [14] Mikhail Belkin and Partha Niyogi, “Towards a theoretical foundation for laplacian-based manifold methods,” in *COLT*, 2005, pp. 486–500.
- [15] Brian Fulkerson, Andrea Vedaldi, and Stefano Soatto, “Class segmentation and object localization with superpixel neighborhoods,” in *ICCV*, 2009.
- [16] Caroline Pantofaru, Cordelia Schmid, and Martial Hebert, “Object recognition by integrating multiple image segmentations,” in *ECCV*, 2008.
- [17] Jamie Shotton, Matthew Johnson, and Roberto Cipolla, “Semantic texton forests for image categorization and segmentation,” 2008.

# A new measure for distance-field based shape matching

Pavel Voronin  
Kurchatov NBICS Center  
pavel.voronin@gmail.com

Andrew Adinetz  
Research Computing Center,  
Lomonosov Moscow State University  
Joint Institute for Nuclear Research  
adinetz@gmail.com

Dmitry P. Vetrov  
Lomonosov Moscow State University  
Kurchatov NBICS Center  
vetrovd@yandex.ru



**Figure 1:** Shape registration via distance field matching: top left, source image; bottom left, target image; middle, source registered onto target, using the classic dissimilarity measure; right, source registered onto target, using the new dissimilarity measure. The target's contour is superimposed in magenta.

## Abstract

One popular method to match two shapes is to register their distance fields as images. We discuss the well-known robustness problems of this approach and identify the noncommutativity of distance transform and geometric transformations as a core issue. Building on this, we propose a simple modification of the method, deriving a new dissimilarity measure. As it involves multiple distance field computations, we also present an efficient GPU-based algorithm for this problem.

**Keywords:** shape matching, registration, distance fields

## 1 Introduction

*Shape registration* (or *shape matching*) is one of the fundamental computer vision problems: given two figures, transform one of them, *source*, into a figure as similar in shape to the other one, *target*, as possible. This task arises naturally in such fields as shape reconstruction [Levoy 2000], shape tracking [Zhou 2005], shape interpolation [Kilian 2007], semantic deformation transfer [Baran 2009], shape recognition [Fahmi 2008], shape retrieval [Belongie 2001], and statistical shape modeling [Zhu 1996]. Combined with intensity-based methods, it has also been widely used for image segmentation [Tsai 2003] and registration [Babalola 2006]. Some of the most important applications of these methods are in medical imaging and related areas [Maintz 1998, Zitova 2003], including data alignment [Malcolm 2008], detection of tumors and nodules [Ginneken 2001], and guidance in neurosurgery [St-Jean 1998]. Our particular interest lies in automatic or user-assisted mapping of experimental brain images onto an atlas (Fig. 1).

More formally, the problem of shape registration can be reformulated as follows. For a pair of admissible figures  $A, B \in \Upsilon$ , and a permissible set of transformations  $\Omega$ , find a transformation  $T^{opt} \in \Omega$  that, when applied to  $A$ , results in a shape that minimizes a certain dissimilarity measure  $F_{dis}$ , regularized by a smoothness

measure  $F_{sm}$ , with respect to  $B$ :

$$T^{opt} = \underset{T \in \Omega}{\operatorname{argmin}} (F_{dis}(T(A), B) + \lambda \cdot F_{sm}(T)). \quad (1)$$

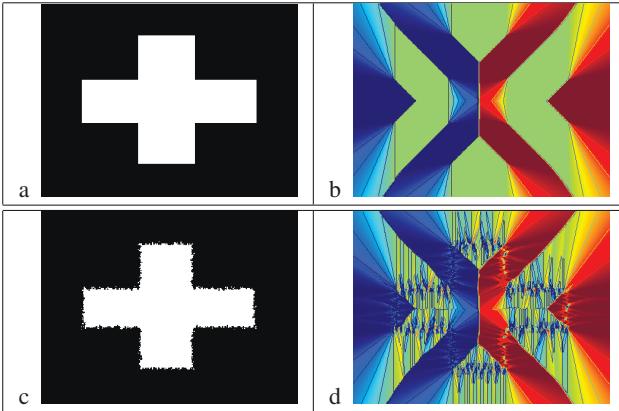
We will use integral of the sum of squared second derivatives as a smoothness measure, which is a popular, but by no means the only, choice [Munim 2007, Paragios 2003].

Design and complexity of shape registration are largely defined by the permissible set of transformations, or *transformation model* [Zitova 2003]. It is usually determined by the data acquisition process and shape variability among the objects being registered. Linear transformations include rigid motions, similarity transforms, affine transforms and perspective projections; these are global mappings defined by a small number of parameters (3, 4, 6, and 8, accordingly, in the planar case). Non-linear transformations can be explicitly parameterized, usually by tens to hundreds variables, each influencing a small region (examples include polynomial splines, radial basis functions and partition-of-unity). Another option is to derive the transformation from a physical model of the objects being registered, by defining external stretching forces, working towards improving local similarity, and internal resisting forces, minimizing amount of bending and stretching — and iterating towards the minimum energy state (popular examples of the models include those based on elastic rubber sheet and viscous fluid).

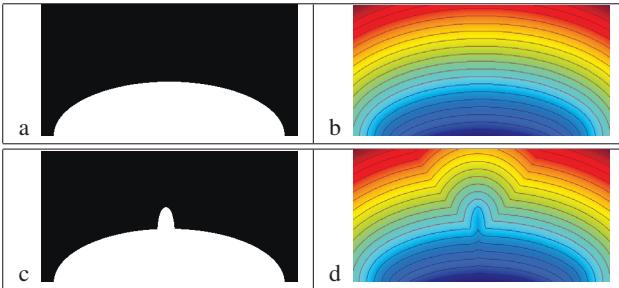
*Shape representation* is another important factor. Point clouds, spectral descriptors, parametrical curves, snake models, shock graphs and skeletons have all been extensively used in shape matching [Veltkamp 1999]. A more recent approach is to represent the figure's border implicitly as zero level-set of its distance field [Paragios 2003]. Distance field of a closed planar contour  $\gamma$  is a scalar function defined for any point  $x$  as follows:

$$DF(x; \gamma) = \sigma(x; \gamma) \cdot dist(x, \gamma); \quad (2)$$

for  $x$  inside  $\gamma$ ,  $\sigma(x; \gamma) = -1$ ; otherwise,  $\sigma(x; \gamma) = 1$ .



**Figure 2:** Distant fields are not robust to noise: left, source figure and its noisy counterpart; right, first components of the gradients of their distant fields.



**Figure 3:** A localized change in shape (a, c) can have a far-reaching effect on its distant field (b, d).

The main advantage of using distance fields for shape registration is that by discretizing the field on a rectangular support we can recast the problem as one of image registration, and draw upon the vast literature on the subject. Dissimilarity measure for such *distance images*<sup>1</sup> is usually the sum-of-squared-distances (SSD), or, in case of significant variation in shapes' scale, mutual information [Huang 2006] (correlation ratio was also discussed, but not implemented, in [Hong 2006]). Optimization methods for both linear and various nonlinear transformations have been proposed [Paragios 2003, Huang 2006, Liu 2011].

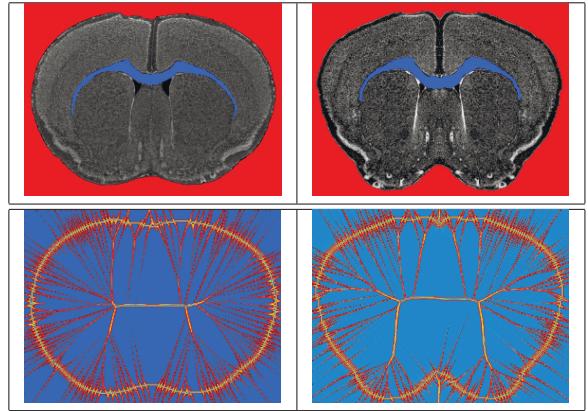
## 2 Analysis

Combining equations 1 and 2, we get the following expression for the transformation matching contours  $A$  and  $B$ :

$$T^{opt} = \underset{T \subset \Omega}{\operatorname{argmin}} (F_{dis}(T(DF(A)), DF(B)) + F_{sm}(T)). \quad (3)$$

This approach presents certain problems. First of all, distance fields are not robust to small noise-like perturbations [Hong 2006]. Figure 2 demonstrates this effect: given a simple figure (a), that defines a distance field with smooth gradients (b), we add only a modest amount of binary noise to the figure's contour (c), which results in significant degradation of the gradients (d). Effects of the noise

<sup>1</sup>We'll use  $DF(\gamma)$  to refer to the distance image of  $\gamma$ . The image's dimensions are defined by the application.



**Figure 4:** Two similarly shaped figures show pronounced differences in the structure of their distance fields (shown: magnitudes of gradients, contrast-enhanced).

are not local to the contour, effectively changing the distance field's structure.

What's more, you don't need to add distortions all over the contour to affect its distance field [Liu 2011]. As figure 3 illustrates, a small localized transformation of a contour can change values and smoothness of a sizable part of its distance field.

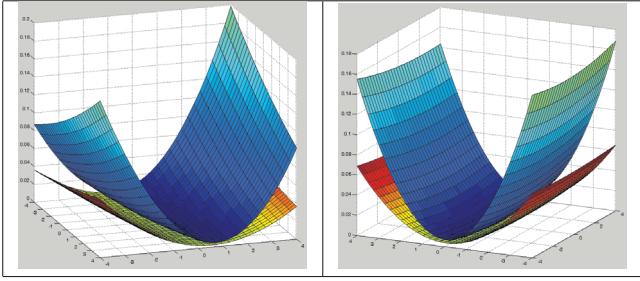
A more general fact, related the previous two, is as follows: two figures can have similar shape, but differently structured distance fields. In figure 4, this is demonstrated by comparing a pair of brain slices. Taken from two different brains, but in anatomically close positions, their shape is roughly equivalent. At the same time, their distance fields have markedly distinct features. No geometrical transformation can match these fields exactly. For shape registration, this can result in slower optimization and suboptimal fit.

Limitations of the approach can be summed up by a simple observation [Liu 2011]. For any transformation  $T$  acting upon contour  $A$  in a non-rigid way:

$$T(DF(A)) \neq DF(T(A)). \quad (4)$$

In other words, transformation of a shape's distance field is no more than an estimate of the distance field of the actual transformed shape. As a consequence, dissimilarity measure in equation 3 only approximates the actual dissimilarity, imposed by  $T$ .

At least three basic approaches have been employed by the research community to improve the method's robustness: limiting the calculation of the dissimilarity measure to narrow bands around the contours [Paragios 2003]; changing the shape representation to a generalized version of distance fields (integral kernels [Hong 2006], vector distance functions [Munim 2007]); changing the dissimilarity measure to a more robust one (mutual information [Huang 2006], variational chamfer-matching energy [Liu 2011]). Of these, only the latter work addresses property 4, albeit somewhat indirectly. Authors propose a symmetric variational measure, generalizing the "narrow bands" idea (eliminating the need for a heuristic to control the width of the bands). While their method does achieve very promising results, it uses complicated techniques for shape representation and optimization, making it much more difficult to implement than any other algorithm discussed here.



**Figure 5:** Comparing dissimilarity measure used in equation 3 (warm colours) to that of equation 5 (colder palette).

### 3 Proposed method

Considering the importance of property 4, discussed above, we would like to propose the following simple modification of procedure 3:

$$T^{opt} = \underset{T \in \Omega}{\operatorname{argmin}} (F_{dis} (DF(T(A)), DF(B)) + F_{sm}(T)). \quad (5)$$

While both alternative shape representations and dissimilarity measures, discussed above, could be incorporated into this algorithm, for space and clarity considerations, we shall only discuss the case of distance fields and SSD. When calculating the smoothness measure, we found it useful not to integrate over the whole image, but to exclude a narrow band around the contour. This usually results in better fit and improves convergence.

Although one could match distant field images using any image registration algorithm, most methods discussed above boost performance rate by introducing some kind of efficient approximation of the gradient of the dissimilarity measure, often allowing analytical derivation. In our case, though, transformation is incorporated into the measure in a highly non-linear and opaque fashion, which seems to preclude this kind of approach. On the other hand, derivative-free optimization methods remain either unreliable or difficult to tune for expensive functions in higher dimensions. So, we have to fall back to the use of finite differences to approximate the gradient.

Since every component of the gradient needs two evaluations of the dissimilarity measure, we need a transformation model that has as little parameters as possible, while still being relatively flexible. Ideally, each parameter should also influence only a small part of the image, allowing to group them in batches for simultaneous processing. Classical apparatus of cubic B-splines happens to be just such a model [Huang 2006]. Actually, it has the following neat property: gradient's computational complexity is virtually independent on the number of control points [Knott 2000]. Still, it takes 30+ function evaluations per 1 gradient evaluation. Considering the cost of the gradient computation, as well as the significant dimensionality of the search-space, we chose the limited-memory version of the Broyden–Fletcher–Goldfarb–Shanno algorithm (L-BFGS) as the optimizer, since it's particularly suited to this kind of problems [Byrd 1995]. For both L-BFGS optimization and B-spline transformations, we used the efficient implementations published by Dirk-Jan Kroon to accompany his PhD thesis [Kroon 2011].

Unlike in previous approaches, when calculating the distance field for a contour on a grid, we can not assume that the contour's points all come from the grid. Nor can we just project the points onto the grid before the calculation, as we need our measure to be sensitive

to small perturbations of the contour (it is crucial for the gradient approximation). Consequently, we can't use the fast image-based distance transform algorithms [Ciesielski 2011], and have to calculate distance field of a polygon instead, which is a very computationally-intensive procedure. In fact, we found that with a straightforward implementation, it takes over 95% of the calculation time. We address this problem by introducing an efficient GPU-based library for computing distance fields, based on NUDA (see section 4). In this, we follow [Voronin 2009], but use more advanced and up-to-date programming libraries and techniques. Note that we shall only discuss the magnitude of the distance field and not the sign ( $\sigma$  in equation 2), since its computation proceeds in a similar way and takes only a small fraction of the combined time.

To compare the two dissimilarity measures, classic  $F_{old} = F_{dis} (DF(A), DF(B))$  of equation 3 and proposed  $F_{new} = F_{dis} (DF(T(A)), DF(B))$  of equation 5, we conducted the following experiment. Using a sample mouse brain contour, 16x16 B-spline control point grid (512 variables), and 200x200 distance field discretization, and registering the contour upon a transformed version of itself, we studied the measures in vicinity of identity transform when all but two variables (pertaining to a single control point), are fixed. For each control point  $P_i$ , we went through 32x32 regularly spaced pairs of values for variables  $v_i^1$  and  $v_i^2$ , and calculated  $F_{new}$  and  $F_{old}$  for each of them. Choosing from the cases that exhibited significant variation, in figure 5 we show typical examples as surfaces (while these are all free of local extrema and even convex, this can not be expected in general case). Notably,  $F_{new} > F_{old}$  and  $\|\nabla F_{new}\| > \|\nabla F_{old}\|$  in all cases. That  $F_{new}$  shows steeper behaviour indicates that it is indeed a more sensitive measure of dissimilarity, and is also important for the convergence speed of the optimization procedure.

Using both measures, we've implemented a tool for brain slice alignment. You can see an example of its work in figure 1. While the difference between the results may not always be as pronounced as in this case, in our experience, the new measure consistently outperforms the old one and is more robust. With GPU-optimized distance field computations, coregistering 200x300 slices takes, depending on how different they are, 2-10 seconds.

### 4 GPU implementation

Nemerle is an extensible language, that is, a language which allows extending its syntax and semantics relatively easily. Nemerle accomplishes this with the help of macros, special functions which execute at compile-time as compiler plugins, perform code transformations and extend syntax of the language. Nemerle also provides code quoting facilities for concise creation of code trees from templates, and concise analysis of code trees passed as macro parameters. Nemerle macros are actually quite rich: any .NET method, either standard or user-written, can be called from inside a macro. Also, many features considered "built-in" in other languages, such as loops, conditionals, locks or asynchronous execution, are in fact implemented as macros in Nemerle.

NUDA (= Nemerle Unified Device Architecture) [Adinetz 2012, Adinetz 2011] is a set of Nemerle extensions for programming graphics processors. Extensions provided include multi-dimensional for-like loops, sending loops to execute on GPU, user-defined on-GPU functions and a set of loop transformations. Simple and structural types, as well as arrays, are supported for use in GPU kernels. libgpuvvm enables using of ordinary .NET arrays on GPU with little copying overhead through lazy synchronization and moving data to host with userspace pagefault handling. Special array types automatically synchronized between host and GPU are also provided, though this is currently not a preferred way of using

**Algorithm 1** Array addition in NUDA (for better readability, reserved words are in bold type, while the comments are italicized).

---

```
// allocate arrays
def a = array(n) : array[float];
def b = array(n) : array[float];
def c = array(n) : array[float];
// ... initialization ...
// array addition on GPU
nuwork(64) do(i in n)
  c[i] = a[i] + b[i];
// some work with array c
```

---

**Algorithm 2** Different versions of the distance field computation, top to bottom: baseline; optimized for NVidia Fermi; optimized for AMD (optimization-related annotations are underscored for better readability).

---

```
nuwork(128) do(i in m) {
  def p = ps[i];
  mutable d2 = 1e38f;
  do(j in n1)
    d2 = min(d2, dist2(es[j], p));
    df[i] = sqrt(d2);
} // baseline version

nuwork(128) dmine(4) do(i in m) {
  def p = ps[i];
  mutable d2 = 1e38f;
  unroll(2) do(j in n1)
    d2 = min(d2, dist2(es[j], p));
    df[i] = sqrt(d2);
} // version optimized for NVidia Fermi GPUs

nuwork(128) dmine(10) do(i in m) {
  def p = ps[i];
  mutable d2 = 1e38f;
  unroll(2) do(j in n1)
    d2 = min(d2, dist2(es[j], p));
    df[i] = sqrt(d2);
} // version optimized for AMD GPUs
```

---

data in GPU kernels.

NUDA makes GPU programming really straightforward, once the program consists of parallel loops processing arrays. For example, array addition, traditional GPU "hello world!", is really straightforward, see Algorithm 1. In fact, **do**(...) is a loop, which can be multi-dimensional, and **nuwork**(64) is an annotation which sends the loop to which it is applied to GPU; here, 64 is the thread block size, and the total number of threads is derived from the number of iterations on the GPU. The implementation of array addition on GPU above is, in fact, near-optimal, as the problem is memory-bound, and the implementation almost saturates memory bandwidth. For other problems, however, such naïve implementations are far from optimal. In these cases, annotations performing loop transformations can be applied to optimize the code. Annotations include full loop unroll (**inline**), standard unroll (**unroll**), deep loop unroll (**dmine**), caching data in local memory (**ulocal**), loop tiling as well as other transformations.

Distance field computation is the problem of computing distances from a set (an array) of points to the border of a polygon, one distance per point. In the basic case, for each point, the distance is the minimum of distances from the polygon to each of the points in the set. There are two main approaches to distance field computation. The first one is *all-to-all* approach, in which for each point, distance to each of the edges is computed, and minimum dis-

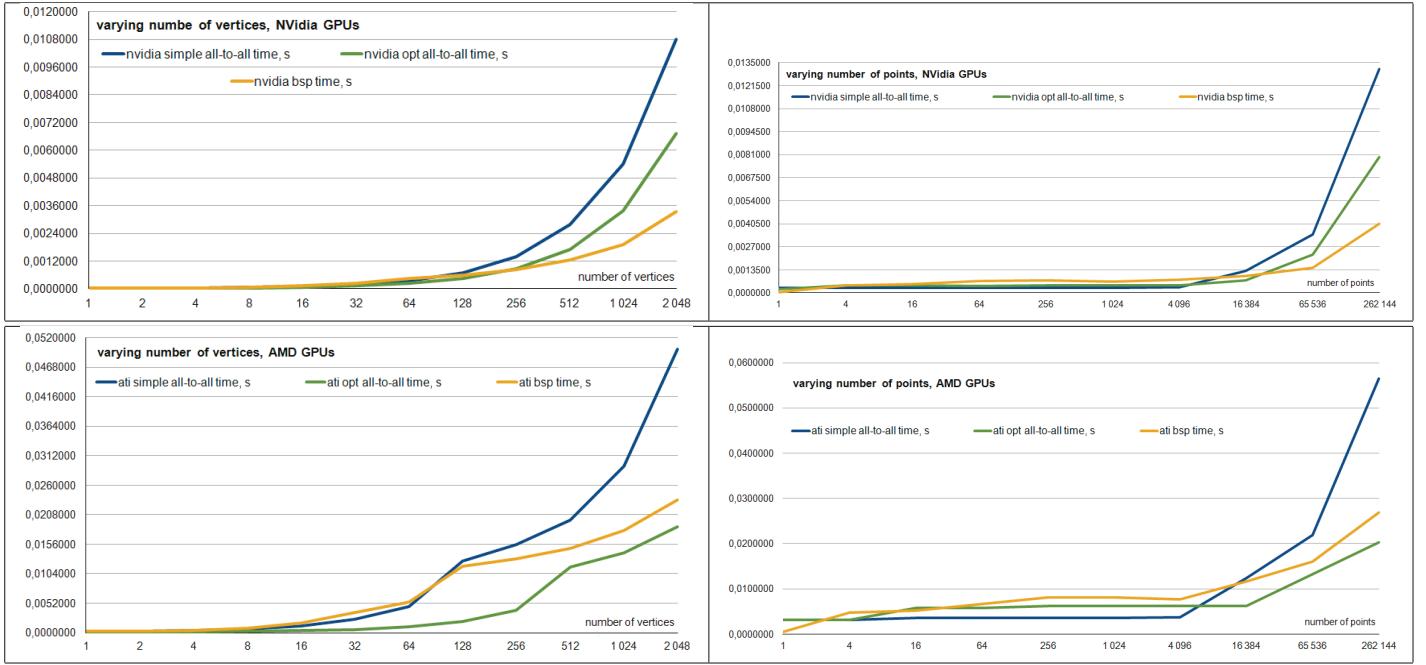
tance is found. The second approach is to use an HBV (hierarchy of bounding volumes), e.g. BSP trees. While the second approach is asymptotically better, it is not clear outright which one is faster, since all-to-all approach allows more aggressive optimization and is more cache friendly.

For our testing, we implemented both approaches on CPU and GPU. CPU implementation was done using C++ with OpenMP and SSE intrinsics, while GPU variant was implemented using Nemerle and NUDA. For all-to-all approach, two versions are compared: the trivial version and the optimized version. Here, optimization was done using NUDA annotations without compromising readability, see Algorithm 2. The basic GPU code version, together with annotations used for optimization, is presented below. **dmine**(k) annotation does deep loop unrolling. This means creating k copies of the loop and zipping them together, including inner loop. It can also be thought of as computing k elements in a single GPU thread at the same time. For AMD, this allows benefiting from its 5-way VLIW instructions, and it also results in less global memory bandwidth for both architectures. **unroll**(k) does simple k-sized loop unrolling, and reduces looping overhead for the inner loop on NVidia GPUs.

For the second approach, we implemented HBV-assisted distance computation on GPU. We first construct a BSP tree using fixed-point (at half size) recursive space subdivision, and switching subdivision axis for subsequent subdivisions. We stop when either the maximum depth or minimum number of edges is reached, or when the overhead resulting from the subsequent subdivision is higher than a certain threshold. After that, we transform the BSP into HBV node-wise from the bottom up, by recomputing the bounding volume based on all edges belonging to a specific node. For distance computation, we use a traditional recursive algorithm. We first compute the distance from the point to both subvolumes in HBV, and then go into the nearest one. Once we have the upper estimate for the nearest distance, we prune all subvolumes with greater distance. Since not all current GPUs support recursion, we've rewritten the algorithm, first as a stack-based traversal, and then as a stack-less traversal by adding pointer-to-parent to each GPU HBV node. HBV traversal is obviously less regular than all-to-all computation, so none of the optimizations described above is used. Each GPU thread traverses HBV for one point only.

We tested the algorithms we implemented on two GPUs. For both approaches, all computations were done in single precision. The first one is NVidia Tesla C2050 (nvidia) with 3 GB of GPU RAM and 1030 GFlop/s single-precision peak performance. The second one is ATI HD Radeon 5830 (ati) with 256 MB of GPU RAM 1792 GFlop/s single-precision peak. We performed two series of experiments. Both involved computing a distance field from a uniform 2D  $m \times m$ -sized grid of points to a regular polygon centered at  $0.5 * m$  inscribed into a circle with radius  $0.4 * m$ . In the first series of experiments, the number of points was fixed at 102400, and the number of vertices varied (exponentially) from 1 to 2048. In the second series, the number of vertices was fixed at 1000, while the number of points varied from 1 to 262144, also exponentially. In both cases, we measured time it took to compute the distance field for simple all-to-all, optimized all-to-all and HBV approaches. We also computed accelerations of optimized vs. baseline and HBV vs. all-to-all baseline approaches. We shall now discuss the results, presented in figures 6 and 7.

In all cases, the growth of time with the number of vertices seems to be linear. For all-to-all approach, this must be the case, while for HBV, this means that even 2000 vertices is too few to reach logarithmic growth. When the number of points varies but stays below 2K to 4K, time varies little with the number of points, as GPU is under-saturated. After that, the expected linear growth can be observed.



**Figure 6:** Average distance field computation time for varying number of vertices, left, and points right. Top, NVidia; bottom, ATI.

Obviously, for all-to-all approach, the computation time does not depend on the configuration of point field and the polygon. For HBV approach, however, time depends on the configuration of both. For ATI GPUs, HBV is always worse than optimized all-to-all for the uniform point grid. For NVidia, obviously, the acceleration increases with the number of vertices, reaching about 2 times for 2K vertices. We also tested points uniformly distributed on a circle, with the circle being close to the original polygon. Here, HBV approach can be an order of magnitude faster than optimized approach. Overall, HBV gives better acceleration for NVidia GPUs than for ATI, since NVidia Fermi GPUs have cache, and are therefore better at handling random memory accesses.

We also compared optimized all-to-all computation to simple computation. Starting with a reasonable number of points ( $> 4K$ ) and almost for any number of vertices ( $> 4$ ) the version optimized with annotations described above is always faster than the baseline version, with acceleration increasing with the number of points, and stopping at near  $2\times$  for both NVidia and ATI GPUs. Overall, the provided optimizations allow us to utilize GPUs efficiently, reaching about 54% peak for NVidia GPUs, and 31% peak for ATI GPUs. The times above do not include time spent on copying data (though we found it to be negligible) and on building HBV (non-negligible). Therefore, all-to-all approach can be significantly faster if the distance field is computed for each polygon only once, since it avoids the costly pre-processing step.

## 5 Conclusion

We analyzed the robustness problems of the existing distance field-based shape matching algorithms, isolating what we believe is a core issue, and proposed a new dissimilarity measure, explicitly designed to overcome it. We also presented an efficient NUDA-based GPU implementation of the distance field calculation, which makes the proposed approach computationally feasible. This is an ongoing project, with work underway to extend the approach to 3d.

## Acknowledgements

This work was supported by the RFBR grant 11-04-12174-ofi-m-2011. The first author was partially supported by the contract No 07.514.12.4030. The second author was supported by CUDA Center of Excellence in the Lomonosov Moscow State University. The third author was partially supported by the contract No 13.G36.31.0002.

## References

- [**Adinetz 2011**] A.V. Adinetz. Extensible languages for GPU programming. In Proceedings of PaVT-2011. (in Russian)
- [**Adinetz 2012**] A.V. Adinetz. Extran and NUDA Programmer's Guide. June 2012. <http://sourceforge.net/projects/nuda/files/0.0.6/extran-guide.pdf/download>
- [**Babalola 2006**] K. Babalola, T. Cootes. Registering richly labelled 3d images. In: Proc. of the Int. Symp. on Biomedical Images (2006)
- [**Baran 2009**] I. Baran, D. Vlasic, E. Grinspun, and J. Popović. Semantic deformation transfer. Proceedings of SIGGRAPH (2009), article no 36.
- [**Belongie 2001**] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. The 8th ICCV, Vancouver, Canada, pages 454-461, 2001.
- [**Byrd 1995**] R.H. Byrd, P. Lu, J. Nocedal, C.Y. Zhu. A Limited Memory Algorithm for Bound Constrained Optimization. SIAM Journal on Scientific Computing, Vol. 16, No. 6. (1995), pp. 1190-1208.
- [**Ciesielski 2011**] K.C. Ciesielski, X. Chen, J.K. Udupa and G.J. Grevera. Linear Time Algorithms for Exact Distance Transform. Journal of Mathematical Imaging and Vision, vol. 39, no. 3 (2011), 193-209.

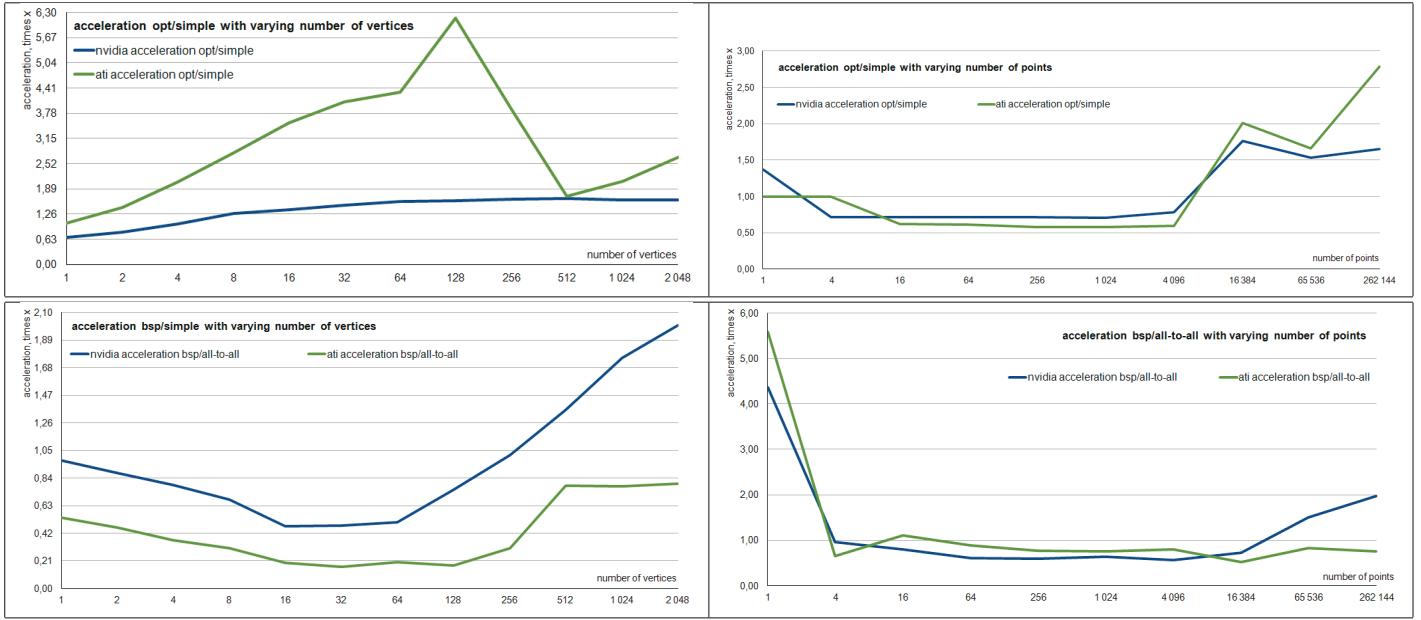


Figure 7: Acceleration for GPU-specific optimization, top, and BSP-trees, bottom, for varying number of vertices, left, and point, right.

**[Fahmi 2008]** R. Fahmi and A. A. Farag. A Novel Shape Registration Framework and Its Application to 3D Face Recognition in the Presence of Expressions. Lecture Notes in Computer Science, 2008, Volume 5359/2008, 287-296.

**[Ginneken 2001]** B. Ginneken, B. Romeny, and M. Viergever. Computer-Aided Diagnosis in Chest Radiography: A Survey. IEEE Transactions on Medical Imaging, Vol. 20, 2001.

**[Hong 2006]** B.-W. Hong, E. Prados, S. Soatto, L. Vese. Shape Representation based on Integral Kernels: Application to Image Matching and Segmentation. IEEE Conference on Computer Vision and Pattern Recognition 1 (2006) 833- 840.

**[Huang 2006]** X. Huang, N. Paragios, D.N. Metaxas. Shape Registration in Implicit Spaces Using Information Theory and Free Form Deformations. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 8, Aug. 2006.

**[Kilian 2007]** M. Kilian, N.J. Mitra, and H. Pottman. Geometric modeling in shape space. Proceedings of SIGGRAPH (2007), volume 26, pp. 64:1-64:8.

**[Knott 2000]** G.D. Knott. Interpolating Cubic Splines. Springer, 2000.

**[Kroon 2011]** D.J. Kroon. Segmentation of the Mandibular Canal in Cone-Beam CT Data. PhD thesis, University of Twente. 2011.

**[Levoy 2000]** M. Levoy, S. Rusinkiewicz, M. Ginzton, J. Ginsberg, K. Pulli, D. Koller, S. Anderson, J. Shade, L. Pereira, J. Davis, and D. Fulk. The Digital Michelangelo project: 3d scanning of large statues. Proceedings of SIGGRAPH (2000), pp. 131-144.

**[Liu 2011]** W. Liu and E. Ribeiro. A Meshless Method for Variational Nonrigid 2-D Shape Registration. Proceeding of ISVC 10, vol II, pp. 262-272.

**[Maintz 1998]** J.B.A. Maintz, , M.A. Viergever. A survey of medical image registration. Medical Image Analysis 2 (1998) 1-36.

**[Malcolm 2008]** J. Malcolm, Y. Rathi, and A. Tannenbaum. Label Space: A Multi-object Shape Representation. Lecture Notes in

Computer Science, 2008, Volume 4958/2008, 185-196.

**[Munim 2007]** H.A.E. Munim, A.A. Farag. Shape representation and registration using vector distance functions. Proceedings of CVPR 2007.

**[Paragios 2003]** N. Paragios, M. Rousson, and V. Ramesh. Non-rigid registration using distance functions. Computer Vision and Image Understanding, 89(2-3):142{165, 2003.

**[St-Jean 1998]** P. St-Jean, A.F. Sadikot, L. Collins, D. Clonda, R. Kasrai, A.C. Evans, and T.M. Peters. Automated Atlas Integration and Interactive ThreeDimensional Visualization Tools for Planning and Guidance in Functional Neurosurgery. IEEE Transactions on Medical Imaging, Oct. 1998.

**[Tsai 2003]** A. Tsai, A. Yezzi, W. Wells, C. Tempany, D. Tucker, A. Fan, W. Grimson, A. Willsky. A shape-based approach to the segmentation of medical imagery using level sets. Transactions on Medical Imaging 22(2), 137-154 (2003).

**[Veltkamp 1999]** R.C. Veltkamp and M. Hagedoorn. State of the art in shape matching. Technical Report UU-CS-1999-27, University of Utrecht, 1999.

**[Voroin 2009]** P. Voronin, A.V. Adinetz. On computing distance fields of planar polygons. In proceedings of GraphiCon-2009. (in Russian)

**[Zitova 2003]** B. Zitova and J. Flusser. Image registration methods: a survey. Image and Vision Computing, 21 (11) (2003) 977-1000.

**[Zhou 2005]** X.S. Zhou, A. Gupta, and D. Comaniciu. An information fusion framework for robust shape tracking . IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 1, 2005.

**[Zhu 1996]** S.-C. Zhu and A. Yuille. Forms: a flexible object recognition and modeling system. IJCV, 20:187-212, 1996.



## Technical section (Russian)

---

GraphiCon'2012

October 01–05, 2012  
Moscow, Russia

# Simile Classifiers for Face Classification

Vadim Konushin, Tatiana Lukina, Artem Kuharenko, Anton Konushin  
 Lomonosov Moscow State University  
 {vadim, tlukina, akuharenko, ktosh}@graphics.cs.msu.ru

## Abstract

We present a new approach to face classification using simile classifiers. Unlike other methods we explicitly estimate similarity distances to the known reference people and use these similarities as high-level features for the classification of the test face.

We test our algorithm on gender classification problem. Our algorithm shows classification accuracy of 92.96% on LFW dataset.

**Keywords:** attribute classification, classifier training, gender classification

## 1. INTRODUCTION

Attribute classification from face images is a widely studied problem. Range of attributes includes gender, ethnicity, age, hair style, presence of glasses and so on. Attribute classification is used for content-based image retrieval in web, video surveillance, audience measurement and other applications.

A common approach to face classification is to extract some kind of low-level features from image and then use a machine learning method to find dependencies of the classified attribute on the features. Such methods do not search for similar faces or compare a test face to a set of training faces directly. If SVM classifier is used, a number of training faces are selected as support vectors and used implicitly. In this case the classification results depend on distances to support vectors (in fact, training faces), but the distances are defined by SVM kernels and may correspond poorly to real face similarity.

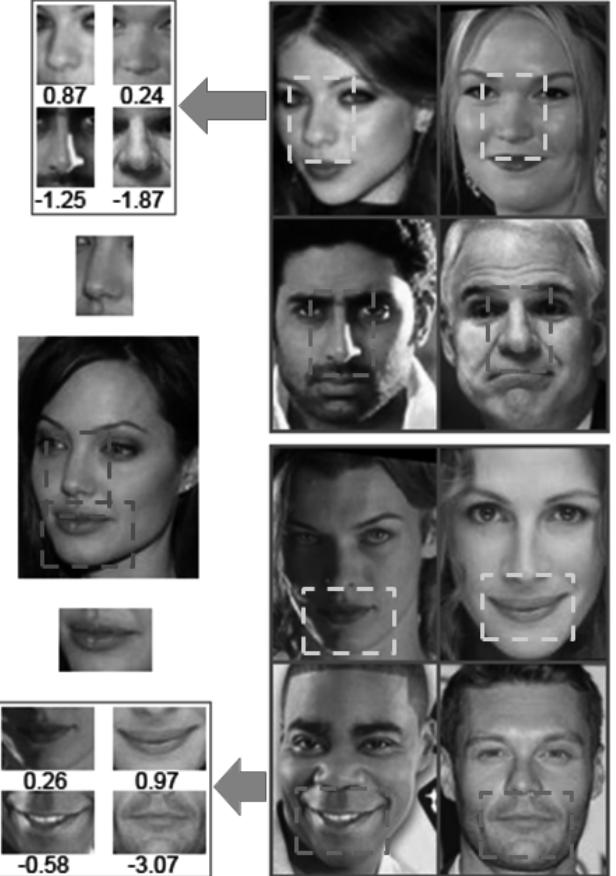
We propose to explicitly train similarity classifiers to compute distances to the selected set of people and then use these distances as input to a second classifier, which gives the final result. Our work was inspired by the face recognition method [10], where it was proposed to compare people faces based on their attributes (like age, gender, nose size, etc.) and similarities to other “reference” people. For this task they trained so called simile classifiers for individual face components, like eyes or lips.

We use a similar approach, but for the task of face classification. Figure 1 shows an algorithm illustration. We test our algorithm on the gender classification problem and show 92.96% accuracy on LFW dataset [8].

## 2. RELATED WORK

Most face classification algorithms follow the same pipeline, which consists of face normalization, features extraction and classification. The difference between algorithms is mainly in specific methods, used in these steps.

A common choice for face normalization is consecutive image rotation, scaling and cropping, so that eyes are placed into specific pixels. As features researchers have tried raw pixel intensities [1, 13], Haar-like features [12], Active Appearance



**Figure 1.** Illustration of our algorithm applied to a gender classification problem. Components of a test face are compared with corresponding components of the reference people. Classification decision is made based on similarities to these people and their gender.

Model parameters [2, 6, 12], Local Binary Patterns [11, 17] and Biologically Inspired Features [7, 15], which are based on Gabor functions.

For the classification step boosting [1, 12, 17], SVM [2, 6, 9, 10, 11, 12, 13, 15, 17] and Random Forest [14] have been used, with SVM being the most popular choice.

Several papers [4, 7] investigated an influence of one attribute on another, for example preliminary gender classification on age estimation [7] or ethnicity classification on gender classification [4].

We are not aware of any face classification method, which explicitly searches for similar faces and makes a decision, based on these similarities. Algorithms, which use SVM, do this implicitly by distances to the support vectors, but the used distance metrics are defined by the SVM kernels and may be a bad approximation to the real face similarities.

The most similar method to ours is a face recognition algorithm of [10]. Face recognition algorithms are in general very close to face classification algorithms and often use the

same normalization and features. In [10] for face verification it was proposed to use similarities to some other known people as features. For this task special ‘simile’ classifiers are learned, which, for example, could estimate “how similar are these lips to Angelina Jolie’s lips” or “how similar is this nose to the nose of Brad Pitt’s”.

One another somewhat related method to ours is a recent face recognition algorithm of [19]. For rotated test faces they search for the most similar components of known reference people. During face comparison these components are switched to the components of the found reference people, but in a frontal view. By this trick the algorithm becomes more robust to face out-of-plane rotation, which is possibly the main problem of current face recognition algorithms.

We use a similar approach to face classification. We measure similarity distances of a test person to the set of known people and use their known attribute labels for the classification.

### 3. PROPOSED ALGORITHM

Our algorithm consists of 2 steps:

- Apply simile classifiers to the test face components and find similarity distances to all reference people
- Use computed similarities as input to the final classifier

#### 3.1 Simile Classifiers

We follow approximately the same approach to simile classifiers training as in the original paper [10]. Some details can be different due to somewhat brief description of implementation in [10].

At first a face is detected by a Viola-Jones face detector [18]. Then we find fiducial points, including eye and lips corners by our own implementation of [3]. Then we select face components, as illustrated on Figure 2. For each component we train separate simile classifiers.

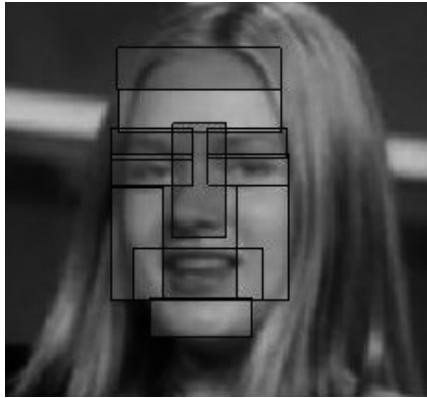


Figure 2. Used components illustration.

For simile classifiers a number of different low-level features are used. Features are constructed by choosing a pixel value type, normalization and the aggregation type. For pixel value types we use colors in RGB and HSV color spaces, intensities, edge magnitudes and orientation. Normalization types include

mean normalization  $\hat{x} = \frac{x}{\mu}$  or energy normalization  $\hat{x} = \frac{x - \mu}{\sigma}$ ,

where  $x$  refers to the input value,  $\mu$  and  $\sigma$  - to the mean and standard deviation of  $x$ ,  $\hat{x}$  is the normalized output value. Aggregation types include a histogram or the mean and variance. All these feature types are summarized in the Table 1.

RGB	None	None
HSV	Mean-Normalization	Histogram
Image Intensity	Energy-Normalization	Statistics
Edge Magnitude		
Edge Orientation		
Pixel Value Types	Normalizations	Aggregations

Table 1. Feature types used for simile classifiers.

For each component and for each reference person we train a simile classifier, which recognizes similarity to this component of this specific person. For the positive sample we take the cropped component from different images of this person, and for the negative sample we take the same component images taken from random images of other persons in the training set.

We train a separate SVM classifier with RBF kernel for each feature type, component and reference person.

#### 3.2 Face Classification using Simile Classifiers

At this step we have a  $N \cdot M \cdot K$  dimensional descriptor  $F$  for a test face, where  $N$  stands for the reference people number,  $M$  stands for the feature type number and  $K$  is a number of components. Each  $F_{n,m,k}$  is a similarity between  $k^{th}$  components of a test person and  $n^{th}$  reference person using  $m^{th}$  feature type and in fact is an output of a corresponding SVM classifier.

We consider only the case of binary face classifications. Multi-class classification problems could be reduced to binary problems by one-against-one or one-against-all strategy. In [2] it was proposed to reduce age estimation problem to series of binary classifications, which answer the question “Is a test person older, than given age?” This strategy shows one of the top age estimation accuracies so far.

Let  $l_n \in \{-1, 1\}$  be a binary label of a reference person  $n$ . We need to estimate binary label of a test face  $L(F)$

We have tested several classification algorithms, listed below. The resulting accuracies are given in the Experiments section.

- The most simple option is to use a nearest neighbor approach – to take a label of the most similar person considering only 1 component  $k$  and 1 feature type  $m$ :

$$n'(m, k) = \arg \max_n (F_{n, m, k})$$

$$L(F) = l_{n'(m, k)}$$

- To use information from all components and features types we can combine all  $K \cdot M$  values  $F_{n'(m, k), m, k}$  in a single feature vector and feed it to a single SVM classifier with RBF kernel. To account for values of binary labels  $l_{n'(m, k)}$ , we switch sign of those  $F_{n'(m, k), m, k}$ , which correspond to reference people with a label  $l_{n'(m, k)} = -1$ . In our experiments  $F_{n'(m, k), m, k}$  was positive for all  $k, m$ , so this switch introduces information about label values into feature vector.
- A logical extension is a  $k$ -nearest neighbor approach. In this case we find  $p$  most similar reference persons for each component and each feature type, concatenate  $p \cdot K \cdot M$  corresponding  $F_{n, m, k}$  elements (again with sign switching), and input them into a SVM classifier.

- Another approach would be just to sum all  $F_{n,m,k}$  elements. We multiply on  $-1$  those  $F_{n,m,k}$ , for which corresponding  $l_n = -1$ , and keep other  $F_{n,m,k}$  unchanged.

Note, that in this case some of  $F_{n,m,k}$ , as SVM outputs, are already negative. So we treat dissimilarity to a reference person as a similarity to a person with the opposite label.

$$L(F) = \text{sign} \left( \sum_{n,m,k} F_{n,m,k} + C \right),$$

where  $C$  is a parameter.

The more similar are the reference people with label  $l$  to the test face, the greater the chance that the test face has the same label  $l$ .

However  $F_{n,m,k}$  for different feature types usually have different variations and in  $\sum_{n,m,k} F_{n,m,k}$  only some feature types would matter.

So we summed  $F_{n,m,k}$  only with the same feature type, and combined results for different  $m$  with AdaBoost or SVM classifier:

$$L(F) = \text{classifier} \left( \sum_{n,k} F_{n,1,k}, \sum_{n,k} F_{n,2,k}, \dots, \sum_{n,k} F_{n,M,k} \right)$$

- Instead of a simple summing we could use a separate classifier for each feature type. So we have a two-layer classifier:  $M$  different SVM classifiers, that take concatenated  $N \cdot K$  values  $F_{n,m_i,k}$ , and one SVM classifier, that takes  $M$  outputs of the previous classifiers.
- We also test another variant of two-layer classifier. First, we train  $K$  different SVM classifiers for each component, where each classifier uses  $N \cdot M$  values  $F_{n,m,k_i}$ . Then we train one SVM on top of these  $K$  outputs.

It is important to note, that in the last two cases, where we use only classifiers, we don't need reference people labels, because an order of reference people in feature vector is fixed. Classifiers automatically learn dependencies, like "similarity to this reference person means, that the test face is more likely to have label 1, and similarity to that reference person probably means, that it has label -1".

## 4. EXPERIMENTS

As an example of a face classification problem we chose a gender classification because it is a well-studied problem and, unlike age estimation, it is relatively easy to prepare ground truth data.

Unfortunately, unlike face recognition or to some extent age classification, there is no standard test dataset for gender classification problem. In different papers a subset of FERET [16], CAS-PEAL face database [5], FaceTracer [9] and LFW databases were used.

A natural choice for our experiments was to use a face image dataset PubFig [10] and LFW [8] database, as they were used in the original paper [10], where simile classifiers were proposed.



Figure 3. Example images from LFW dataset.



Figure 4. Example images from PubFig dataset.

LFW database consists of 13233 images of 5749 different persons, all images are collected from the web, and the only condition on image is a detected face by Viola-Jones face detector. Nowadays LFW is a standard database for face recognition algorithms comparison.

PubFig is a complement to LFW database, proposed in [10]. It consists of 60000 images of 200 people. The important property of PubFig is that there are at least 50 different images for each person and therefore it is well suited to be a reference people dataset (as there are enough images for simile classifiers).

Example images of LFW and PubFig are shown in Figure 3 and 4. We have manually annotated gender in both PubFig and LFW datasets.

In our experiments we used PubFig Development Set as reference people set (59 people, originally there were 60 people in this set, but the database is uploaded as a list of URLs and some links are currently missing) and 5-fold cross-validation on LFW.

For comparison with existing gender classification methods we selected two algorithms, with testing procedure closest to our:

- In [10] a gender classification along with other attributes classification are used for further face recognition. Exact training and test subsets are not described, but it seems that some parts of PubFig and LFW were used. An accuracy of 81.22% was reported.
- In [17] a 5-fold cross-validation on a subset of LFW was used for the experiments. The faces, that are not near frontal and those, for which it is difficult to establish the ground truth, were not considered. This resulted in 7443 out of 13233 images. The resulting classification accuracy was impressive 94.81%. Authors promised to make this dataset publicly available, but to our knowledge haven't done it yet.

Our results are summarized in Table 2.

Nearest neighbor approach performs poorly: 67% with 1 feature type and 1 component and 77% when combined by one SVM. When we switch it to  $k$ -nearest neighbors approach we gradually get better results with increasing  $k$ .

But even simple summing of all the similarities  $F_{n,m,k}$  gives equal result. Note that summing  $F_{n,m,k}$  for only one feature type can result in 84%. Applying classifier to these sum values

improves the accuracy up to 89%. SVM with RBF kernel performs better than AdaBoost.

The best approach is to use a two-layer classifier. If on the first level we use separate SVM for each feature type, we get 91.6% (90.8% by a single SVM classifier for the best feature). If we use separate SVM classifier for each component, and then combine them by another SVM, we get our best 92.96%.

It is significantly better than 81.2% of [10], where the same kind of low-level features were used. Though, as we don't know the exact training/testing samples of [10], it could be a result of a smaller training sample size.

We lose to 94.81% of [17], but we have used a more complex test sample (though maybe there is more sense in discarding strongly rotated photos) and possibly less discriminating low-level features.

Algorithm	Classification accuracy
Nearest neighbor, only 1 best combination of feature and component types	67%
Nearest neighbor and SVM	77%
K-nearest neighbors, $k = 2$ , SVM	80.5%
K-nearest neighbors, $k = 3$ , SVM	81.6%
K-nearest neighbors, $k = 4$ , SVM	82%
Summing all $F_{n,m,k}$	82%
Summing all $F_{n,m,k}$ only for one (best) feature type	84%
Summing all $F_{n,m,k}$ with the same feature type, boosting, 10 iterations	86%
Summing all $F_{n,m,k}$ with the same feature type, boosting, 50 iterations	87.5%
Summing all $F_{n,m,k}$ with the same feature type, SVM	89%
SVM over $N \cdot K$ elements $F_{n,m_i,k}$ with the same feature type (best one)	90.8%
Combination of $M$ SVMs over $N \cdot K$ elements $F_{n,m_i,k}$ with the same feature type by one SVM	91.6%
Combination of $K$ SVMs over $N \cdot M$ elements $F_{n,m,k_i}$ with the same component by one SVM	<b>92.96%</b>

**Table 2.** Classification accuracy results by different proposed algorithms.

## 5. CONCLUSION

We have presented a new approach to face classification. We use a reference people dataset and train simile classifiers, which estimate the similarity between a test and the reference person.

We have tested several algorithms that use such similarity distances to the reference people and their labels for the final classification. Best result was demonstrated by a two-layer classifier, which uses only the similarities, with no information about reference people labels.

We demonstrate the effectiveness of the proposed algorithm on the task of gender classification, where we achieve 92.96% accuracy on LFW dataset.

## 6. AKNOLEDGMENTS

This work was supported by Russian Foundation for Basic Research, grant N. 11-01-00957-a.

## 7. REFERENCES

- [1] S. Baluja and H. A. Rowley. Boosting sex identification performance. *Int. J. of Comput. Vision*, 71(1): 111–119, 2007.
- [2] K.-Y. Chang, C.-S. Chen, and Y.-P. Hung. Ordinal Hyperplanes Ranker with Cost Sensitivities for Age Estimation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2011.
- [3] D. Cristinacce and T. Cootes. Boosted Regression Active Shpae Models. *British Machine Vision Conference*, 2007.
- [4] P. Du and D. Xiaoqing. The Application of Decision Tree in Gender Classification. *Congress on Image and Signal Processing*, 2008
- [5] W. Gao, B. Cao, S. G. Shan, et al. The CAS-PEAL Large-Scale Chinese Face Database and Baseline Evaluations, technical report of JDL, available on [http://www.jdl.ac.cn/~peal/peal\\_tr.pdf](http://www.jdl.ac.cn/~peal/peal_tr.pdf), 2004
- [6] G. Guo, Y. Fu, T. S. Huang, and C. Dyer. A probabilistic fusion approach to human age prediction. In *IEEE CVPRSLAM workshop*, 2008.
- [7] G. Guo, G. Mu, Y. Fu, C. Dyer, and T. S. Huang. A Study on Automatic Age Estimation using a Large Database. *ICCV*, 2009
- [8] G. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled Faces in the Wild: A database for studying face recognition in unconstrained environments. *UMass Amherst Technical Report 07-49*, October 2007.
- [9] N. Kumar, P. N. Belhumeur, and S. K. Nayar. FaceTracer: A search engine for large collections of images with faces. *ECCV*, 2008.
- [10] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. *ICCV*, 2009.
- [11] H. C. Lian and B. L. Lu. Multi-view gender classification using local binary patterns and support vector machines. *Proceedings of the 5th International Conference on Artificial Neural Networks (ISNN '06)*, 2: 202–209, Chengdu, China, 2006.
- [12] E. Makinen and R. Raisamo. Evaluation of gender classification methods with automatically detected and aligned faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30: 541–547, 2008
- [13] B. Moghaddam and M. Yang. Learning gender with support faces. *IEEE Trans. Pattern Anal. Machine Intell.* 24(5): 707–711, 2002
- [14] A. Montillo and H. Ling. Age Regression from Faces Using Random Forests. *IEEE International Conference on Image Processing*, 2009
- [15] G. W. Mu, G. D. Guo, Y. Fu, and T. S. Huang. Human age estimation using bio-inspired features. *Proc. of Computer Vision and Pattern Recognition*, 2009

- [16] P. J. Phillips, H. Moon, S. A. Rizvi, and P. Rauss. The FERET Evaluation Methodology for Face Recognition Algorithms. *IEEE PAMI*, 22(10): 1090-1104, 2000.
- [17] C. Shan. Learning local binary patterns for gender classification on real-world face images. *Pattern Recognition Letters*, 33(4): 431-437, 2012.
- [18] P. Viola and M. J. Jones. Robust Real-Time Face Detection. *Int'l J. Computer Vision*, 57(2): 137-154, 2004
- [19] Q. Yin, X. Tang, and J. Sun. An associate-predict model for face recognition. In *CVPR*, 2011

# Метод параллельных цепей для распознавания дактилоскопических изображений

Гудков Владимир Юрьевич  
Челябинский Государственный Университет, Челябинск, Россия  
[Diana@Sonda.ru](mailto:Diana@Sonda.ru)

## Аннотация

В работе предлагается метод измерения направлений светотеней линий дактилоскопических изображений, устойчивый к изменению ширины линий и просветов, разрывам линий и другим дефектам изображения. Результатом применения метода является поле направлений, снабженное достоверностью и отражающее ориентацию двух параллельных теневых или световых линий.

**Ключевые слова:** Отпечаток пальца, поле направлений, светотени, параллельные цепи.

## 1. ВВЕДЕНИЕ

В компьютеризированных системах (КС) идентификацию изображений, как правило, выполняют после обработки изображений [1-3, 5]. При этом ошибки обработки напрямую влияют на величины ошибок идентификации [6]. Вследствие этого ошибки обработки изображения необходимо минимизировать.

При распознавании дактилоскопических изображений (ДИ) выделяют признаки, формирующие структуру самого ДИ. К таким основным признакам, согласно [7], относят: поле направлений линий, поле плотности линий и поле качества линий в виде матриц, размерность которых определяется сегментацией ДИ. В ячейку каждой такой матрицы записывают локально вычисленное значение признака (яркость, контрастность, градиент) [5]. Это случайная функция двух аргументов (координат), значение которой неизвестно до исхода эксперимента. Очевидно, что случайные поля [4], генерируемые обработкой, определяются распределением яркостей точек ДИ.

На рис. 1 представлено ДИ с дефектами, увеличивающими ошибки идентификации [6]. Это вынуждает разработчиков КС применять новые методы распознавания [2].



Рис. 1: Изображение отпечатка пальца

## 2. ПОСТАНОВКА ЗАДАЧИ

Изображение как множество действительных чисел формируют в виде  $F = \{f(x, y) | (x, y) \in X \times Y\}$  в прямоугольной области  $G$  мощностью  $|G| = x_0 y_0$ , где  $X = 0..x_0 - 1$  и  $Y = 0..y_0 - 1$ . Структурно обработку изображения представляют в виде пирамиды  $\mathcal{K}$  слоев (препараторов изображения) из взаимосвязанных иерархий [5]. Сегментация  $l$ -го слоя  $k$ -й иерархии  $F_k^{(l)}$  разбивает слой на  $x_h y_h$  непересекающихся квадратных сегментов  $S_{hk}^{(l)}(x, y)$  с длиной стороны  $2^{h-k}$  и вершинами  $(x, y) \in X_h \times Y_h$ , где  $k < h$  и  $h$  – номер иерархии;  $X_h = 0..x_h - 1$  и  $Y_h = 0..y_h - 1$ . Доступ к каждой точке сегмента  $S_{hk}^{(l)}(x, y)$  записывают в координатах  $(u, v) \in \bar{X}_{hk} \times \bar{Y}_{hk}$ :

$$\begin{cases} \bar{X}_{hk} = \{u + x2^{h-k} | x \in X_h \wedge u \in 0..2^{h-k} - 1\}, \\ \bar{Y}_{hk} = \{v + y2^{h-k} | y \in Y_h \wedge v \in 0..2^{h-k} - 1\}, \end{cases} \quad (1)$$

а центры сегментов  $(u, v) \in \hat{X}_h \times \hat{Y}_h$  находят в виде

$$\begin{cases} \hat{X}_h = \{2^{h-1} + x2^h | x \in X_h\}, \\ \hat{Y}_h = \{2^{h-1} + y2^h | y \in Y_h\}. \end{cases} \quad (2)$$

Для формализации методов классификационного анализа (КА) применяют аппарат апертур. Прямолинейные щелевые  $A_h(x, y, \alpha, w)$  и точечные  $A_h^0(x, y, \alpha, w)$  апертуры как множества точек и углов в виде элементов упорядоченных троек  $(u, v, \beta)$  определяют по формулам:

$$\begin{cases} A_h(x, y, \alpha, w) = \{(u, v, \beta) = (x + w_x, y + w_y, \beta) | w \in Z_w\}, \\ A_h^0(x, y, \alpha, w) = \{(u, v, \beta) = (x + w_x, y + w_y, \beta) | w \in Z_w^0\}, \end{cases} \quad (3)$$

где  $w_x = w \cos(\alpha)$  и  $w_y = w \sin(\alpha)$ ;  $(x, y) \in X_h \times Y_h$  – центр апертуры;  $(u, v) \in \bar{X}_h \times \bar{Y}_h$  – точка апертуры;  $w$  – размер апертуры; множества  $Z_w = 1..w$  и  $Z_w^0 = w$ ;  $\alpha$  – угол направления апертуры. Угол, определяющий направление из центра  $(x, y)$  в точку  $(u, v)$  апертуры, находят в виде

$$\beta = \arctg \left( \frac{v - y}{u - x} \right) + \pi n \text{ при } n \in 0..1.$$

Алгоритмы движения по точкам ДИ задает отношение

$$R = \{(x_d, y_d) | d \in 0..7\}, \quad (4)$$

полного строго порядка по направлению движения  $d$  в 8-связанной области, где  $(1, 0)$  есть элемент для  $d = 0$ .

Отношение  $R$  определяет функцию перехода  $T_h^d(x,y)$  в  $h$ -й иерархии в направлении  $d$ . Функция описывает смещение как отдельной точки с координатами  $(x,y) \in X_h \times Y_h$ , так и синхронное смещение множества точек в виде

$$T_h^d(\{(x,y)\}) = \{(a,b) = (x+x_d, y+y_d) \mid (a,b) \in X_h \times Y_h\}. \quad (5)$$

Задача заключается в построении помехоустойчивого метода измерения полей направлений ДИ с опорой на формальный аппарат (1-5).

Измерение полей направлений является частью общей задачи обработки ДИ, представляемой в виде последовательности процедур и функций, сгруппированных поэтапно, очередность выполнения которых определяется при разработке КС и реже – обучении. Это увязывает слои пирамиды  $\mathfrak{P}$  между собой запрограммированной последовательностью отображения данных.

### 3. МЕТОД ПАРАЛЛЕЛЬНЫХ ЦЕПЕЙ

На основе формализма (1-5) рассматривается коррекция изображения, его сглаживание, построение светотеней и измерение полей направлений.

#### 3.1 Коррекция изображения

Обычной процедурой КА является коррекция исходного изображения  $F_0^{(0)} = \{f_0^{(0)}(x,y)\} = \{f(x,y)\} = F$  для обеспечения полного динамического диапазона яркостей:

$$F_0^{(1)} = \{f_0^{(1)}(x,y)\} = \left\{ \frac{(f_0^{(0)}(x,y) - f_{\min})(2^b - 1)}{f_{\max} - f_{\min}} \right\}, \quad (6)$$

где  $f_{\max}$  и  $f_{\min}$  – наибольшее и наименьшее значение яркости изображения;  $b$  – глубина изображения [1]. Расчет  $f_{\min}$  и  $f_{\max}$  можно улучшить, опираясь на гистограмму яркостей или модулей градиента [3]. В последнем случае параметры функции (6) оценивают на этапе обучения программного объекта.

#### 3.2 Сглаживание изображения

Данная процедура выполняется по формуле двумерной дискретной свертки

$$F_0^{(2)} = \{f_0^{(2)}(x,y)\} = \{\mathbf{H} \ast \ast f_0^{(1)}(x,y)\} \quad (7)$$

с ядром вида

$$\mathbf{H} = [h(i,j)] = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

Такая фильтрация применяется для расфокусировки изображения, преимущественно краев линий [1, 3]. Это улучшает результаты КА, особенно для изображений, снятых с ксерокопий и содержащих обедненный состав серого цвета. Утверждать, что сглаживание исходного изображения может увеличить ошибки распознавания изображения при его обработке не следует, так как слои  $F_0^{(0)}$ ,  $F_0^{(1)}$ ,  $F_0^{(2)}$  одновременно хранятся в памяти машины, а функции КА имеют доступ к любому слою пирамиды .

### 3.3 Построение светотеней

Сглаженный слой по формуле (7) служит основой для формирования слоев светотеней  $F_0^{(d+3)} = \{f_0^{(d+3)}(x,y)\}$  по формуле двумерной дискретной свертки

$$F_0^{(d+3)} = \mathbf{H}^d \ast \ast F_0^{(2)}, \quad (8)$$

где  $d \in D = 0..3$  – четыре направления засветки изображения через  $45^\circ$ ;  $\mathbf{H}^d$  – маски Собела [5] в виде

$$\mathbf{H}^0 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad \mathbf{H}^1 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}, \dots, \quad \mathbf{H}^3 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}.$$

Четыре слоя светотеней, вычисленные для ДИ, показаны на рис. 2. Нулевая реакция окрашена серым цветом, положительные значения светлее, а отрицательные – темнее.

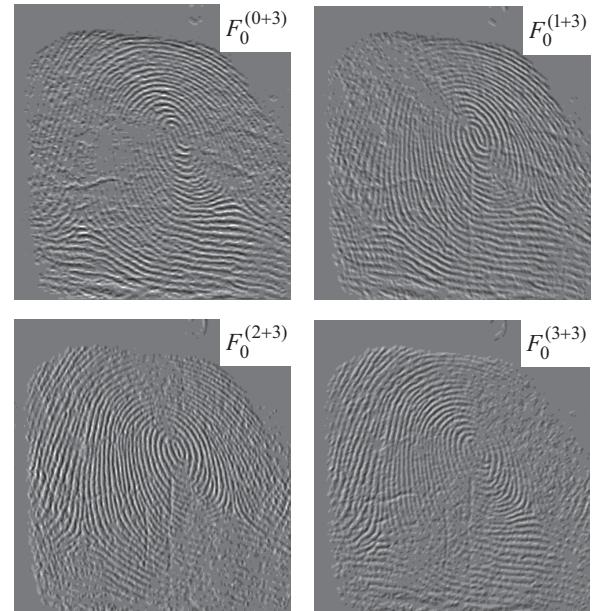


Рис. 2: Четыре слоя светотеней

### 3.4 Измерение полей направлений

Направление линии, по сути, близко к ориентации простой окрестности [5]. Поскольку светотени преимущественно формируются границами линий, применим метод локального адаптивного параллельного сканирования указанных слоев вдоль путей «тени» и «света» в качестве базовой оценки матриц направлений. Выполняемая процедура для направления  $d \in D = 0..3$  реализует отображение

$$\Gamma: \{S_h^{(d+3)}\} \rightarrow \{\{\Delta_h\}, \{\Lambda_h\}\},$$

где  $S_h^{(d+3)} = \{S_h^{(d+3)}(u,v) \mid (u,v) \in \hat{X}_h \times \hat{Y}_h\}$  – множество сегментов с центрами по (2) слоев  $F_0^{(d+3)}$ ;  $\Delta_h = [\delta_h(x,y)]$  – слой  $F_h$  как матрица направлений на сегментах изображения  $S_h$  с углами  $0 \leq \delta_h(x,y) < \pi$ ;  $\Lambda_h = [\lambda_h(x,y)]$  – слой как

матрица достоверностей  $\Lambda_h$  для матрицы направлений  $\Delta_h$ ;  $h$  – номер иерархии, на которой отображаются случайные поля, и  $h \in H = 2..n$ . Измерения выполняются в двух каналах, означенных символом  $k \in \{0,1\}$ : 0 – канал «тени» и 1 – канал «света».

**3.4.1 Первый этап.** Первый этап фиксирует для каждого базового сегмента и заданного направления  $d$  четыре позиции:

$$\begin{cases} p_q^{lk} = (x_q^{lk}, y_q^{lk}) \in P^{lk}, \\ p_q^{rk} = (x_q^{rk}, y_q^{rk}) \in P^{rk}, \end{cases} \quad (9)$$

где  $q$  – длина цепи;  $k \in \{0,1\}$  – метка канала «тени» и «света»;  $l$  и  $r$  – метки левой и правой позиции;  $(x, y)$  – координаты;  $P$  – цепь. Цепи формируются в слоях  $F_0^{(d+3)}$  как выделенные последовательности точек  $\{p_i = (x_i, y_i)\}$  в направлении  $s \in G = \{d+1, d+2, d+3\}$ , которое в среднем перпендикулярно направлению засветки. Суммирование выполняют по модулю 8. Эти отсчеты соответствуют четырем простым цепям на точках изображения, как на вершинах графа. Две простые цепи для «тени» и две простые цепи для «света»  $P^{lk}$  и  $P^{rk}$  в процессе движения развиваются независимо:

$$\begin{cases} P^{lk} = \{p_i^{lk} = (x_i^{lk}, y_i^{lk}) \mid i \in 0..q \wedge p_0^{lk} = p^l\}, \\ P^{rk} = \{p_i^{rk} = (x_i^{rk}, y_i^{rk}) \mid i \in 0..q \wedge p_0^{rk} = p^r\}, \end{cases} \quad (10)$$

где  $k \in \{0,1\}$  – метка канала;  $q$  – длина цепи;  $p^l = (x^l, y^l)$  и  $p^r = (x^r, y^r)$  – две стартовые точки для левых и правых простых цепей. Обычно стартовые точки  $p^l$  и  $p^r$  располагаются в соседних сегментах. Они для сегментов  $S_h^{(d+3)}$  с центрами  $\{(u, v) \in \hat{X}_h \times \hat{Y}_h\}$  по (2) при  $h=2$  определяются методом переноса точки  $(u, v)$  в направлении  $d$  и зеркальном ему  $\bar{d} = d+4$  (суммирование по модулю 8) в апертуре по (3):

$$\begin{cases} \{(x^l, y^l)\} = \{A_0^\circ(u, v, 45d, t) \mid (u, v) \in \hat{X}_h \times \hat{Y}_h\}, \\ \{(x^r, y^r)\} = \{A_0^\circ(u, v, 45\bar{d}, t) \mid (u, v) \in \hat{X}_h \times \hat{Y}_h\}, \end{cases}$$

где  $t$  – расстояние, на которое переносится точка из центра сегмента (в реализации  $t \in 4..7$ ). Точка остается в центре сегмента, если при переносе ее координаты выходят за границы изображения.

Независимое развитие четырех простых цепей и определяется формулой

$$f_0^{(d+3)}(x_i, y_i) = f_0^{(d+3)}(T_0^s(x_{i-1}, y_{i-1})), \quad (11)$$

где  $i > 0$ ;  $d \in D = 0..3$  – направление засветки;  $T_0^s(\cdot)$  – функция перехода по (5);  $s$  – направление для функции перехода как аргумент выражения

$$s = \begin{cases} \arg \max_G (f_0^{(d+3)}(x_i, y_i)) & \text{при } k = 0, \\ \arg \min_G (f_0^{(d+3)}(x_i, y_i)) & \text{при } k = 1. \end{cases}$$

Итак, на каждом шаге развития цепи путь исследуется. Графы функций переходов показаны на рис. 3.

Этап фиксирует в каждом сегменте для текущего направления четыре позиции, точность определения которых существенна. Поэтому величина выбирается оптимальной по критерию: рост приводит к увеличению времени обработки, уменьшение снижает точность определения направлений даже на плавных линиях из-за наличия на ДИ пор, складок, залипаний, рыхлого фона старой бумаги и т.д. Для диагональных направлений выбирают целую часть величины  $q' = q \cdot 0,707$ .

Дополнительная оптимизация выполняется шевелением точек  $p^l$  и  $p^r$  в виде четырех шагов: шаг в направлении  $d$ , два шага в направлении  $\bar{d}$  и еще шаг в направлении  $d$  по (11). При этом точки  $p^l$  и  $p^r$  расщепляются и образуются две, а четыре стартовые точки для развития четырех независимых простых цепей аналогично (10).

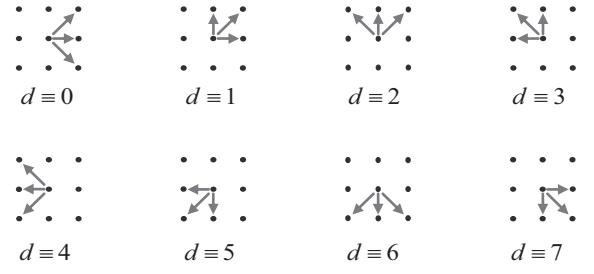


Рис. 3: Графы функций переходов  $T^d$  для  $d \in 0..7$

**3.4.2 Второй этап.** На сегментах  $S_h$  измеряются параметры направлений. Из цепей по (10) зададим два множества  $P^k = \{p^{lk}, p^{rk}\} = \{p_q^{lk}, p_q^{rk}\}$ . Каждое множество содержит две точки для «тени» либо две точки для «света». Расчет сводится к выделению в слоях  $F_0^{(d+3)}$  по (8) отсчетов, отвечающих последовательностям точек

в направлении  $s \in \bar{G} = \{d+5, d+6, d+7\}$ . Множество  $\bar{G}$  по направлениям зеркально множеству  $G$ , а суммирование выполняют по модулю 8. Выбор направлений из  $\bar{G}$  позволяет прослеживать светотени в направлении, обратном предшествующему этапу. Выделенные точки формируют две простые параллельные цепи для «тени» и две простые параллельные цепи для «света», причем параллельные цепи формируются одинаковой функцией перехода на каждом шаге:

$$\begin{cases} \bar{P}^{lk} = \{p_i^{lk} = (x_i^{lk}, y_i^{lk}) \mid i \in q..3q\}, \\ \bar{P}^{rk} = \{p_i^{rk} = (x_i^{rk}, y_i^{rk}) \mid i \in q..3q\}, \end{cases} \quad (12)$$

где  $k \in \{0,1\}$ ;  $3q - q$  – длина цепи;  $l$  и  $r$  – метки левой и правой цепи. Синтез на решетке двух копий простых параллельных цепей определяется условием

$$f_0^{(d+3)}(x_i, y_i) = f_0^{(d+3)}(T_0^s(x_{i-1}, y_{i-1})), \quad (13)$$

где  $T_0^s(\cdot)$  – функция перехода по (5); начала цепей  $(x_q, y_q) \in P^k$ ;  $s$  – направление движения для функции перехода как аргумент выражения

$$s = \arg \psi_G^k(f_0^{(d+3)}(x_i^l, y_i^l), f_0^{(d+3)}(x_i^r, y_i^r))$$

с функцией

$$\psi_G^k(a, b) = \begin{cases} \min_{s \in G} \max(a, b) & \text{при } k = 0, \\ \max_{s \in G} \min(a, b) & \text{при } k = 1 \end{cases}$$

исследования пути на каждом шаге развития цепей.

Для определения направлений «тени» и «света» на сегменте  $S_h^{(d+3)}(u, v)$  при рассматриваемых засветках изображения используются пары точек  $(p_q^{lk}, p_{3q}^{lk})$  или  $(p_q^{rk}, p_{3q}^{rk})$  нулевой иерархии. Связанные с ними направления  $\delta_h^{(dk)}(x, y) \in F_2$  для заданного направления  $d$  определяются по формуле для левой (или правой) цепи:

$$\delta_h^{(dk)}(x, y) = \left( \arctg \left( \frac{y_q^{lk} - y_{3q}^{lk}}{x_q^{lk} - x_{3q}^{lk}} \right) + \pi n \right) \bmod \pi \quad \text{при } n \in 0..1. \quad (14)$$

Действительно, выбор точек для расчета направлений цепей  $\bar{P}^{lk}$  или  $\bar{P}^{rk}$  не имеет значения, так как цепи канала  $k$  параллельны. Показателем достоверности такого решения могут служить величины  $\lambda_h^{(dk)}(x, y) \in F_h$ , собираемые в слоях данных:

$$\lambda_h^{(dk)}(x, y) = \frac{1}{q} \left| \sum_{i=q}^{3q} \psi^k(f_0^{(d+3)}(x_i^{lk}, y_i^{lk}), f_0^{(d+3)}(x_i^{rk}, y_i^{rk})) \right|, \quad (15)$$

где точки  $f_0^{(d+3)}(x_i, y_i)$  выбираются по (13); в  $-$ -м канале функция выбора точек из параллельных цепей находится в виде

$$\psi^k(a, b) = \begin{cases} \max(a, b) & \text{при } k = 0, \\ \min(a, b) & \text{при } k = 1; \end{cases}$$

и  $-$  – метки левой и правой цепей. Достоверность  $\lambda_h$  можно рассматривать как аналог вероятности, отображенной на шкалу  $0 - 2^b$  при глубине  $b$  изображения.

Итак, из двух параллельных цепей выбирается одна, а ее начальная и конечная вершины определяют направление как угол. Достоверность направления оценивается величинами, которые накапливаются в процессе развития цепей. Эти величины выбираются из двух параллельных цепей как значения, наименее отклоняющиеся от нуля. Выбор выполняется на каждом шаге развития цепей и напоминает корреляцию цепей, синтезируемых при движении по критерию (13).

Две независимые цепи по (10), две параллельные цепи (выделены) по (12) и перенос центра сегмента (пунктир) представлены на рис. 4.

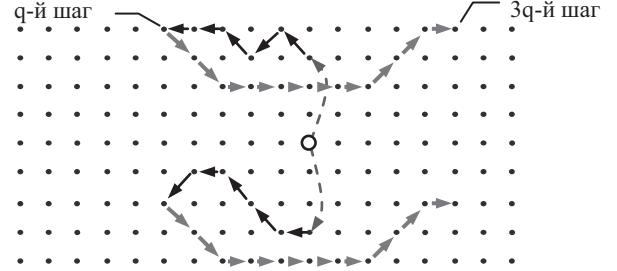


Рис. 4: Граф цепей для  $d = 2$ ,  $G = \{3,4,5\}$  в слое «тени»

Важность процедуры сглаживания изображения (7) очевидна. Она обеспечивает плавность перепадов яркостей точек. Без этого точность измерений ухудшается.

**3.4.3 Третий этап.** На третьем этапе оптимизируются элементы матриц на базовых сегментах  $S_h$  для заданного  $d \in D = 0..3$  повторением этапов 1–2. Заметим, что движение на втором этапе производится в секторе, зеркальном сектору первого этапа. И хотя длина цепи первого этапа вдвое короче длины цепи второго этапа, цепи, казалось бы, должны центрироваться на сегменте. На самом деле это не происходит, цепи отклоняются, измерения в направлении не совпадают с измерениями в противоположном направлении  $\bar{d} = d + 4$ . Но направления  $d$  и  $\bar{d}$  равноправны и можно выбрать для «тени» и «света» наилучшее по достоверности измерение.

Оптимизация сводится к вычислению в слоях  $F_h$  восьми матриц направлений

$$\Delta_h^{(dk)} = [\delta_h^{(dk)}(x, y)] \quad (16)$$

и соответствующих им восьми матриц достоверностей

$$\Lambda_h^{(dk)} = [\lambda_h^{(dk)}(x, y)] \quad (17)$$

при  $k \in \{0,1\}$ . При этом в вершину каждого сегмента  $S_h^{(d+3)}$  слоя  $F_h^{(d+3)}$  записываются собственно направления

$$\delta_h^{(dk)}(x, y) = \delta_h^{(\vartheta(x, y, k)k)}(x, y)$$

и соответствующие им достоверности

$$\lambda_h^{(dk)}(x, y) = \lambda_h^{(\vartheta(x, y, k)k)}(x, y),$$

где направление

$$\vartheta(x, y, k) = \arg \max_{\{d, \bar{d}\}} (\lambda_h^{(dk)}(x, y), \lambda_h^{(\bar{d}k)}(x, y)); \quad (18)$$

$$\vartheta(x, y, k) \in \{d, \bar{d}\}.$$

Направление  $\vartheta(x, y, k)$  максимизирует достоверность. Пусть графы цепей такие, как на рис. 4. Изменим направление на противоположное ему  $\bar{d}$ . Несмотря на совпадение отсчетов

$\{p^l, p^r\}$ , вычисляемых по (10), графы независимых цепей, вероятно, изменятся, вследствие чего и граф параллельных цепей тоже изменится. Тогда матрицы направлений и их достоверностей оптимизируются. Экспериментально обнаружено, что выбор по (18) улучшает результаты КА.

**3.4.4 Четвертый этап.** Это процедура перколяции [2, 5] матриц (16), (17) для направления  $d$  с иерархии измерений  $h = \min(H)$  на более высокие иерархии пирамиды  $\mathfrak{R}$ . Процедура сводится к рекурсивному вычислению по иерархиям в векторных пространствах восьми матриц направлений

$$\Delta_h^{(dk)} = \left[ \delta_h^{(dk)}(x, y) \right] = \left[ \frac{1}{2} \left( \arctg \left( \frac{im_{h-1}^{(dk)}}{re_{h-1}^{(dk)}} \right) + \pi n \right) \right] \quad (19)$$

при  $n \in 0..1$  и восьми матриц достоверностей

$$\Lambda_h^{(dk)} = \left[ \lambda_h^{(dk)}(x, y) \right] = \left[ \kappa \sqrt{(re_{h-1}^{(dk)})^2 + (im_{h-1}^{(dk)})^2} \right], \quad (20)$$

где  $k \in 0..1$  – номер канала; иерархия  $h \in H = 2..n$ ; направление  $d \in D = 0..3$ . Действительная и мнимая части векторов для сегмента  $S_{h-1}(x, y)$  определяются по модели сложения векторов с величинами модуля  $\lambda_{h-1}^{(dk)}(x, y)$  и аргумента  $\delta_{h-1}^{(dk)}(x, y)$  в виде

$$\begin{aligned} re_{h-1}^{(dk)} &= \sum_{(u,v) \in G} \lambda_{h-1}^{(dk)}(u, v) \cos(2\delta_{h-1}^{(dk)}(u, v)), \\ im_{h-1}^{(dk)} &= \sum_{(u,v) \in G} \lambda_{h-1}^{(dk)}(u, v) \sin(2\delta_{h-1}^{(dk)}(u, v)), \end{aligned}$$

где  $G = \bar{X}_{h-1} \times \bar{Y}_{h-1}$  – множество точек сегмента по (1); аргумент  $\delta_{h-1}^{(dk)}(u, v)$  удваивается для расширения полуплоскости  $[0, \pi)$  до плоскости  $[0, 2\pi)$  [5, 7];  $\kappa$  – коэффициент нормировки векторов из  $G$  вида

$$\kappa = \frac{\max_{(u,v) \in G} \lambda_{h-1}^{(dk)}(u, v)}{\sum_{(u,v) \in G} \lambda_{h-1}^{(dk)}(u, v)}.$$

Результат сложения векторов запоминается в вершине сегмента. Данные жестко привязаны к координатам ДИ на разных уровнях пирамиды. Коэффициент нормировки масштабирует достоверность  $\lambda(x, y)$ , которая при  $h > 2$  как мера когерентности направлений достигает максимально возможной величины для идеальной локальной ориентации, а для изотропной структуры она равна нулю [5].

**3.4.5 Пятый этап.** Выполняется отбор направлений на сегментах  $S_h^{(d+3)}$ . Формально алгоритм реализуется последовательным выбором в каналах направлений из  $D = 0..3$  для иерархии  $H = 2..n$  итерационно  $h = \min(H) \Rightarrow H = H \setminus \{h\}$  по формулам

$$\Delta_h = \left[ \delta_h(x, y) \right] = \left[ \delta_h^{(\mathcal{G}(x, y)k)}(x, y) \right], \quad (21)$$

$$\Lambda_h = \left[ \lambda_h(x, y) \right] = \left[ \lambda_h^{(\mathcal{G}(x, y)k)}(x, y) \right], \quad (22)$$

где  $k \in 0..1$  и  $\mathcal{G}(x, y, k) \in D$  – направление-победитель на отдельном сегменте иерархии  $h$  – определяются в виде пары

$$(\mathcal{G}(x, y)k) = \arg \max_{k \in 0..1} \max_{d \in D} \lambda_h^{(dk)}(x, y).$$

Направления покрывают все многообразие ориентаций папиллярных линий  $0 \leq \delta < \pi$  при равнoprавном выборе направлений и  $\bar{d}$  по (18) для простых окрестностей. Матрицы  $\{\Delta_h\}$ ,  $\{\Lambda_h\}$  содержат наиболее правдоподобные направления из каждого канала, но необязательно корректные. Здесь для каждой иерархии  $h \in H$  формируется 18 слоев:  $\{\Delta_h^{(dk)}\}$ ,  $\{\Delta_h\}$ ,  $\{\Lambda_h^{(dk)}\}$ ,  $\{\Lambda_h\}$ , где  $d \in 0..3$ ;

На этом заканчивается измерение полей направлений ДИ.

Матрицы направлений и величин их достоверностей показаны на рис. 5-7.

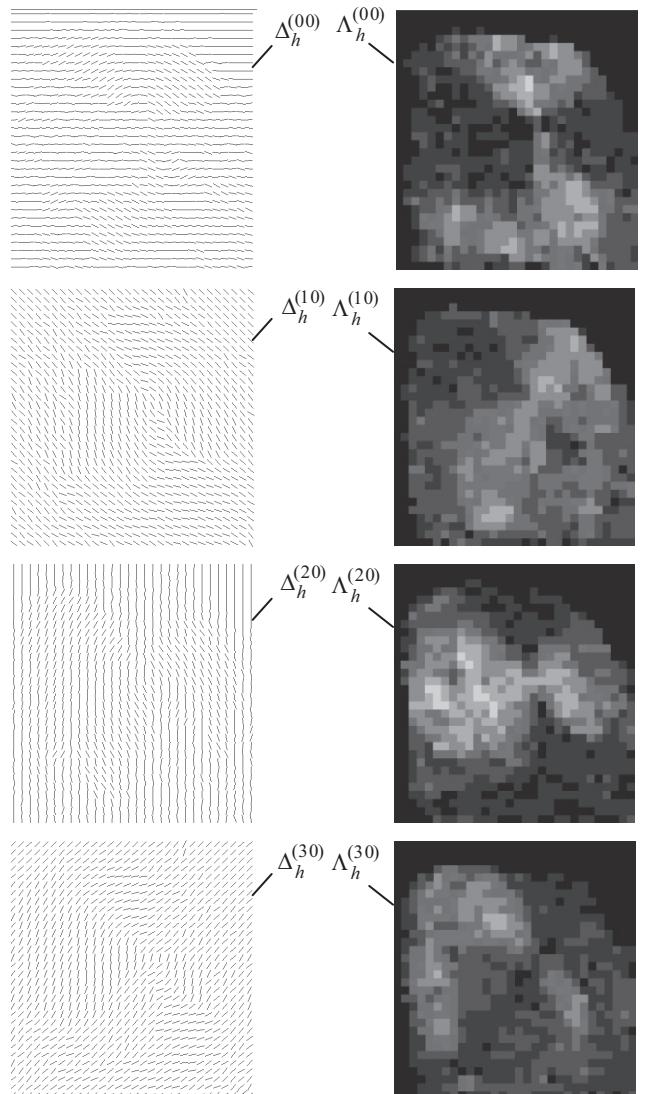


Рис. 5: Направления и достоверности канала «тени»

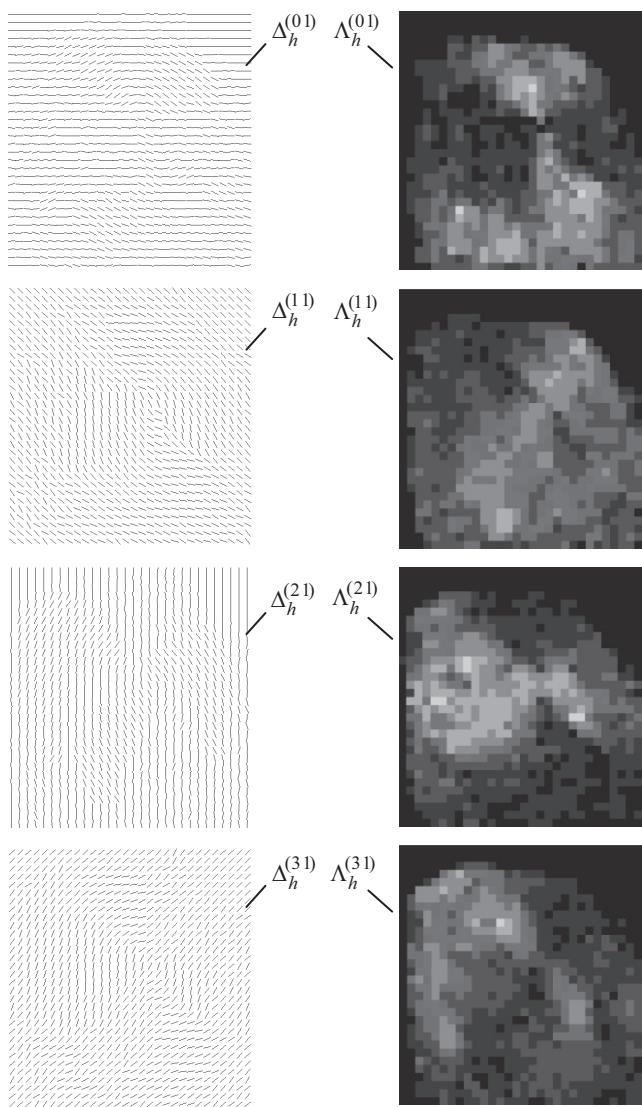


Рис. 6: Направления и достоверности канала «света»

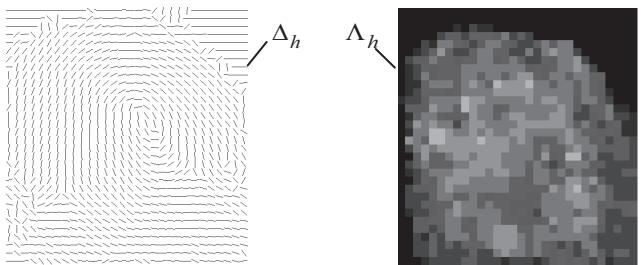


Рис. 7: Отобранные направления и достоверности

На рис 7 можно заметить, что, несмотря на дефекты ДИ, обозначенные на рис. 1, отобранные на основе максимизации достоверности направления (в виде черточек) в этих дефектных областях построены корректно.

Итак, метод измерения полей направлений основан на оценке величин отсчетов при параллельном движении в слоях светотеней, а разделение результатов КА на два канала и

изобилие результатов измерений позволяет улучшить КА изображения и уменьшить ошибки обработки ДИ и, тем самым, уменьшить ошибки идентификации [6].

#### 4. ЗАКЛЮЧЕНИЕ

Предложен метод параллельных цепей для распознавания направлений линий на ДИ. Он базируется на формализме (1)–(5) и выполняется на основе следующих новых решений:

- отслеживание слоев светотеней на основе корреляции точек из параллельных цепей (13), зеркалирования направления движения по (16)–(18) и вероятностной оценки достоверностей направлений по (15);
- многоканальное независимое измерение полей направлений и их достоверностей согласно уравнениям (12);
- переколацию матриц направлений и их достоверностей на более высокие иерархии пирамиды  $\mathcal{K}$  по (19, 20) в рамках пирамидальной модели;
- отбор наиболее достоверных направлений в каналах и синтез матриц более высокого качества отобранных направлений и их достоверностей по (21, 22).

Предложенный метод существенно отличается от метода простых окрестностей [5] как по точности, так и по качеству измерений.

В дальнейшем планируется развитие метода за счет применения трех параллельных цепей и присоединения к ним метода тензорного анализа простых окрестностей.

#### 5. ССЫЛКИ

1. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс; пер. с англ.; под ред. П.А. Чочиа. – М.: Техносфера, 2006. – 1072 с.
2. Новейшие методы обработки изображений / А.А. Потапов, А.А. Пахомов, С.А. Никитин, Ю.В. Гуляев. – М.: Физматлит, 2008. – 496 с.
3. Обработка и анализ изображений в задачах машинного зрения / Ю.В. Визильтер, С.Ю. Желтов, А.В. Бондаренко и др. – М.: Физматкнига, 2010. – 672 с.
4. Справочник по теории вероятностей и математической статистике / В.С. Королюк, Н.И. Портенко, А.В. Скороход, А.Ф. Турбин. – М.: Наука. – Главная редакция физико-математической литературы, 1985. – 640 с.
5. Яне, Б. Цифровая обработка изображений / Б. Яне; пер. с англ. А.М. Измайловой. – М.: Техносфера, 2007. – 584 с.
6. Bolle, R.M. Automatic fingerprint recognition systems / R.M. Bolle, N.K. Ratha. – New York: Springer-Verlag, 2004. – 458 p.
7. Maltoni D. Handbook of fingerprint recognition / D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar. – London: Springer-Verlag, 2009. – 496 p

#### Об авторе

Гудков Владимир Юльевич – доцент Челябинского государственного университета. Его адрес: [Diana@Sonda.ru](mailto:Diana@Sonda.ru).

Область научных интересов:

обработка изображений, распознавание образов, идентификация отпечатков пальцев, комплексы программ.

# Тополого-Геометрическая Модель Изображения Отпечатка Пальца

Гудков Владимир Юльевич, Челябинский государственный университет, Челябинск, Россия  
 diana@sonda.ru

Ушмаев Олег Станиславович, Институт проблем информатики, Москва, Россия  
 oushmaev@ipiran.ru

## Аннотация

В работе предлагается топологическая модель изображения отпечатка пальца, в состав которой входят геометрические и топологические характеристики окрестностей частных признаков. От известных модель отличается тем, что содержит длины отрезков папиллярных линий. Топологическая модель сохраняется в шаблоне изображения. Идентификация изображений выполняется по шаблонам.

**Ключевые слова:** топологические векторы, события, длина связи, идентификация отпечатка пальца.

## 1. ВВЕДЕНИЕ

Обычно идентификация дактилоскопических изображений (ДИ) выполняется на основе шаблонов [2]. Их базис составляет описание частных признаков, которые могут быть детектированы по серому изображению, однако простых методов для этого явно недостаточно [3, 6, 7]. Поэтому при построении шаблона, как правило, опираются на скелет, показанный на рис. 1. При классическом подходе в шаблоне дополнительно сохраняют гребневый счет [6].



Рис. 1: Скелет и частные признаки

В настоящее время известны различные математические модели, используемые для доказательства индивидуальности узора [7]. Каждая из таких моделей нацелена на повышение точности идентификации, однако неизвестно ни одной лучшей, свободной от недостатков модели [6].

Математическая модель должна содержать необходимое и достаточное количество признаков для уверенного распознавания изображений [1, 3]. В дактилоскопии обычно выделяют общие признаки, частные признаки и гребневый счет между ними [6]. Однако гребневый счет, являясь морфологической метрикой, не выдерживает критики. Действительно, в силу предписанного правила измерять гребневый счет вдоль прямой от точки до точки, он не свободен от следующих недостатков [2, 8-10]:

- при мутации окончания в разветвление или разветвления в окончание гребневый счет может изменить значение;
- при измерении гребневого счета вдоль кривых линий его значение неопределенно;
- при смещении частных признаков вдоль линий гребневый счет может изменить значение;
- если прямая, вдоль которой измеряется гребневый счет, проходит вблизи другого частного признака, то значение гребневого счета неоднозначно;
- множество частных признаков, до которых измеряется гребневый счет для данного частного признака, или неопределено или условно ограничено (радиус-вектор).

В условиях помех и операционной неопределенности это приводит к увеличению ошибок идентификации и снижению производительности биометрических систем [5].

Первая попытка преодоления указанных недостатков на основе перехода от морфологических характеристик к топологическим характеристикам ДИ изложена в [8-10]. В данной работе предлагается развитие топологического подхода на основе шаблона изображения в виде отображения

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

где  $F_0^{(m)} = [f_0^{(m)}(x, y)]$  – скелет ДИ;  $L_m$  – список частных признаков;  $L_s$  – список общих признаков;  $L_v$  – список топологических векторов для частных признаков, снабженный длинами отрезков папиллярных линий.

## 2. ШАБЛОН ИЗОБРАЖЕНИЯ

В работе шаблон ДИ по (1) представляется множеством из трех списков данных. Рассмотрим эти списки по порядку.

### 2.1 Список частных признаков

Список окончаний и разветвлений находят в виде

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

где  $M_i$  – частный признак;  $|L_m| = n_1$  – мощность списка;  $i$  – индекс частного признака;  $(x_i, y_i)$ ,  $\alpha_i$ ,  $t_i$ ,  $s_i$ ,  $v_i$ ,  $\theta_i$ ,  $p_i$  и  $h_i$  – координаты, направление, тип, метка сомнительности,

величина и направление кривизны, вероятность и плотность линий в окрестности частного признака. Частные признаки детектируют в информативной области изображения [2], которая на рис. 1 затемнена.

Координаты  $(x_i, y_i)$  частного признака  $M_i$  определяются координатами вершины скелета [6]. Направление  $\alpha_i$  как угол определяется простой цепью вершин скелета для окончания и тремя простыми цепями для разветвления [2]. Тип  $t_i \in \{0,1\}$  устанавливается валентностью вершины скелета как вершины графа [4], где 0 – разветвление и 1 – окончание. Координаты  $(x_i, y_i)$ , направление  $\alpha_i$  и тип  $t_i$  являются основными параметрами  $M_i$ , используемыми для идентификации ДИ [1, 6, 7].

Метка сомнительности  $s_i \in \{0,1\}$  определяется положением частного признака, близким к неинформативным областям (фону), причем 1 означает сомнительность. На границе неинформативных областей частные признаки не детектируют. Величину  $v_i$  и направление  $\theta_i$  кривизны определяют по изменению направлений линий в окрестности  $\varepsilon$  частного признака [6]. Вероятность  $p_i$  рассчитывают как отношение средней величины оценок качества изображения в окрестности  $\varepsilon$  к наилучшей оценке качества в информативной области ДИ [2]. Плотность линий  $h_i$  рассчитывают как среднее количество линий, умещающихся в окрестности  $\varepsilon$  на прямой, проведенной перпендикулярно линиям [2]. Алгоритм обработки ДИ определяет размер окрестности  $\varepsilon$ .

## 2.2 Список общих признаков

Список петель, дельт и завитков  $L_s$  находят в виде

$$L_s = \{S_i = \{(x_i, y_i), \alpha_i, t_i, s_i, p_i, h_i\} | i \in 1..n_2\}, \quad (3)$$

где  $S_i$  – общий признак;  $|L_s| = n_2$  – мощность списка;  $i$  – индекс общего признака;  $x_i, y_i, \alpha_i, t_i, s_i, p_i$  и  $h_i$  – координаты, направление, тип, метка сомнительности, вероятность и плотность линий в окрестности общего признака. Величина и направление кривизны для общего признака теряют смысл и не используются. В целом список подобен списку  $L_m$  по (2).

Координаты  $(x_i, y_i)$  общего признака определяются координатами центра фигур (завиток, петля и дельта), вписанных в соответствующую область изображения. Направление  $\alpha_i \in S_i$  определяется прослеживанием хода линий, а тип  $t_i \in \{s_w, s_l, s_d\}$  определяется по характеру изгиба линий, где  $s_w$  – завиток,  $s_l$  – петля,  $s_d$  – дельта [2]. Координаты  $(x_i, y_i)$ , направление  $\alpha_i$  и тип  $t_i$  являются базовыми параметрами  $S_i$ , используемыми при идентификации ДИ по общим признакам и типу узора [6]. Метка сомнительности  $s_i$  определяется как вероятность  $p_i$ , меньшая заданной величины, причем 1 означает сомнительность. Вероятность  $p_i$  и плотность линий  $h_i$  рассчитываются аналогично (2), но окрестность выбирается значительно большего размера.

## 2.3 Список топологических векторов

Этот список  $L_v$  вычисляют на основе списка частных признаков  $M_i$ , слоя скелета  $F_0^{(m)}$  и других матриц, элементы которых отображают локальные свойства ДИ.

Для каждого частного признака  $M_i$  синтезируют один топологический вектор  $V_i$ . Множество всех топологических векторов записывают в виде

$$L_v = \{V_i = \{e_j, l_j, n_j\} | i \in 1..n_1, j \in 1..m_i\}, \quad (4)$$

где  $V_i$  – топологический вектор для  $M_i$ ;  $|L_v| = |L_m| = n_1$  – мощность списка;  $i$  – индекс топологического вектора, совпадающий с индексом частного признака;  $j$  – номер связи в топологическом векторе;  $e_j$  – событие, а  $l_j$  – длина связи, сформированная частным признаком с номером  $n_j$ ;  $m_i$  – количество связей, зависящее от типа частного признака.

Опишем процедуру построения топологических векторов. В области ясного ДИ выделяют линии и формируют стилизованное изображение в виде скелета. По скелету детектируют два типа частных признаков: окончания и разветвления [6]. На рис. 1 показаны направления частных признаков. Направление как угол указывает в область увеличения числа линий. Оно параллельно касательной к папиллярной линии в малой окрестности частного признака.

Каждый частный признак нумеруют и описывают координатами, направлением, типом, меткой сомнительности, величиной и направлением кривизны, вероятностью и плотностью линий в окрестности частного признака (2). Затем от каждого частного признака фиксируют две проекции: вправо и влево перпендикулярно направлению частного признака на смежные скелетные линии. На рис. 2 проекции показаны пунктиром, а две соответствующие вершины скелета на линиях 1 и 2 закрашиваются «цветом проекции».

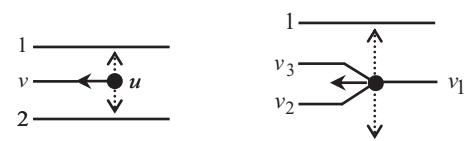


Рис. 2: Проекции от окончания и разветвления

Выберем один частный признак  $M_i$  и проведем через координаты  $(x_i, y_i)$  вправо и влево сечение на глубину нескольких линий  $m$  перпендикулярно касательным к пересекаемым линиям и пронумеруем по спирали, разворачивающейся по часовой стрелке, рассеченные линии, которые назовем связями. Сечение проходит, отслеживая направление кривизны линий [2]. Глубина сечения обычно варьируется от одной до восьми линий вправо и так же влево. Одна линия в сечении образует две связи. Количество связей в топологическом векторе рассчитывается по формуле

$$m_i = \begin{cases} 4m+1, & \text{если } t_i \in \{1\} - \text{окончание} \\ 4m+3, & \text{если } t_i \in \{0\} - \text{разветвление.} \end{cases} \quad (5)$$

Топологический вектор определяют по сечению методом слежения за ходом каждой связи от сечения до встречи с другим частным признаком, расположенным на связи, или с проекцией от него на связь. При этом на связях детектируются следующие возможные события, показанные на рис. 3 и представленные в двоичном коде:

- 0000** – на связи нет ни частного признака, ни проекции от него (линия обрывается на краю отпечатка);
- 1101** – на связи проекция от окончания, расположенного справа по ходу связи, направление окончания ориентировано навстречу ходу связи;
- 1001** – на связи проекция от окончания, расположенного справа по ходу связи, направление окончания ориентировано по ходу связи;
- 1110** – на связи проекция от окончания, расположенного слева по ходу связи, направление окончания ориентировано навстречу ходу связи;
- 1010** – на связи проекция от окончания, расположенного слева по ходу связи, направление окончания ориентировано по ходу связи;
- 0101** – на связи проекция от разветвления, расположенного справа по ходу связи, направление разветвления ориентировано навстречу ходу связи;
- 0001** – на связи проекция от разветвления, расположенного справа по ходу связи, направление разветвления ориентировано по ходу связи;
- 0110** – на связи проекция от разветвления, расположенного слева по ходу связи, направление разветвления ориентировано навстречу ходу связи;
- 0110** – на связи проекция от разветвления, расположенного слева по ходу связи, направление разветвления ориентировано по ходу связи;
- 0011** – разветвление на связи, направление которого ориентировано по ходу связи;
- 0111** – разветвление на связи, образованной линией, касательная к которой образует минимальный угол при повороте направления разветвления против часовой стрелки;
- 1011** – разветвление на связи, образованной линией, касательная к которой образует минимальный угол при повороте направления разветвления по часовой стрелке;
- 1111** – окончание на связи, направление которого ориентировано навстречу ходу связи;
- 1100** – связь по линии замыкается, какой-либо частный признак или проекция от него отсутствует.

С событием как двоичным числом, детектированным на связи, ассоциируют номер частного признака, инициирующего это событие. Событие привязано к номеру связи. Для событий 0000 и 1100 номера частных признаков отсутствуют. Нумерованный набор связей с заданными событиями и номерами частных признаков есть *базовый топологический вектор* (экономный). Событие и номер частного признака образуют упорядоченную пару. Событие может быть дополнено длиной связи от сечения до позиции, в которой детектируется это событие. Так формируется *расширенный топологический вектор*. Событие, номер частного признака и длина связи образуют упорядоченную тройку. Для событий 0000 и 1100 длины связей описывают информативные области, свободные от частных признаков. При этом длины связей, обрывающихся на краю отпечатка, устойчивы в том смысле, что не укорачиваются при полной прокатке пальца.

Местоположение бита в двоичном числе определяет тип частного признака, его направление по отношению к направлению хода связи, его местоположение по отношению к связи и др. (см. рис. 3). События позволяют оперативно сопоставить (простыми компьютерными операциями) важную детальную информацию, необходимую для скоростной и надежной идентификации узора.

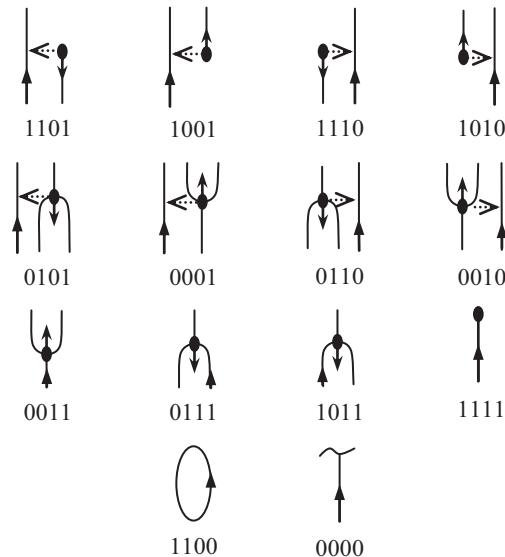


Рис. 3: События, детектируемые на связи

Сечение разрезает линии на связи, пронумерованные по спирали, разворачивающейся по часовой стрелке. На рис. 4 и рис. 5 сечения показаны пунктиром. Длины связей, соответствующие сечениям, представлены в табл. 1 и табл. 2 символами.

При *мутациях* частного признака окончание замыкается в разветвление, а разветвление разрывается в окончание. Вследствие *мутаций* связи перенумеровываются, но их содержание сохраняется. Так окончание на рис. 4 после мутации в разветвление приобретает топологическую структуру, показанную на рис. 5. Несложно синтезировать закон перенумерации связей для соответствующей мутации частного признака. Так в табл. 1, 2 видно, что третья связь для окончания перенумерована в нулевую связь для разветвления без изменения содержания, нулевая – во вторую и т.д.

Синтез топологического вектора выполняют для каждого частного признака, а последовательность действий – в информативной области, где прослеживается ход связи. Поэтому необходима сегментация узора на информативные и неинформативные области.

На этом синтез списка топологических векторов завершает. Базовый топологический вектор инвариантен к повороту и переносу ДИ, масштабному преобразованию ДИ, линейным и нелинейным деформациям ДИ [2, 10]. Расширенный топологический вектор отражает морфологические свойства узора и теряет «топологическую чистоту». Если не учитывать несовершенство методов распознавания, то изменить содержание базового топологического вектора могут только дефекты изображения. Так как не известны методы, идеально обрабатывающие произвольные искажения узора [6], то при прочих равных условиях такой топологический вектор оптимален.

**Таблица 1.** Топологический вектор для окончания

Номер связи	Событие	Индекс	Длина связи
0	1110	22	$l_0$
1	0001	21	$l_1$
2	1110	23	$l_2$
3	1001	24	$l_3$
4	1111	22	$l_4$
5	0011	21	$l_5$
6	1111	23	$l_6$
7	1010	24	$l_7$
8	0010	25	$l_8$
9	0010	21	$l_9$
10	1010	20	$l_{10}$
11	1010	26	$l_{11}$
12	0011	25	$l_{12}$
13	0000	—	—
14	1001	20	$l_{14}$
15	1111	27	$l_{15}$
16	0001	25	$l_{16}$

**Таблица 2.** Топологический вектор для разветвления

Номер связи	Событие	Индекс	Длина связи
0	1001	24	$l_0$
1	1111	22	$l_1$
2	1110	22	$l_2$
3	0001	21	$l_3$
4	1110	23	$l_4$
5	1010	24	$l_5$
6	0010	25	$l_6$
7	0011	21	$l_7$
8	1111	23	$l_8$
9	1010	26	$l_9$
10	0011	25	$l_{10}$
11	0010	21	$l_{11}$
12	1010	20	$l_{12}$
13	1111	27	$l_{13}$
14	0001	25	$l_{14}$
15	0000	—	—
16	1001	20	$l_{16}$
17	1100	—	—
18	1100	—	—

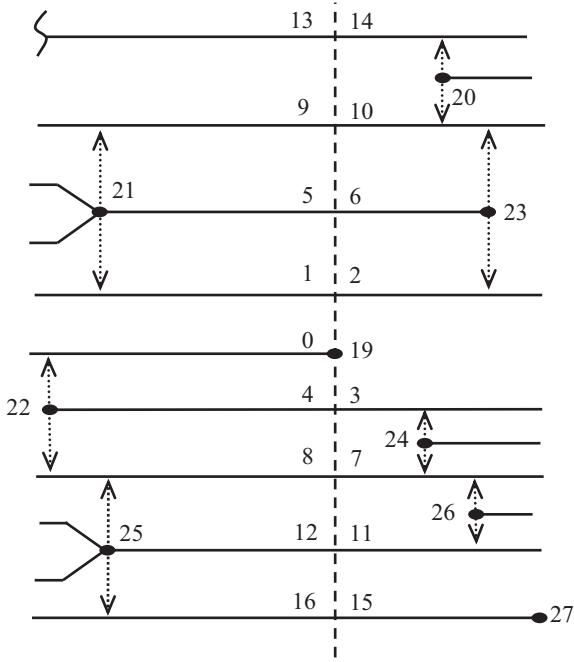


Рис. 4: Сечение для окончания

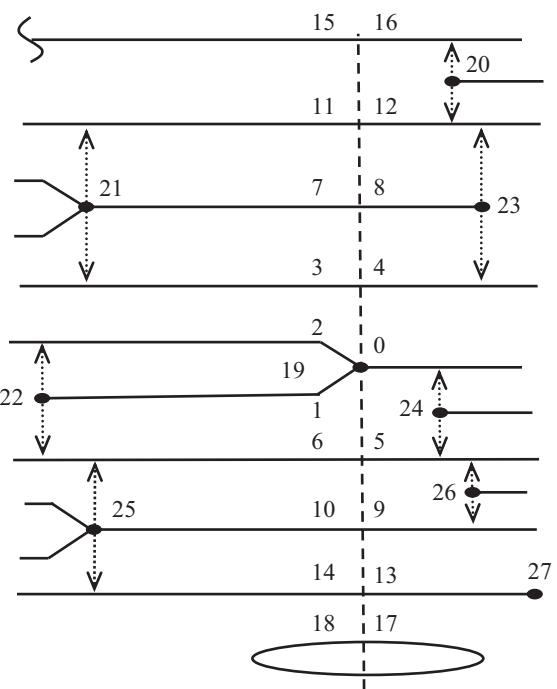


Рис. 5: Сечение для разветвления

На рис. 4 в сечении для окончания 19 пронумерованы связи 0–16, а соответствующий топологический вектор представлен в табл. 1. Нумерация связей начинается с линии, на которой лежит окончание (связь №0 на рис. 4). При глубине сечения  $m=4$  для окончания формируется семнадцать связей  $m_t=17$  по (5).

На рис. 5 в сечении для разветвления 19 пронумерованы связи 0–18, а соответствующий топологический вектор представлен в табл. 2. Нумерация связей начинается с линии, которая расщепляется на две другие линии (связь №0 на рис. 5). При глубине сечения  $m=4$  для разветвления формируется девятнадцать связей  $m_t=19$  согласно (5).

Предлагаемый базовый топологический вектор развивает работу американского изобретателя Spartrow [8-10], первым наметившего указанный подход. Простое сопоставление публикаций Spartrow с предлагаемой моделью выявляет следующие преимущества топологических векторов по (4):

- спиралевидный способ нумерации позволяет сохранить монотонность роста номеров связей при произвольном изменении глубины сечения  $m$ ;
- внутренние связи с номером  $j \in 0..4$  участвуют в формировании топологического вектора, что позволяет синтезировать однозначные правила перенумерации связей при различных мутациях частных признаков;
- сечение отслеживает направления кривизны узора, что повышает устойчивость топологического вектора при нелинейных деформациях узора;
- события представлены наиболее полно за счет использования проекций;
- полнота событий позволяет однозначно определить состав множества частных признаков, отраженных в топологическом векторе и определяемых структурой изображения, а не условными геометрическими характеристиками;
- числовая характеристика события не зависит от деформаций изображения, так как сечение отслеживает направление кривизны линий;
- длины связей дополняют топологические векторы новыми данными, которые невозможно рассчитать на основе геометрических характеристик частных признаков, так как учитывают искривленность линий.

Топологический вектор несет информацию не только о величине гребневого счета в виде номера связи, но и новые данные, недоступные при использовании гребневого счета. Так из гребневого счета в общем случае невозможно определить, навстречу или по ходу связи ориентирован частный признак, а также справа или слева от связи он находится. Действительно, при анализе частных признаков, расположенных по разные стороны петли или завитка (линии закручиваются), геометрические характеристики не позволяют однозначно рассчитать событие, получаемое при прослеживании связи. Гребневый счет здесь не работает. Новые данные в виде событий обеспечивают более полную информацию о фрагменте узора, чем гребневый счет: события, дополненные частными признаками, позволяют реконструировать скелет ДИ в виде ломаных линий [10]. Номер одного частного признака присутствует в различных топологических векторах. Образуется сильно связанный граф, по которому можно переходить от одного топологического вектора к другому. Это обеспечивает целостность и устойчивость описания узора. Количество связей  $m$ , по (5) играет важную роль. Их количество влияет на вероятность того, что граф с вершинами  $V_i$  связанный.

Топологические векторы позволяют описать структуру изображения на различном уровне. По сути, они формируют банк данных детального описания папиллярных узоров. Действительно, крючок образуется разветвлением, для которого событие 1111 находится на первой или второй связи, а длина связи мала. Фрагмент линии образуется двумя окончаниями, замыкающимися друг на друга нулевыми связями с малой длиной. Глазок образуется двумя разветвлениями, первая и вторая связи которых замыкаются на эти разветвления с событиями 0111 и 1011 и имеют малую длину. Мостик образуется двумя разветвлениями, для

которых первая или вторая связь является общей, а длина связи мала. Эти часто встречающиеся фрагменты узора представлены на рис. 6. Под малой длиной связи понимается величина, соизмеримая с локально усредненным периодом линий.

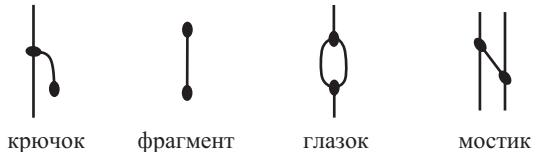


Рис. 6: Детальное описание фрагментов узора

Очевидно, что банк данных детального описания папиллярных узоров улучшает информативность признаков и точность идентификации [6, 7]. Но это не единственное преимущество модели. Расположенные в определенных областях ДИ, эти информативные признаки могут порождать сигнатуры, позволяющие ускорить идентификацию ДИ в десятки раз.

### 3. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

По результатам исследований в компании Sonda Technologies разработано программное обеспечение для распознавания изображений отпечатков пальцев и их следов, содержащее функцию обработки изображения и функцию идентификации изображений по шаблонам. Функция обработки генерирует шаблоны, снабженные топологическими векторами для частных признаков, а функция идентификации вычисляет степень подобия шаблонов. Оценка качества работы этих функций без вычисления длин связей выполнена на teste FVC 2004 [11], на котором 43 разработчика программ представили 67 алгоритмов распознавания ДИ. Результаты оценок в виде DET-кривых показаны на рис. 7.

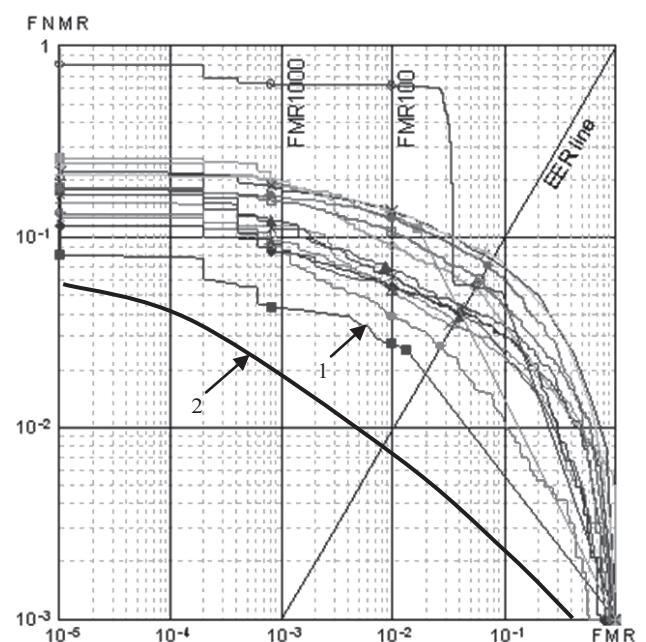


Рис. 7: Результаты экспериментов

На рис. 7 кривая, отмеченная символом 1, рассчитана на основе топологических векторов, не содержащих длины связей. Она построена на тесте FVC 2004 по базе изображений отпечатков пальцев DB1, снятых с оптического сенсора V300 фирмы CrossMatch. Видно, что алгоритм, использующий топологические векторы, допускает минимальные ошибки распознавания ДИ даже без учета длин связей. Кривая, отмеченная символом 2, получена на той же базе изображений DB1, но с учетом длин связей в составе топологических векторов. Эта кривая получена в лабораторных условиях. Величины ошибок идентификации существенно уменьшились.

#### 4. ЗАКЛЮЧЕНИЕ

Предложена топологическая модель, для построения которой требуется выполнение сравнительно непростых методов обработки ДИ [1-3]. Эта модель сохраняется в шаблоне изображения. Граф каждой модели сильно связан и обладает высокой степенью избыточности, а ребра графа помечены длиной связи.

В тополого-геометрической модели используются новые способы представления шаблона ДИ. Объемом шаблона ДИ можно управлять выбором базового или расширенного топологического вектора. Для некоторых режимов идентификации достаточно одного списка частных признаков.

Экспериментально доказано, что идентификация ДИ на основе шаблонов, в состав которых входят тополого-геометрические векторы с длинами связей, может быть выполнена с повышенной точностью и надежностью. Преимущества новой модели по сравнению с ранее использованной показаны на примере теста FVC 2004.

Дальнейшее развитие работ планируется в направлении тополого-геометрического подхода для описания отрезков линий, а не точек.

#### 5. ССЫЛКИ

1. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс; пер. с англ.; под ред. П.А. Чочиа. – М.: Техносфера, 2006. – 1072 с.
2. Гудков, В.Ю. Топологические модели и обработка изображений отпечатков пальцев: монография / В.Ю. Гудков. – Saarbrücken: LAP, 2011. – 304 с. – ISBN: 978-3-8465-4568-3 (Germany).
3. Ковалевский, В.А. Методы оптимальных решений в распознавании изображений / В.А. Ковалевский. – М.: Наука, 1976. – 328 с.
4. Новиков, Ф.А. Дискретная математика для программистов: учебник / Ф.А. Новиков. – СПб.: Питер, 2001. – 304 с.
5. Ушмаев, О.С. Адаптация биометрической системы кискажающим факторам на примере дактилоскопической идентификации / О.С. Ушмаев // Информатика и ее применение. – 2009. – Т. 3, Вып. 2. – С. 25–33.
6. Handbook of fingerprint recognition / D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar. – London: Springer-Verlag, 2009. – 496 p.
7. Komarinski P. Automated fingerprint identification systems / P. Komarinski. – New York: Academic Press, 2005. – 295 p.

8. Pat. 4817183 USA, Int. Cl. G06K 9/00. Fingerprint recognition and retrieval system / M.K. Sparrow (Winchester). – Field: Apr. 1, 1987; Date of patent: Mar. 28, 1989; U.S.Cl. 382/4. – 50 p.
9. Pat. 5631971 USA, Int. Cl. G06K 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.
10. 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.
11. <http://bias.csr.unibo.it/fvc2004/results.asp>.

#### Об авторах

Гудков Владимир Юльевич – кандидат технических наук, доцент Челябинского государственного университета. Его адрес: [diana@sonda.ru](mailto:diana@sonda.ru).

Область научных интересов: обработка изображений, распознавание образов, дактилоскопические технологии, комплексы программ.

Ушмаев Олег Станиславович – доктор технических наук, доцент Института проблем информатики РАН. Его адрес: [oushmaev@ipiran.ru](mailto:oushmaev@ipiran.ru).

Область научных интересов: мультибиометрические системы идентификации, обработка изображений, криптография, комплексы программ.

# Получение устойчивого криптографического ключа из биометрической характеристики изображения отпечатков пальцев

Ушмаев О., С., Кузнецов В. В.

Институт проблем информатики РАН,  
oushmaev@ipiran.ru, ikuvlad@gmail.com

## Аннотация

В статье рассматривается метод извлечения устойчивой двоичной строки из изображения отпечатка пальца. Особые точки отпечатка – минюции – и их связи друг с другом формируют топологическое описание шаблона отпечатка пальца, которое ставит в соответствие каждой минюции 88-битный бинарный вектор. Топологическое описание само по себе является графом, и для введения на нем порядка разработаны три метода: с введением внутреннего порядка на множестве вершин, с публикацией контрольных точек, а также метод, объединяющий оба предыдущих. Предложенные методы позволяют строить вектор длиной 600-1728 битов, содержащий около 25% ошибочных позиций. Для исправления ошибок используются коды Адамара, а также каскадное кодирование, включающее БЧХ-коды и репликацию. Таким образом с вероятностью около 90% находится исходный ключ с энтропией в 20-36 битов.

**Ключевые слова:** биометрическая идентификация; отпечатки пальцев; защищенная идентификация.

## 1. ВВЕДЕНИЕ

Развитие телекоммуникаций последних лет привело к проникновению интернет-технологий в повседневную жизнь. Вместе с этим возникла потребность в удаленном доступе к услугам, которые одновременно предполагают и надежность аутентификации, и анонимность доступа пользователя. Для надежности подтверждения личности все чаще прибегают непосредственно к использованию биометрии, однако ей присущи свои проблемы. В сложившихся на практике системах биометрической идентификации биометрический шаблон полностью раскрывает личность человека, что сильно затрудняет применение биометрической идентификации в масштабных информационных системах. Также можно отметить сложность интеграции в традиционную инфраструктуру шифрования/ЭЦП и невозможность перевыдачи фальсифицированной ключевой информации. Решить эти проблемы позволяет генерация отзываемого криптографического (или идентификационного) ключа, которая использует биометрию пользователя, но не включает ее.

## 2. ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ

Основная проблема всех биометрических признаков состоит в их относительной нестабильности, т.е. в двух последовательных предъявлениях их образы будут побитно неустойчивы. Как следствие, невозможно применение шифрования и обычных хеш-функций с последующей проверкой результатов разных предъявлений на равенство. Для решения этой проблемы в мировой практике сложились два подхода: «нечеткое хранилище» («fuzzy

vault») и «нечеткий экстрактор» («fuzzy extractor») [2, 5]. Объединяет их использование инфраструктуры открытого хелпера, аналогичной той, что используется при шифровании с открытым ключом. На основе пары «хелпер – предъявленный идентификатор» при аутентификации происходит построение криптографического ключа. Различие между нечетким хранилищем и нечетким экстрактором состоит в том, что в первом случае открытый хелпер представляет собой множество произвольного вида, в то время как во втором – конечномерный вектор. Защита от разглашения биометрического идентификатора состоит в том, что ни открытый хелпер, ни криптографический ключ не позволяют вычислить пользовательскую биометрию. Как следствие, невозможно установить личность путем анализа других биометрических систем, но при этом возможно подтвердить, что данные, предъявленные при регистрации и аутентификации, принадлежат одному человеку с достаточной степенью достоверности. При этом оба подхода позволяют перевыдать ключевые данные в новой сессии регистрации, благодаря чему по удобству использования биометрия становится ближе к паролю и другим перевыдаваемым идентификаторам.

Большинство предыдущих исследований, посвященных совмещению криптографии с биометрией отпечатков пальцев, использовали подход нечеткого хранилища [4, 6, 7], а единственный описанный в литературе нечеткий экстрактор имеет высокую чувствительность к шуму [1]. Такой дисбаланс в количестве публикаций связан традиционными методами построения и сравнения шаблонов отпечатков. Стоит отметить, что проблемы совмещения биометрии и криптографии не популярны среди отечественных исследований, при анализе литературы удалось найти только две работы [3, 9].

## 3. НЕЧЕТКИЙ ЭКСТРАКТОР ДЛЯ ОТПЕЧАТКА ПАЛЬЦА

В данной работе рассматривается нечеткий экстрактор для отпечатков пальцев, т.е. алгоритм преобразования изображения отпечатка в побитно устойчивый вектор. Структура экстрактора представляет собой объединение двух алгоритмов: построения зашумленного биометрического вектора и коррекции ошибок. Для построения зашумленного биометрического вектора из изображения отпечатка пальца на сегодняшний день не существует общепринятого подхода, что связано с особенностями шаблона отпечатка пальца. Традиционно шаблон отпечатков представляется контрольными точками (разветвлениями и окончаниями папиллярных линий) и их локальными атрибутами. Глобально эту структуру невозможно описать конечномерным вектором, однако локальные особенности контрольной точки допускают подобное описание с использованием топологии, описанной в совместной статье О. С. Ушмаева и Ю. В. Гудкова [3]. Топология описывает отпечаток как граф,

в котором контрольные точки связаны ближайшими к ним папиллярными линиями. Это позволяет дополнить локальные описания контрольных точек вектором (называемым топологическим), описывающим связь с другими точками. На основе этих описаний реализованы два метода получения устойчивого криптографического ключа: с введением канонической нумерации особых точек через публикацию части координат (метод с введением внешнего порядка) и метод с введением внутреннего порядка в пространстве топологических векторов.

### 3.1 Метод с введением канонической нумерации

Метод с введением канонической нумерации предусматривает публикацию координат нескольких контрольных точек. Их координаты используются в качестве точки отсчета для введения на отпечатке системы координат, благодаря чему появляется возможность занумеровать папиллярные линии и ключевые точки каноническим образом. Каноническая нумерация позволяет сопоставлять особые точки в разных предъявлениях и из их топологических векторов формировать конечномерный вектор (называемый биометрическим) относительно большой размерности – вплоть до 704 битов. Биометрические вектора не являются побитно устойчивыми: в среднем, ошибка наблюдается в 22% битов. Открытый хелпер включает координаты контрольных точек для позиционирования (от пяти до восьми) и результат логического сложения биометрического вектора с криптографическим ключом. Во время аутентификации по точкам из открытого хелпера вводится нумерация папиллярных линий на предъявленном отпечатке, строится биометрический вектор, который складывается с вектором-суммой из хелпера, и результат подвергается этапу коррекции ошибок, что делает возможным построение непосредственно криптографического ключа.

### 3.2 Исправление ошибок

Исправление ошибок реализовано с использованием каскадного кодирования, включающего кодирование с последовательным применением корректирующих кодов Боуза-Чоудхури-Хокингхема и репликации. Такое кодирование обусловлено структурой ошибок: БЧХ корректируют шумовые ошибки, в то время как репликация позволяет справляться с блочными (связанными с «выпадением» минюции).

### 3.3 Метод с введением внутреннего порядка

Метод, основанный на введении внутреннего порядка, состоит в разбиении пространства топологических векторов на кластеры. Каждый бит биометрического вектора сопоставляется с кластером. В качестве исходных данных выступает множество топологических векторов обрабатываемого отпечатка. Для каждого вектора находятся  $k$  ближайших в метрике Хемминга кластеров. Биты биометрического вектора, сопоставленные с этими кластерами, полагаются единичными. В качестве центров кластеров используются вектора-столбцы матрицы Адамара, что делает распределение векторов по кластерам более устойчивым к помехам. Биометрический вектор содержит в среднем 35% ошибочных битов, но при этом он имеет большую длину (до 1024 битов) и значительно устойчивее к блочным ошибкам. Этапы регистрации, аутентификации и коррекции ошибок, в целом, аналогичны предыдущему методу.

Для ключа длиной 30 битов первый метод позволяет достичь уровня ошибок  $FRR < 19\%$  при  $FAR < 0.01\%$ . Второй метод в тех же условиях дает  $FRR = 37\%$  при  $FAR = 3\%$ . К тому же регулировка осуществляется подбором параметров этапа коррекции ошибок, что далеко не всегда оптимально. С другой стороны, при относительно большой энтропии метод с введением внешнего порядка имеет окончательно  $FAR$ , но высокий уровень  $FRR$ , и имеет смысл снизить последний ценой некоторого увеличения уровня ложного допуска. Для решения этой задачи были проанализированы биометрические вектора, на которых метод с введением внешнего порядка дает ошибку ложного недопуска, и было установлено, что большая часть из них связана с блочными ошибками. Действительно, даже если биометрический вектор строится из восьми минюций, из которых выпадают две, то это ограничивает снизу число ошибочных битов 25%, что находится на границе возможностей БЧХ (при условии полного отсутствия других шумовых ошибок). Основная идея борьбы с блочными ошибками состоит в совмещении биометрических векторов, полученных методами с введением внутреннего порядка (устойчивых к блочным ошибкам) и внешнего порядка. Поскольку в работе [8] была продемонстрирована связь эффективности коррекции ошибок с методами группирования битов с разной вероятностью единичной ошибки, в рамках построения итогового метода были исследованы различные алгоритмы объединения векторов. Первый алгоритм заключался в простом конкатенировании векторов, второй – в последовательном чередовании битов из разных векторов (например, при объединении векторов равной длины биты первого биометрического вектора займут нечетные позиции, а второго – четные) и полностью случайный порядок следования битов. Была показана корреляция между эффективностью коррекции ошибок в рамках выбранного алгоритма кодирования. Результаты для совмещения векторов равной длины приведены в таблицах 1-5. Таким образом, для ключа с энтропией в 30 битов была показана возможность более чем двукратного уменьшения ошибки ложного недопуска за счет увеличения  $FAR$  на 0,8%.

## 4. ЗАКЛЮЧЕНИЕ

Рассмотренный метод получения устойчивого криптографического ключа в полной мере отвечает предъявляемым требованиям в области анонимности аутентификации и использует существующую инфраструктуру шифрования с открытым ключом. Стоит отметить, что на непосредственно используемый криптографический ключ не накладывается никаких ограничений, и он может быть отозван и перевыдан в случае фальсификации. Это обеспечивает преимущество перед существующими системами шифрования и в перспективе дает возможность электронного доступа к услугам, требующим анонимного доступа, таким как процедура тайного голосования. Показан способ построения более удобного варианта нечеткого экстрактора, однако не решены проблемы потенциальной утечки информации через открытый хелпер.

Энтропия ключа, битов:

	<b>26</b>	<b>30</b>	<b>36</b>
FAR, %	0	0	0
FRR, %	19,3	19	19,8

Таблица 2: Результаты для метода с введением внутреннего порядка

FAR, %	11,2	3	1,4
FRR, %	30	37	46,8

Таблица 3: Результаты для объединения конкатенацией

FAR, %	5,6	2	0,8
FRR, %	6,8	9,9	13,6

Таблица 4: Результаты для объединения чередованием

FAR, %	4	0,8	0,3
FRR, %	7,8	8,5	10,6

Таблица 5: Результаты для случайного объединения

FAR, %	3,8	1,1	0,2
FRR, %	5,1	9,5	13,3

Таблица 1: Результаты для метода с введением внешнего порядка

## 5. ССЫЛКИ

- [1] Arakala A., Jeffers J., Horadam K.J. Fuzzy Extractors for Minutiae-Based Fingerprint Authentication // Advances in Biometrics, Springer, pp. 760-769, 2007.
- [2] Dodis Y., Ostrovsky R., Reyzin L., Smith A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data // SIAM Journal on Computing, vol. 38 (1), pp. 97-139, 2008.
- [3] Gudkov V.Yu., Ushmaev O.S. A Topological Approach to User-Dependent Key Extraction from Fingerprint // Proc. ICPR2010, p.1281-1284.
- [4] Jo J.G., Seo J.W., Lee H.W. Biometric Digital Signature Key Generation and Cryptography Communication Based on Fingerprint. // FAW 2007, LNCS 4613, Springer, pp. 38-49, 2007.
- [5] Juels A., Sudan M., A fuzzy vault scheme // Proc. IEEE Int'l. Symp. Inf. Theory, p. 408, 2002.
- [6] Li P., Yang X., Cao K., Shi P., Tian J. Security-enhanced fuzzy fingerprint vault, based on minutiae's local ridge information // Advances in Biometrics (ICB 2009), LNCS 5558, pp. 930-939, 2009.
- [7] Nandakumar K., Jain A.K., Pankanti S. Fingerprint-Based Fuzzy Vault: Implementation and Performance. // IEEE Transactions on Information Forensics and Security 2 (4), pp. 744-757, 2007.
- [8] F. Scotti, Cimato S., Gamassi M., Piuri V., Sassi R. Privacy-aware Biometrics: Design and Implementation of a Multimodal Verification System // 2008 Annual Computer Security Applications Conference, IEEE, pp. 130-139.
- [9] Иванов А. И., Фунтиков В. А., Ефимов О. В. Нейросетевая защита биометрических данных пользователя, а так же его личного криптографического ключа при локальной и дистанционной аутентификации // Вопросы защиты информации № 2 (81) 2008 г., с. 25-27.

## Об авторах

Олег Станиславович Ушмаев – ведущий научный сотрудник ИПИ РАН, д.т.н. Его адрес: oushmaev@ipiran.ru.

Кузнецов Владислав Владимирович – младший научный сотрудник ИПИ РАН. Его адрес: ikuvlad@gmail.com.

# Метод Проекционной Фазовой Корреляции в Ключевых Точках Радужной Оболочки Глаза

Е.А. Павельева

Лаборатория Математических Методов Обработки Изображений

Факультет Вычислительной Математики и Кибернетики

Московский Государственный Университет имени М.В. Ломоносова, Москва, Россия

E-mail: [paveljeva@yandex.ru](mailto:paveljeva@yandex.ru)

## Аннотация

Метод проекционной фазовой корреляции применяется в ключевых точках радужной оболочки глаза. Сравнение ключевых точек происходит по локальной информации окрестностей ключевых точек.

Ключевые точки выделяются методом сверток функции интенсивности изображения с выбранной функцией преобразования Эрмита. Для сопоставления ключевых точек вычисляется проекционная фазовая корреляция локальных окрестностей взятых ключевых точек. Корреляция вычисляется путем разложения функций в ряд по функциям Эрмита. При небольших размерах окрестностей ключевых точек предложенный метод позволяет избежать ошибок, связанных с ложным оконтурированием, возникающих в методе фазовой корреляции.

Предложенный метод устойчив к наличию век, ресниц и бликов на изображении, а также к локальным сдвигам частей изображения.

**Ключевые слова:** распознавание по радужной оболочке глаза, фазовая корреляция, функции Эрмита, ключевые точки, биометрия.

## 1. ВВЕДЕНИЕ

Одним из подходов к параметризации данных радужной оболочки является выделение на изображении ключевых точек – точек, в которых рисунок радужной оболочки имеет текстурные особенности [1-4]. Далее происходит сравнение изображений радужных оболочек по ключевым точкам.

В работе [1] ключевые точки выделяются с помощью фильтра Габора. В качестве вектора свойств радужной оболочки вычисляются расстояния между ключевыми точками. При сравнении радужных оболочек сравниваются их векторы свойств. Однако в случае попадания на изображение радужной оболочки бликов, век, ресниц может отсутствовать большое количество ключевых точек радужной оболочки.

В работе [4] ключевые точки выделяются с помощью сверток с выбранной функцией преобразования Эрмита. Сравнение ключевых точек происходит по методу Хэмминга. Изображение разбивается на блоки, и значение блока равняется единице, если там присутствует хотя бы одна ключевая точка и равняется нулю в противном случае. Мерой «похожести» изображений является количество совпадающих блоков. Однако, в случае локальных сдвигов изображений [5] соответствующие друг другу ключевые точки изображений одного глаза могут попадать в соседние блоки. Если же глаза разные, то в одной и той же области

могут быть ключевые точки, соответствующие разным текстурным особенностям.

В данной работе предлагается сравнение ключевых точек с помощью параметров, характеризующих локальные окрестности ключевых точек. При сравнении ключевых точек вычисляется проекционная фазовая корреляция между локальными окрестностями взятых ключевых точек. Это позволяет добиться хороших результатов сравнения и избежать ошибок, связанных с локальными сдвигами частей изображений и с наличием век и ресниц на изображениях.

## 2. ПРЕДОБРАБОТКА ИЗОБРАЖЕНИЙ РАДУЖНЫХ ОБОЛОЧЕК ГЛАЗ

Сначала на изображении радужной оболочки глаза выделяется маска радужной оболочки – области радужной оболочки, не закрытые ресницами, веками и бликами. Далее радужная оболочка переводится в прямоугольное нормализованное изображение (рис. 1) [4].



Рис. 1. Выделение маски радужной оболочки

Перед тем, как проводить параметризацию изображения, необходимо убрать неровности освещенности. Для этого применяется box-фильтр, т.е. значение интенсивности в каждой точке изображения усредняется по окрестности  $16 \times 16$  пикселей (рис. 2(б)). Далее изображение средней интенсивности вычитается из исходного (рис. 2(в)), и контрастность полученного изображения увеличивается с помощью растягивания гистограммы (leveling) с отбрасыванием одного процента низших и высших значений гистограммы (рис. 2(г)).

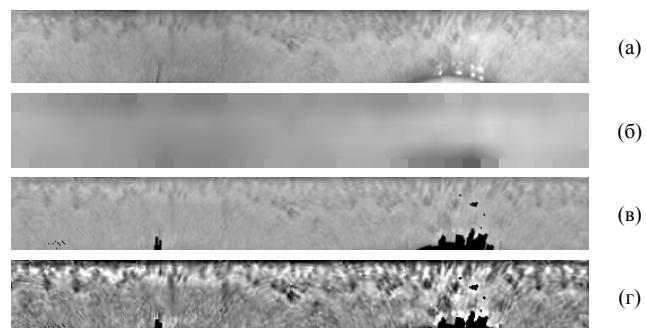


Рис. 2. Выравнивание освещенности на нормализованном изображении: (а) исходное изображение; (б) изображение средней освещенности; (в) изображение с выровненной освещенностью; (г) изображение повышенной контрастности.

### 3. МЕТОД КЛЮЧЕВЫХ ТОЧЕК

После предобработки изображения радужной оболочки, на изображении выделяются ключевые точки при помощи свертки с функцией преобразования Эрмита [4].

Функции преобразования Эрмита [6] задаются формулой:

$$\varphi_n(x, \sigma) = \frac{(-1)^n \cdot e^{-\left(\frac{x}{\sigma}\right)^2}}{\sqrt{2^n \cdot n! \cdot \pi}} \cdot H_n\left(\frac{x}{\sigma}\right), \quad n = 0, 1, 2, 3, \dots, \text{ где}$$

$H_n(x)$  - полиномы Эрмита:

$$H_0(x) = 1, \quad H_1(x) = 2 \cdot x, \\ H_n(x) = 2 \cdot x \cdot H_{n-1}(x) - 2 \cdot (n-1) \cdot H_{n-2}(x).$$

Двумерные функции преобразования Эрмита (рис. 3) можно представить в виде произведения одномерных функций:

$$\varphi_{m,n}(x, y, \sigma_x, \sigma_y) = \varphi_m(x, \sigma_x) \cdot \varphi_n(y, \sigma_y).$$

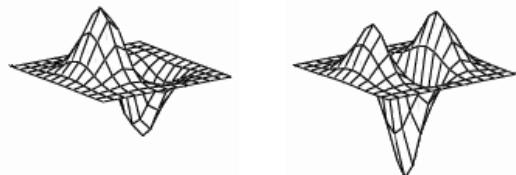


Рис. 3. Функции преобразования Эрмита  $\varphi_{1,0}$  и  $\varphi_{2,0}$ .

Текстура радужной оболочки глаза сильнее изменяется в горизонтальном направлении нормализованного изображения. Поэтому будем рассматривать функции преобразования Эрмита  $\varphi_{m,0}$ .

Сверткой с функцией  $\varphi_{1,0}$  можно выделить точки, лежащие на границе текстурных особенностей изображений (вдоль ОХ), т.е. точки, в которых присутствует скачок интенсивности вдоль горизонтальной составляющей изображения. На рис. 4 черным цветом обозначены точки, в которых свертка функции интенсивности изображения радужной оболочки с функцией  $\varphi_{1,0}$  больше порогового значения  $p = 10$ , а белым цветом – точки, в которых свертка меньше  $-p = -10$ .

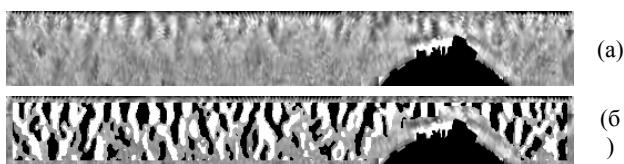


Рис. 4: (а) нормализованное изображение радужной оболочки после предобработки; (б) на изображении выделены области со значениями сверток с функцией  $\varphi_{1,0}(x, y, \sigma_x, \sigma_y)$  больше порогового значения  $p = 10$  и меньше  $-p = -10$ , где  $\sigma_x = 3, \sigma_y = 3$ .

Сверткой (или корреляцией) с функцией  $\varphi_{2,0}$  можно выделить сами текстурные особенности радужной оболочки глаза. На рис. 5 выделены точки, в которых свертка изображения радужной оболочки с функцией  $\varphi_{2,0}$  больше трети максимального значения свертки. На рис. 5 также показано влияние параметра  $\sigma$  на выделение

областей изображения с наибольшими значениями сверток. Далее во всей работе будем считать значение  $\sigma_y = 3$ .

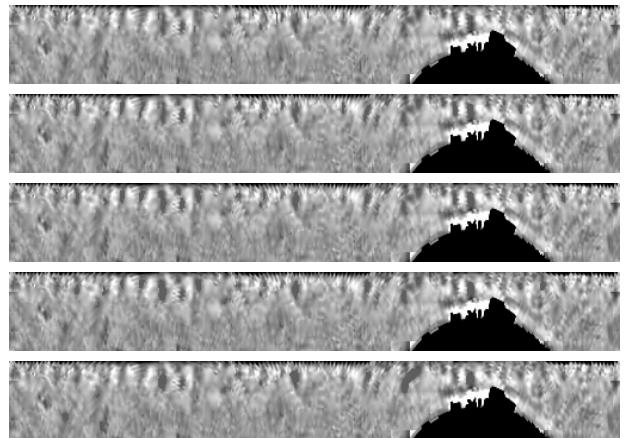


Рис. 5. Выделенные области с наибольшими значениями сверток с функцией  $\varphi_{2,0}(x, y, \sigma_x, \sigma_y)$ , где  $\sigma_y = 3$ , а  $\sigma_x = 1, \sigma_x = 3, \sigma_x = 5, \sigma_x = 7, \sigma_x = 10$  соответственно.

Для нахождения ключевых точек радужной оболочки в каждой точке нормализованного изображения радужной оболочки вычисляется свертка данного изображения с функцией преобразования Эрмита  $\varphi_{2,0}$ :

$$F(x_0, y_0) = (I(x, y) * \varphi_{2,0}(x, y))_{(x_0, y_0)}.$$

Для поиска ключевых точек в данной работе выбираются точки со значениями сверток с функцией преобразования Эрмита  $\varphi_{2,0}(x, y, 3, 3)$  больше трети максимального значения свертки, и отстоящие друг от друга на расстояние не менее  $k = 3$  пикселей (рис. 6). Ключевыми точками считаются  $N$  выбранных точек ( $N \leq 150$ ) с наибольшими значениями сверток.

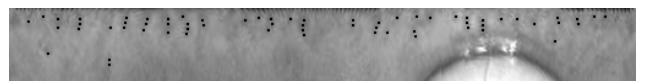


Рис. 6. Ключевые точки на исходном нормализованном изображении, полученные с помощью сверток с функцией преобразования Эрмита  $\varphi_{2,0}(x, y, \sigma_x, \sigma_y)$ , где  $\sigma_x = 3, \sigma_y = 3$ .

### 4. МЕТОД ПРОЕКЦИОННОЙ ФАЗОВОЙ КОРРЕЛЯЦИИ

Метод фазовой корреляции [7, 8] вычисляет меру «похожести» изображений. Для двух дискретных изображений  $f(x, y)$  и  $g(x, y)$ , размера  $M \times N$  вычисляются их преобразования Фурье  $F(u, v)$  и  $G(u, v)$ . Далее для спектральных функций  $F(u, v)$  и  $G(u, v)$  вычисляется их взаимный фазовый спектр

$$R_{FG}(u, v) = \frac{F(u, v) \overline{G(u, v)}}{|F(u, v) \overline{G(u, v)}|} = e^{i(\varphi_F(u, v) - \varphi_G(u, v))},$$

т.е. спектральная функция с единичным модулем, фаза которой равна разности фаз функций  $F$  и  $G$ . Далее от функции взаимного фазового спектра вычисляется обратное преобразование Фурье – получается функция фазовой корреляции (РОС-функция):

$$POC_{fg}(x, y) = F^{-1}[R_{FG}(u, v)].$$

В случае «похожих» изображений РОС-функция дает четкий пик, высота пика определяет меру «похожести» изображений, а положение пика соответствует смещению одного изображения относительно другого. Если же функции «не похожи», то РОС-функция не дает четкого пика.

В методе проекционной фазовой корреляции [5] прямое и обратное преобразования Фурье производятся путем разложения исходных непрерывных функций интенсивности изображений по полной ортонормированной в  $L_2(R_2)$  системе функций Эрмита  $\psi_{m,n}(x, y)$ , являющихся собственными функциями непрерывного преобразования Фурье [9].

Одномерные функции Эрмита задаются следующей формулой:

$$\psi_n(x, \bar{\sigma}) = \frac{1}{\sqrt{2^n \cdot n! \cdot \sqrt{\pi}}} \cdot e^{-\frac{x^2}{2\bar{\sigma}^2}} \cdot H_n\left(\frac{x}{\bar{\sigma}}\right), \quad n = 0, 1, 2, 3, \dots$$

Двумерные функции Эрмита можно представить в виде произведения одномерных:

$$\psi_{m,n}(x, y, \bar{\sigma}_x, \bar{\sigma}_y) = \psi_m(x, \bar{\sigma}_x) \cdot \psi_n(y, \bar{\sigma}_y).$$

Пусть исследуемые функции  $f(x, y)$  и  $g(x, y)$  заданы в прямоугольнике  $D = [-A, A] \times [-B, B]$ , продолжим их нулем в  $R_2 \setminus D$  и рассмотрим частичную сумму разложения в ряд по функциям Эрмита  $\psi_{ij}(x, y)$ ,  $i = \overline{0, m}$ ,  $j = \overline{0, n}$

$$f(x, y) \approx \tilde{f}(x, y) = \sum_{i=0}^m \sum_{j=0}^n c_{ij} \psi_{ij}(x, y, \bar{\sigma}_x, \bar{\sigma}_y), \quad (1)$$

где коэффициенты Эрмита

$$c_{i,j} = \iint_{R_2} f(x, y) \psi_{i,j}(x, y, \bar{\sigma}_x, \bar{\sigma}_y) dx dy.$$

Далее для приближенных функций  $\tilde{f}(x, y)$  и  $\tilde{g}(x, y)$  применяется метод фазовой корреляции. Функции Эрмита являются собственными функциями непрерывного преобразования Фурье с собственными значениями  $\pm 1, \pm i$ :  $F(\psi_n) = (-i)^n \psi_n$ . Следовательно, аппроксимация преобразования Фурье имеет следующий вид:

$$F[\tilde{f}] = F \left[ \sum_{i=0}^m \sum_{j=0}^n c_{ij} \psi_{ij}(x, y) \right] = \sum_{i=0}^m \sum_{j=0}^n c_{ij} (-i)^{i+j} \psi_{ij}(x, y).$$

Для подсчета функции проекционной фазовой корреляции  $HPOC_{fg}(x, y) = F^{-1}[R_{FG}(u, v)]$  используется следующее свойство:  $f(x, y) = F^{-1}[F(u, v)] = F[F(-u, -v)]$ .

В работе [5] было показано, что при небольших размерах изображений метод проекционной фазовой корреляции дает более точные результаты, чем метод фазовой корреляции. Это связано с тем, что метод НРРОС не требует периодического продолжения изображений для вычисления преобразования Фурье (в отличие от метода РОС), и поэтому в методе НРРОС не возникает ошибок от скачков на границах (ложного оконтурирование). В случае изображений маленького размера ошибки в

методе РОС, связанные с ложным оконтурированием, могут приводить к неверным результатам.

## 5. СРАВНЕНИЕ ИЗОБРАЖЕНИЙ РАДУЖНЫХ ОБОЛОЧЕК ГЛАЗ В КЛЮЧЕВЫХ ТОЧКАХ МЕТОДОМ ПРОЕКЦИОННОЙ ФАЗОВОЙ КОРРЕЛЯЦИИ

В данной работе предложен метод сравнения изображений радужных оболочек глаз, основанный на анализе фазовой корреляции окрестностей ключевых точек радужных оболочек.

В каждой ключевой точке выделяется локальная окрестность размера  $8\sigma_x \times 8\sigma_y$  пикселей с центром в данной точке, т.е. окрестность, дважды превышающая по длине и ширине область локализации функции преобразования Эрмита  $\varphi_{2,0}(x, y, \sigma_x, \sigma_y)$ . Данное изображение раскладывается в сумму (1) по функциям Эрмита  $\psi_{i,j}$ .

Возникает вопрос о выборе числа функций Эрмита ( $m$  и  $n$ ) и значений  $\bar{\sigma}_x$  и  $\bar{\sigma}_y$  для разложения. Возьмем  $\bar{\sigma}_x = \sigma_x$ ,  $\bar{\sigma}_y = \sigma_y$ . По теореме Сонина [10] функция Эрмита  $\psi_n(x, \bar{\sigma})$  содержит все свои точки перегиба на отрезке  $[-\bar{\sigma}\sqrt{2n+1}, \bar{\sigma}\sqrt{2n+1}]$ . Отрезок  $[-\bar{\sigma}(\sqrt{2n+1}+1), \bar{\sigma}(\sqrt{2n+1}+1)]$  будем подразумевать под областью локализации функции  $\psi_n(x, \bar{\sigma})$ . Тогда необходимое число  $m$  функций Эрмита, локализованных на отрезке  $[-2\bar{\sigma}(\sqrt{2n+1}+1), 2\bar{\sigma}(\sqrt{2n+1}+1)] = [-\bar{\sigma}(\sqrt{2m+1}+1), \bar{\sigma}(\sqrt{2m+1}+1)]$  длины  $m = 4n + 2 + 2\sqrt{2n+1}$ . Таким образом, для параметризации ключевых точек берутся функции Эрмита с номерами  $m = 0, 1, \dots, 15$ ,  $n = 0, 1, 2, 3, 4$ .

Далее каждая ключевая точка одного изображения сопоставляется с каждой ключевой точкой второго изображения из окрестности 40 на 5 пикселей, т.е. считается, что возможен поворот глаза на угол до  $\pm 28^\circ$ . Сопоставление происходит методом проекционной фазовой корреляции окрестностей данных ключевых точек. Если отношение пика НРРОС-функции ко второму локальному максимуму этой функции больше значения  $k=3$ , то считается, что ключевые точки соответствуют друг другу. Пример алгоритма сопоставления ключевых точек показан на рис. 7-9.

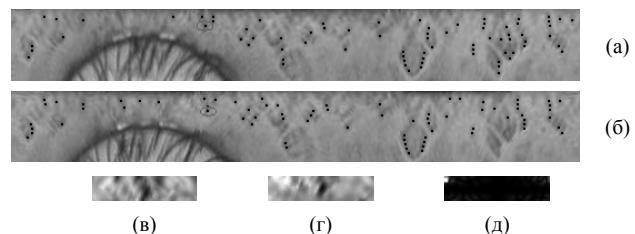


Рис. 7. Сопоставление ключевых точек двух изображений одного глаза методом проекционной фазовой корреляции: (а), (б) исходные нормализованные изображения. Выделены сравниваемые две ключевые точки; (в) окрестность (увеличенная) ключевой точки первого изображения; (г) окрестность ключевой точки второго изображения; (д) НРРОС-функция для данных окрестностей.

В случае изображений одного глаза (рис. 7) для сопоставляемых ключевых точек есть четкий пик НПРОС-функции. В случае изображений разных глаз (рис. 8) для сопоставляемых ключевых точек нет четкого пика НПРОС-функции.

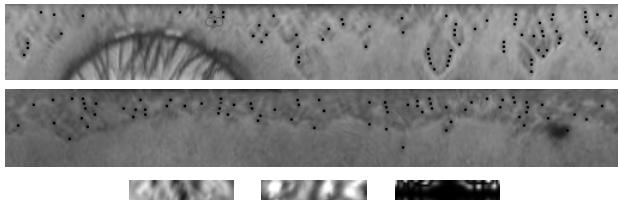
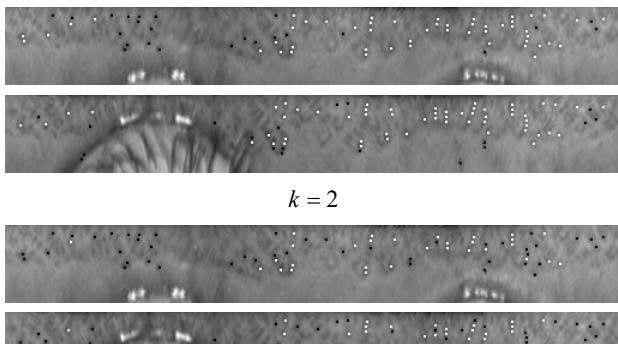


Рис. 8. Сопоставление ключевых точек (близких по месторасположению) изображений разных глаз методом проекционной фазовой корреляции.

На рис. 9 приведены два изображения одного глаза и приведено их сопоставление методом проекционной фазовой корреляции. На рисунке черным цветом обозначены ключевые точки, не имеющие сопоставимые ключевые точки на другом изображении. Белые точки – точки, имеющие сопоставимые ключевые точки на другом изображении. На рисунке также показано, что чем меньше порог  $k$  отношения пика НПРОС-функции ко второму локальному максимуму этой функции, тем больше ключевых точек будут являться сопоставимыми.



$k = 3$  – используется при анализе результатов работы метода в разделе 6.

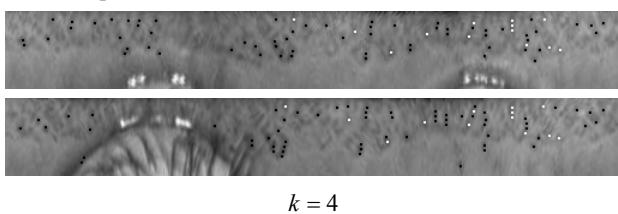


Рис. 9. Сопоставление ключевых точек изображений одного глаза методом проекционной фазовой корреляции при различных порогах  $k$ .

При сравнении ключевых точек методом Хэмминга [4] в случае локальных сдвигов изображений соответствующие друг другу ключевые точки изображений одного глаза могут попадать в соседние блоки. В случае попадания на изображение радужной оболочки бликов, век, ресниц может отсутствовать большое количество ключевых точек, присутствующих на другом изображении той же радужной оболочки (рис. 10). Если же глаза разные, то в одной и той же области могут быть ключевые точки, но соответствующие разным текстурным особенностям.

На рис. 10 приведены те же изображения, что и на рис. 9, и приведено сопоставление ключевых точек этих изображений методом Хэмминга – сопоставление положения ключевых точек второго изображения по сравнению с положением ключевых точками первого изображения. Ключевые точки первого изображения обозначены черным цветом, второго – белым. Разобъем приведенные изображения на три части. Видно, что в правой части изображений ключевые точки на двух изображениях соответствуют друг другу. В центре изображений заметно смещение ключевых точек изображений (это связано с локальным смещением центральных частей изображения). Слева на втором изображении присутствует веко, поэтому там отсутствуют ключевые точки, присутствующие на первом изображении.

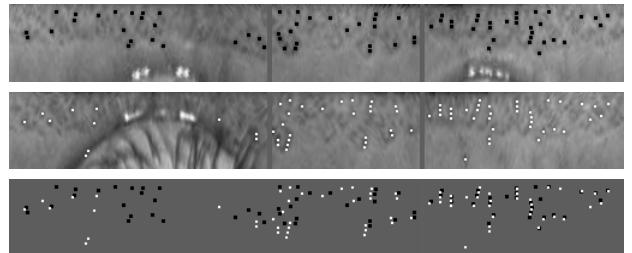


Рис. 10. Сопоставление ключевых точек изображений одного глаза методом Хэмминга.

## 6. РЕЗУЛЬТАТЫ

Для оценки качества предложенного метода исследовались изображения глаз базы данных CASIA-IrisV3 [11]. В данной работе используется часть базы данных CASIA, содержащая изображения глаз первых 20 людей в базе: 224 глаза, образующих 40 множеств  $M_1, \dots, M_{40}$  (каждое множество содержит изображения одного из двух глаз одного из 20 людей). В этой базе данных для каждой пары изображений сопоставляются ключевые точки методом проекционной фазовой корреляции. Мерой «похожести» изображений является количество совпадающих (с точки зрения метода проекционной фазовой корреляции) ключевых точек.

Результаты работы алгоритма приведены в таблице 1 и на рис. 11.

- В первой строке таблицы – мера «похожести»  $d$  изображений, т.е. количество совпадающих ключевых точек при сравнении изображений.
- Во второй (третьей) строке – количество пар изображений одного глаза (разных глаз), имеющих данную меру похожести.
- В четвертой (пятой) строке – величина ошибки I (II) рода, измеренная в процентах, при данной мере «похожести»  $d$ .

$d$	0	1	2	3	4	5	...
<b>inner</b>	0	0	4	2	15	13	
<b>outer</b>	42615	4771	362	22	2	0	0
<b>FAR (%)</b>	0	0	0.18	0.27	0.96	1.5	
<b>FRR (%)</b>	10.7	0.8	0.05	0.004	0	0	0

Таблица 1. Результаты работы алгоритма.

Для оценки качества метода используется анализ величины EER (Equal Error Rate), где EER – величина ошибки работы метода, при которой ошибки I и II рода равны. Для данного метода величина EER примерно равна 0.1% при мере «похожести»  $d = 2$ .

На рис. 11 по оси ОХ отложена мера «похожести»  $d$  изображений, по оси ОY – количество пар изображений одного глаза, имеющих данную меру «похожести» (значения из второй строки таблицы) [12].

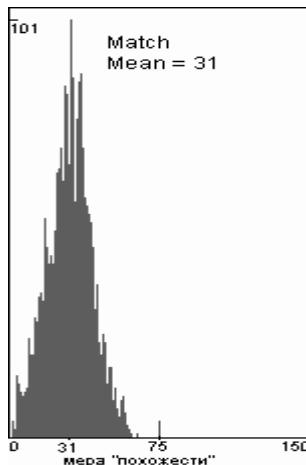


Рис. 11. Результаты работы алгоритма.

## 7. ЗАКЛЮЧЕНИЕ

В работе метод проекционной фазовой корреляции применяется в ключевых точках радужной оболочки глаза. Таким образом, сравнение ключевых точек происходит по локальной информации окрестностей ключевых точек. В методе не используются никакие соответствия между ключевыми точками одного изображения, связанные с их взаиморасположением. Поэтому предложенный метод устойчив к наличию век, ресниц и бликов на изображении, а также к локальным сдвигам частей изображения.

Работа выполнена при поддержке гранта Президента РФ МД-72.2001.9 и гранта РФФИ 10-07-00433.

## 8. СПИСОК ЛИТЕРАТУРЫ

- [1] L. Yu, D. Zhang, K. Wang. *The relative distance of key point based iris recognition*, Pattern Recognition, vol. 40, №2, p. 423-430, 2007.
- [2] 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.
- [3] L. Ma, T. Tan, Y. Wang, and D. Zhang. *Efficient iris recognition by characterizing key local variations*, IEEE Transaction on Image Processing, vol. 13, no. 6, p. 739-750, 2004.
- [4] Е. А. Павельева, А. С. Крылов. *Алгоритм сравнения изображений радужной оболочки глаза на основе ключевых точек*, Информатика и её применения, т.5, вып.1, 2011, с.68-72
- [5] Е. А. Павельева, А. С. Крылов. *Определение локальных сдвигов изображений радужных оболочек глаз методом проекционной фазовой корреляции*, труды конференции GraphiCon'2011, Москва, 2011, с. 188-191.
- [6] J.-B. Martens. *The Hermite transform-theory*, IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 38, no. 9, p. 1595-1606, 1990.

[7] K. Miyazawa, K. Ito, T. Aoki, K. Kobayashi, H. Nakajima. *A Phase-Based Iris Recognition Algorithm*, LNCS (ICB 2006), No. 3832, p. 356-365, 2006.

[8] S. Nagashima, K. Ito, T. Aoki, H. Ishii, K. Kobayashi. *High Accuracy Estimation of Image Rotation using 1D Phase-Only Correlation*, IEICE Trans. Fund. v.E92-A, p.235 243, 2009.

[9] A. Krylov, D. Korchagin. *Fast Hermite Projection Method*, LNCS, v.4141, p.329-338, 2006.

[10] П.К. Суетин. *Классические ортогональные многочлены*, издат. Наука, Москва, изд. 2, 1979.

[11] CASIA-IrisV3 database.  
<http://www.cbsr.ia.ac.cn/IrisDatabase.htm>.

[12] J. Daugman. *How iris recognition works*, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 14, No. 1, P. 21-30, 2004.

## Hermite Projection Phase-Only Correlation Method in Iris Key Points

### Abstract

The Hermite projection phase-only correlation method in iris key points has been proposed. The local information of areas around key points is used for key points matching.

The iris key points are selected using the Hermite transform. The Hermite projection phase-only correlation of areas around taken key points is calculated for key points matching. The correlation is calculated using Hermite projection method of expansion of intensity functions into series of Hermite functions. In case of small local images the proposed method allows avoid errors of Phase-Only Correlation method like Gibbs effect.

The proposed method is robust to eyelids, eyelashes, glares and local shifts of parts of images.

**Keywords:** *iris recognition, phase only correlation, Hermite functions, key points, biometrics.*

### Об авторе

Павельева Елена Александровна – ассистент кафедры математической физики факультета ВМК МГУ.

### About the author

Elena A. Pavelyeva is an assistant of Chair of Mathematical Physics of Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University. E-mail: [paveljeva@yandex.ru](mailto:paveljeva@yandex.ru)

# Модификация алгоритма классификации возраста людей по изображению лица на основе метода ранжирования

Андрей Рыбинцев, Татьяна Лукина, Вадим Конушин, Антон Конушин

Лаборатория компьютерной графики и мультимедиа

Факультет вычислительной математики и кибернетики

МГУ имени М.В. Ломоносова, Москва, Россия

{arybintsev, tlukina, vadim, ktosh}@graphics.cs.msu.ru

## Аннотация

В этой работе рассматривается задача определения возраста человека по изображению лица. Предлагается ряд модификаций для алгоритма на основе множества бинарных классификаций. Экспериментальная проверка модифицированного алгоритма на тестовой базе MORPH показала уменьшение средней ошибки определения возраста с 6.07 до 4.52 и 5 лет при времени классификации 3.21 и 0.32 сек. на одно изображение в зависимости от используемых параметров.

**Ключевые слова:** Классификация возраста человека по изображению лица, Локальные бинарные шаблоны, Метод опорных векторов.

## 1. ВВЕДЕНИЕ

Интерес к задаче классификации возраста человека по изображению лица в последние годы неуклонно растет. Одной из главных причин является стремительное увеличение числа изображений в Интернете и необходимость их автоматической аннотации для задач поиска. Человек является самым распространенным объектом, запечатленным на фотографиях, и автоматическая оценка возраста человека может помочь фильтровать выдаваемые поисковиком результаты.

Другим применением автоматической классификации возраста является анализ возрастного состава аудитории, что полезно для анализа эффективности рекламы.

В будущем классификация возраста также найдёт своё место при создании так называемых «аффективных» человеко-машинных интерфейсов, в которых поведение системы будет подстраиваться под конкретного пользователя, ориентируясь, в числе прочего, и на его возраст.

В этой работе предлагается ряд модификаций для одного из лучших на данный момент алгоритмов оценки возраста человека по фотографии, взятого из статьи Chang et. al. «Ordinal Hyperplanes Ranker with Cost Sensitivities for Age Estimation» [4]. Для повышения точности алгоритма мы предлагаем заменить использованный в исходной работе дескриптор на локальные бинарные шаблоны. Поскольку это также приводит к существенному увеличению размера классификаторов и уменьшению скорости работы алгоритма, мы предлагаем ряд модификаций для решения этой проблемы.

## 2. СУЩЕСТВУЮЩИЕ МЕТОДЫ

Большинство алгоритмов классификации возраста человека по изображению лица состоят из трёх шагов: нормализация изображения лица, вычисление дескриптора,



**Рис. 1.** Примеры изображений из базы MORPH Album 2. классификация. Различаются алгоритмы в первую очередь реализацией отдельных шагов.

На шаге нормализации стандартным решением является поворот, масштабирование и обрезание изображения таким образом, чтобы глаза (уголки или центры глаз) находились в заданном положении на одной горизонтальной линии. Другим методом нормализации является использование аффинного преобразования и деформации на основе триангуляции Делоне [16]. Для повышения устойчивости к точности выделения центров глаз и других антропометрических точек в тренировочную выборку можно добавлять немного смещенные изображения с неточной нормализацией [12].

В качестве признаков изображения могут использоваться непосредственно интенсивности пикселей, чаще всего с последующим понижением размерности, например, с помощью метода проекций, сохраняющих свойство локальности (Locality Preserving Projections) [6]; геометрические признаки, например, расстояния между антропометрическими точками [13]; параметры «Активной модели внешности» (Active Appearance Models) [7], локальные бинарные шаблоны [11]. В [9] были предложены так называемые «биологически обусловленные признаки» (Biologically Inspired Features, BIF), использование которых позволило достичь одни из лучших результатов на задаче классификации пола и возраста.

Для собственно классификации возраста обычно используется либо регрессия [9], позволяющая оценивать точный возраст человека, либо многоклассовая классификация [8, 9], предсказывающая возрастную группу человека. В качестве алгоритмов классификации используют нейронные сети [10], случайный лес деревьев [13]. Наиболее широкое применение нашел в этой задаче метод опорных векторов [3, 4, 7, 8, 9].

В статье [8] предложили вначале классифицировать пол и примерный возрастной диапазон человека, после чего применять специализированный классификатор возраста

для данной половозрастной группы. Авторы статьи показали, что такая двухступенчатая процедура заметно повышает точность итоговой классификации.

В [3] задача классификации возраста человека была сведена к набору бинарных классификаторов. Для каждого возраста строится классификатор, который определяет, старше ли человек, изображенный на тестовом изображении, чем заданный возраст, или младше. Возраст человека определяется как число классификаторов, выдавших положительный результат. Предложенная схема показала более высокую точность классификации, чем существующие аналоги.

В [4] данный метод получил развитие. В частности, в [3] использовался метод опорных векторов, и бинарные классификаторы отличались между собой только сдвигом разделяющей поверхности, в то время как в [4] бинарные классификаторы обучались независимо друг от друга, и в каждом классификаторе могло использоваться своё ядро. Также было предложено вместо минимизации ошибки классификации минимизировать цену ошибки. Для этого для каждой из двух используемых метрик качества алгоритма были предложены специальные функции расчёта цены ошибки.

### 3. БАЗОВЫЙ АЛГОРИТМ

По результатам проведенного обзора существующих методов мы выбрали для последующей модификации алгоритм Chang et. al. «Ordinal Hyperplanes Ranker with Cost Sensitivities for Age Estimation» [4] как показавший один из лучших результатов. Поэтому в данной секции опишем его подробнее.

Обозначим  $i$ -ый тренировочный пример из обучающей выборки как  $x_i$ , а через  $y_i$  - соответствующий ему возраст, где  $y_i \in \{1, \dots, K\}$ ,  $K$  - максимальный возраст. Конкретное значение  $K$  зависит от используемой базы, в [4] взят равным 80.

Для каждого возраста  $k$  построим 2 выборки  $X_k^+$  и  $X_k^-$  следующим образом:

$$\begin{aligned} X_k^+ &= \{(x_i, +1) \mid y_i > k\} \\ X_k^- &= \{(x_i, -1) \mid y_i \leq k\} \end{aligned} \quad (1)$$

По этим выборкам  $X_k^+$  и  $X_k^-$  обучим бинарный классификатор, который будет отвечать на вопрос, больше заданному человеку, чем  $k$  лет, или нет.

Т.к. на практике важность, т.е. цена, разных ошибок отличается, то вводится функция стоимости  $\text{cost}_k(l)$ , где  $k, l = 1 \dots K$ . Эта функция показывает цену неправильной классификации примера возраста  $l$  для  $k$ -го классификатора и зависит от используемой меры качества итогового алгоритма.

Наиболее часто применяемой мерой качества в задаче определения возраста является MAE (mean absolute error), которая определяется следующим образом:

$$MAE = \sum_{j=1}^N |\bar{y}_j - y_j| / M, \quad (2)$$

где  $\bar{y}_j$  - предсказанный возраст,  $y_j$  - истинный возраст,  $M$  - количество тестовых изображений.

Для этой меры в качестве функции стоимости предлагается использовать модуль разности между возрастами:

$$\text{cost}_k(l) = |l - k|, \quad (2)$$

т.к. чем больше разница между истинным и предсказанным возрастом, тем больше вклад в итоговую ошибку.

Второй широко используемой мерой качества является суммарный счёт («cumulative score»), который показывает число примеров с ошибкой предсказания ниже заданного уровня:

$$CS(L) = (M_{e < L} / M) * 100\%,$$

где  $M_{e < L}$  - количество тестовых примеров с ошибкой предсказания  $e$  меньше  $L$  лет.

Т.к. для этой меры разница до  $L$  лет не вносит вклад в общую ошибку, то можно в функции стоимости сделать цену такой ошибки нулевой, а цену всех остальных ошибок одинаковой (например, единичной):

$$\text{cost}_k(l) = \begin{cases} 0, & \text{if } (l - L) \leq k \leq (l + L) \\ 1, & \text{otherwise} \end{cases} \quad (3)$$

Для обучения классификатора используется метод опорных векторов с RBF ядром. Вес каждого примера из обучающей выборки задаётся с помощью функции стоимости ошибки.

Для определения возраста человека нужно применить все  $K - 1$  классификаторов, после чего итоговая оценка считается равной числу положительных откликов классификаторов плюс один:

$$r(x) = 1 + \sum_{k=1}^{K-1} \|f_k(x) > 0\|, \quad (4)$$

где  $f_k(x)$  - выход  $k$ -го классификатора,  $\|\cdot\|$  равна 1, если внутреннее условие верно, и 0 иначе.

В качестве признаков используются параметры Активной модели внешности [5].

Для тестирования алгоритма использовались база FG-NET [1] и часть большей по размеру базы MORPH [15]. Примеры изображений из базы MORPH приведены на рисунке 1.

Используемая часть базы MORPH содержит 5492 изображения людей в возрасте от 16 до 77 лет. Во время тестирования используемая выборка случайным образом 30 раз разбивалась на две части - 80% данных использовались для обучения и 20% для тестирования. Для подбора параметров на обучающей выборке использовался метод кросс-валидации с разбиением на 5 сегментов. Результаты усреднялись по всем 30 разбиениям. Итоговый алгоритм показал ошибку MAE в 6.07 лет.

## 4. ПРЕДЛОЖЕННЫЕ МОДИФИКАЦИИ

### 4.1 Локальные бинарные шаблоны

Вместо используемых в [4] параметров Активной модели внешности мы предлагаем использовать в качестве признаков локальные бинарные шаблоны [14], хорошо себя

зарекомендовавшие в задачах распознавания человека по изображению лица.

Локальные бинарные шаблоны вычисляются путем применения определенного оператора к каждому пикселию изображения. Этот оператор работает следующим образом. Вначале значение интенсивности в пикселе сравнивается со значениями во всех пикселях из некоторой окрестности, например, размером  $3 \times 3$  пикселя. Результат сравнения записывается как 0, если значение рассматриваемого пикселя меньше центрального, и как 1 в противном случае. Для рассматриваемой окрестности 3 на 3 получается 8 цифр, из которых составляется двоичный вектор, который интерпретируется как двоичная запись целого числа. Это число и является результатом применения оператора к пикселию. Итоговые признаки получаются после разбиения всего изображения решеткой на прямоугольные области, подсчета гистограмм частот появления чисел в каждой области и конкатенации гистограмм по всем областям в один вектор. Процесс вычисления локальных бинарных шаблонов показан на рисунке 2.

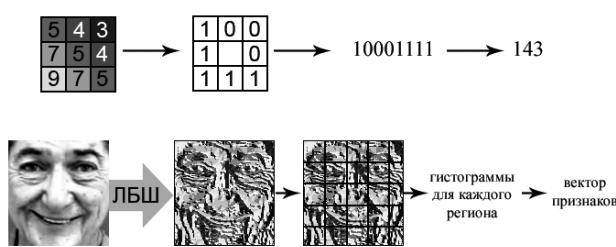


Рис. 2. Вычисление локальных бинарных шаблонов: вверху – для одного пикселя, внизу – для целого изображения.

Эти признаки устойчивы к небольшим изменениям освещенности и небольшим сдвигам в положении лица. Эта устойчивость достигается за счет того, что подсчет ведется не индивидуально для каждого пикселя, а используются области значительного размера.

При замене в базовом алгоритме используемых признаков на локальные бинарные шаблоны, а также использовании линейного метода опорных векторов, ошибка MAE алгоритма на выборке из базы MORPH понизилась с 6.07 до 4.61. При этом использовалась точно такая же процедура тестирования, как и в [4].

В последующих экспериментах мы использовали упрощенную процедуру тестирования. Параметры модели, такие как параметры метода опорных векторов, подбирались с помощью кросс-валидации с 5 разбиениями однократно на всей выборке. После этого вся выборка 30 раз случайным образом делилась на 80% и 20%, на большей части обучались классификаторы, а тестировались на 20%. Данная процедура существенно быстрее исходной, поэтому она лучше подходит для тестирования различных модификаций алгоритма.

Основными недостатками исходного алгоритма являются высокие требования по памяти и длительное время классификации из-за большого количества независимых классификаторов. В работе [4] использовалось 80 классификаторов. При замене параметров Активной модели внешности на локальные бинарные шаблоны данная проблема значительно обостряется, т.к. размер

вектор-признака увеличивается в несколько раз, с нескольких десятков или сотен параметров<sup>1</sup> до 2065.

#### 4.2 Повышение точности классификации

Далее нами были предложены две модификации, дающие дополнительный прирост в точности классификации.

Первая модификация заключается в использовании вероятностного выхода классификатора. В базовом алгоритме подсчитывается число классификаторов, которые выдают результат больше 0, т.е. всех тех, которые считают человека старше, чем сравниваемый возраст. Данная схема была бы правильной, если бы все первые классификаторы выдавали бы положительный результат, а после определенного возраста – отрицательный. Однако на практике в окрестности правильного возраста человека классификаторы могут давать разный результат. Например, среди соседних отрицательных результатов может встречаться и положительный.

Поэтому мы предлагаем суммировать непосредственно вероятностный выход всех классификаторов:

$$r'(x) = 1 + \sum_{k=1}^{K-1} f_k(x)$$

Предварительно к вероятностным выходам классификаторов применяется медианная фильтрация.

Также мы предлагаем изменить функцию стоимости. Исходная функция стоимости (2) линейно зависит от ошибки в оценке возраста, вне зависимости от значения возраста. При этом одна и та же разница в возрасте для детей обычно гораздо заметнее, чем для пожилых людей. Поэтому мы предложили модифицированный вариант этой функции:

$$\text{cost}_k(l) = \alpha \cdot (K - k) \cdot |l - k|,$$

где  $K$  – максимальный возможный возраст,  $\alpha$  – настраиваемый параметр.

Суммарно, указанные модификации уменьшают ошибку MAE до 4.52 года.

#### 4.3 Ускорение алгоритма

Как было сказано выше, основным недостатком используемого алгоритма является большой размер классификатора и медленная классификация.

Наиболее прямолинейным решением является непосредственное уменьшение числа классификаторов. Например, можно оставить лишь часть классификаторов, различающихся на  $n$  лет друг от друга. Это автоматически влечет за собой уменьшение размера общего классификатора и ускорение времени обработки в  $n$  раз. В таблице 1 указана точность алгоритма, использующего все модификации из пункта 4.2, если оставлять лишь каждый  $n$ -ый классификатор.

Разумно было бы предположить, что оптимальный набор возрастов, классификаторы которых стоит оставить при заданном количестве классификаторов, отличается от простого выбора каждого  $n$ -ого. Для выбора оптимального набора возрастов мы исследовали применимость жадного алгоритма, который по одному «выкидывает» отдельные классификаторы так, чтобы ошибка классификации оставалась минимальной. Однако это приводит к

<sup>1</sup> Точный размер вектор-признака в [4] не указан.

n = 1 (все классификаторы)	n = 2	n = 3	n = 4	n = 5
4,52	4,61	4,65	4,72	4,96

**Таблица 1.** Средняя ошибка классификации возраста при использовании каждого n-го классификатора

переобучению, т.к. алгоритм естественным образом выкидывает те возрасты, для которых в выборке меньше всего фотографий. Таким образом, жадный алгоритм можно применять только при наличии полностью равномерной выборки.

Также для ускорения классификации возраста можно применять не все классификаторы, а лишь до того момента, когда n последних классификаторов не дадут отрицательный результат. Суммировать результаты нужно также только по примененным классификаторам. При этом нельзя использовать вероятностные выходы, и приходится округлять их до 0 и 1. В таблице 2 приведены результаты работы алгоритма при различных n.

n=1	n = 2	n = 3	n = 4	n = 5
4,75	4,7	4,65	4,63	4,62

**Таблица 2.** Средняя ошибка классификации возраста при остановке классификации после n отрицательных результатов

Таким образом, при остановке после 3 отрицательных результатов средняя ошибка увеличилась до 4.65, при этом среднее время классификации уменьшилось более чем в 2 раза. Однако уменьшение времени происходит по-разному для различных возрастов. Для старших возрастов уменьшение времени меньше, т.к. приходится применять большее количество классификаторов.

Если же скомбинировать описанные оптимизации вместе, например, используя лишь каждый 3-й классификатор и остановку классификации после 2 отрицательных результатов, можно добиться 8-кратного ускорения и 3-кратного уменьшения размера классификатора при увеличении ошибки определения возраста до 4.72.

Наконец, еще одним способом ускорения является использование бинарного поиска. При этом вначале проверяется средний возраст. Если классификатор даёт положительный результат, то проверяется старшая половина возрастов, если отрицательный - то младшая половина. При этом также можно использовать не все классификаторы, а только каждый n-ый. Проведенные эксперименты показали, что при этом можно в 3 раза уменьшить классификатор и в 10 раз снизить время классификации при увеличении средней ошибки до 5 лет.

#### 4.4 Выводы

Наибольшей точности определения возраста удалось достичь при использовании локальных бинарных шаблонов, суммирования вероятностных выходов и модифицированной функции стоимости.

Компромисс между увеличением точности и временем классификации можно обеспечить за счёт использования каждого третьего классификатора и остановки после 2 отрицательных результатов (обозначим этот алгоритм как «ускоренный вариант 1») или за счёт замены остановки после 2 отрицательных результатов на бинарный поиск («ускоренный вариант 2»).

В таблице 3 показано сравнение указанных вариантов алгоритмов друг с другом и с исходным алгоритмом. Поскольку в исходной работе [4] не указан размер классификаторов и времена работы, эти параметры в таблице не приводятся. Время указано для реализации алгоритма на Матлабе, непосредственно классификация реализована на C++ в виде тех функций. Для классификации использовалась библиотека lib-svm [2].

Алгоритм	Размер классификатора, Гб	Время классификации, с.	MAE, лет
Исходный алгоритм	-	-	6.07
Лучший по точности	1,17	3,21	4.52
Ускоренный вариант 1	0.29	0,40	4.72
Ускоренный вариант 2	0,29	0,32	5

**Таблица 3.** Сравнение различных модификаций алгоритма на основе бинарных классификаторов.

## 5. ЗАКЛЮЧЕНИЕ

В данной статье предлагается ряд модификаций алгоритма [4] определения возраста человека по фотографии лица. Благодаря замене параметров Активной модели внешности на локальные бинарные шаблоны в качестве признаков достигнуто существенное повышение точности классификации возраста, однако при этом алгоритм становится медленнее и существенно более требовательным к объему необходимой памяти. Для обеспечения компромисса между повышением точности, временем работы и требованиями по памяти предложено несколько дополнительных модификаций.

В дальнейшем планируется исследовать возможность повысить точность классификации за счёт использования дополнительных признаков, поиска оптимального разбиения на возрастные группы, а также провести тестирование алгоритма на базе FG-NET [1].

## 6. БЛАГОДАРНОСТИ

Работа была выполнена при частичной поддержке гранта РФФИ № 11-01-00957-а.

## 7. ССЫЛКИ

- [1] The FG-NET aging Database, available at <http://www.fgnet.rsunit.com/>.
- [2] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. <http://csie/wiki.org/~cjlin/libsvm>, 2001.
- [3] K.-Y. Chang, C.-S. Chen, and Y.-P. Hung. A ranking approach for human age estimation based on face images. In *ICPR*, 2010.
- [4] K.-Y. Chang, C.-S. Chen, and Y.-P. Hung. Ordinal Hyperplanes Ranker with Cost Sensitivities for Age Estimation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 585–592, 2011.
- [5] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE TPAMI*, 2001.

- [6] Y. Fu, Y. Xu, and T. S. Huang. Estimating human ages by manifold analysis of face pictures and regression on aging features. In *Proc. IEEE Conf. Multimedia Expo.*, pp. 1383–1386, 2007.
- [7] G. Guo, Y. Fu, T. S. Huang, and C. Dyer. A probabilistic fusion approach to human age prediction. In *IEEE CVPR-SLAM workshop*, 2008.
- [8] G. Guo, G. Mu, Y. Fu, C. Dyer, and T.S. Huang. A Study on Automatic Age Estimation using a Large Database. *Proc. of ICCV*, pp. 1986-1991, 2009
- [9] G. Guo, G. Mu, Y. Fu, and T. Huang. Human age estimation using bio-inspired features. In *CVPR*, 2009.
- [10] A. Lanitis, C. Draganova, and C. Christodoulou. Comparing different classifiers for automatic age estimation. *IEEE TSMC-Part B*, 2004.
- [11] H.-C. Lian and B.-L. Lu. Multi-view gender classification using local binary patterns and support vector machines. In *Proc. 3rd Internat. Sympos. on Neural Networks (ISNN'06)*, Chengdu, China, vol. 2, pp. 202–209, 2006.
- [12] M. Mayo and E. Zhang. Improving Face Gender Classification By Adding Deliberately Misaligned Faces To The Training Data. *Image and Vision Computing New Zealand*, pp. 1-5, 2008.
- [13] A. Montillo and H. Ling. Age Regression from Faces Using Random Forests. *IEEE International Conference on Image Processing*, pp. 2465-2468, 2009
- [14] T. Ojala, M. Pietikainen and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition* 29, pp. 51–59, 1996.
- [15] K. Ricanek Jr. and T. Tesafaye. Morph: A longitudinal image database of normal adult age-progression. In *AFGR*, 2006.
- [16] Z. Yang, M. Li, and H. Ai. An experimental study on automatic face gender classification. *Proc. of the 18th International Conference on Pattern Recognition*, 2006.

# Распознавание растровых текстурных изображений на основе параметрических ациклических марковских моделей

Сергей Двоенко, Шанг Динь

Кафедра автоматики и телемеханики

Тульский Государственный Университет, Тула, Россия

dsd@tsu.tula.ru, dvietsang@gmail.com

## Аннотация

В задаче распознавания растровых текстурных изображений часто требуется выполнить сегментацию поля изображения на области однородных в некотором смысле текстур.

Предполагается, что массив данных представлен в виде двухкомпонентного марковского случайного поля скрытых классов текстур и наблюдаемых признаков. Требуется восстановить неизвестные маргинальные распределения вероятностей скрытых классов.

Взаимосвязи между точками раstra, образующие исходный циклический граф соседства элементов массива в виде решетки, аппроксимируются набором ациклических графов соседства.

Предлагается маргинальные распределения скрытых классов относительно исходной решетки заменить линейной комбинацией маргинальных распределений относительно заданных ациклических графов соседства.

Марковские матрицы переходов между классами также являются параметрами ациклических моделей и могут быть определены только одним диагональным элементом, который требуется определить.

**Ключевые слова:** графические модели, марковские поля, текстурные изображения, машинное обучение, распознавание, сегментация, интеллектуальный анализ данных.

## 1. ВВЕДЕНИЕ

В интеллектуальном анализе данных и машинном обучении в настоящее время интенсивно развивается направление, получившее название графических моделей, опирающееся на графы соседства элементов множества для построения эффективных алгоритмов распознавания и обработки, в том числе и изображений [9].

Пусть массив данных представлен в виде двухкомпонентного марковского случайного поля скрытых классов элементов массива и их наблюдаемых признаков. Исходные взаимосвязи между элементами массива в реальных данных аппроксимированы набором ациклических графов соседства элементов массива.

Требуется по предъявленным наблюдениям восстановить скрытую компоненту случайного поля.

Известно, что для графов общего вида с циклами задача распознавания марковских случайных полей обладает свойствами задачи класса  $NP$  [4, 6, 7]. Как правило, стандартным способом является решение данной задачи в терминах гиббсовских энергий, где минимизация гиббсовской энергии скрытых состояний марковского случайного поля доставляет максимум их неизвестному совместному апостериорному распределению на основе теоремы Хаммерсли-Клиффорда [6].

В данной статье развивается подход, основанный на численной оценке маргинальных распределений вероятностей скрытых классов, где неизвестные маргинальные распределения вероятностей скрытых классов относительно исходного графа соседства заменяются линейной комбинацией маргинальных распределений относительно заданных ациклических графов соседства [1–3].

Марковские матрицы переходов между скрытыми классами являются параметрами таких ациклических моделей и, как показано в данной статье, могут быть определены только одним диагональным элементом, который ранее [1–3] обычно задавался эвристически.

В данной статье построены алгоритмы поиска марковских параметров одновременно с поиском весов ациклических графов соседства в их линейной комбинации с целью эффективного улучшения качества распознавания.

Приведены результаты сравнительных экспериментов в задаче сегментации растровых текстурных изображений.

## 2. АЦИКЛИЧЕСКИЕ МОДЕЛИ МАРКОВСКОГО ПОЛЯ И АЛГОРИТМЫ РАСПОЗНАВАНИЯ

Пусть массив  $T$  взаимосвязанных объектов  $t \in T$  представлен как двухкомпонентное случайное поле  $(X, Y)$ .

Наблюдаемая компонента  $Y = (y_t, t \in T)$  принимает значения из множества  $y_t \in \Theta$ , определенного природой источника данных. Скрытая компонента  $X = (x_t, t \in T)$  подлежит восстановлению и в задаче распознавания принимает значения номеров классов объектов  $\Omega = \{1, \dots, m\}$ . Наблюдения условно независимы относительно реализации скрытого случайного поля классов объектов  $\psi_t(y_t | X) = \psi_t(y_t | x_t)$ .

На множестве элементов  $t \in T$  массива данных определено симметричное антирефлексивное бинарное отношение, представленное в виде неориентированного графа  $G$  без петель, ребра которого  $(s, t) \in T$  соединяют соседние элементы массива данных  $s \in T$  и  $t \in T$ . Граф  $G$  является деревом или цепью, т.е. не содержит циклов.

Скрытое поле  $X$  является марковским, если для всех  $t \in T$  выполнено  $q_t(x_t | X_{(t)}) = q_t(x_t | X_{(t)}^0)$ , где приняты обозначения  $X_{(t)} = (x_s, s \in T \setminus t)$ ,  $X_{(t)}^0 = (x_s, s \in T \setminus t, (s, t) \in G)$ .

Древовидный граф  $G$  разбивает окрестность нетерминального элемента  $x_t$  на две произвольные части  $X_{(t)}^0 = X_{(t)}^{-0} \cup X_{(t)}^{+0}$ . Вершина  $t^* \in T$  в качестве корня задает естественный нисходящий и восходящий порядок просмотра, определяя окрестность  $X_{(t)}^{-0} = x_r$  из одного элемента, предшествующего  $x_t$ , и окрестность  $X_{(t)}^{+0}$  непосредственных потомков  $x_t$ .

Можно доказать [1–3], что такое априорное поле является односторонним марковским  $q_t(x_t | X_{(t)}^0) = q_t(x_t | x_r)$ .

Можно также доказать [1–3], что апостериорное скрытое поле относительно того же графа  $G$  остается односторонним марковским  $p_t(x_t | X_{(t)}, Y) = p_t(x_t | x_r, Y_t^+)$ , где  $Y_t^+$  – поддерево с корнем в  $\mathbf{y}_t$ , включая его.

Для решений о поле  $X$  можно использовать, например, байесовское правило  $\hat{x}_t(Y) = \arg \max_{x_t \in \Omega} p_t(x_t | Y)$ ,  $t \in T$ , где

$p_t(x_t | Y)$  – локальные апостериорные маргинальные распределения вероятностей скрытых классов в элементах массива  $t \in T$ .

Восстановленная компонента  $X$  формирует связные области одинаковых номеров классов, образуя сегментацию поля наблюдений, например, растрового изображения на области с разными типами текстур.

**Базовый алгоритм.** Процедура восстановления скрытой компоненты называется базовым алгоритмом распознавания. Скрытое поле  $X$  распознается в общем случае за три прохода по дереву  $G$  [2, 3].

1. Задается начало обработки в корне  $t^*$  и априорное распределение  $q_{t^*}(x_{t^*})$ ,  $x_{t^*} \in \Omega$ . Нисходящим просмотром для всех  $t \in T$  вычисляются априорные распределения классов

$$q_s(x_s) = \sum_{x_t \in \Omega} q_s(x_s | x_t) q_t(x_t), \quad x_s \in \Omega, \quad s \in T_{(t)}^{+0}.$$

2. Восходящим просмотром от терминалов к корню вычисляются «фильтрационные» апостериорные маргинальные распределения классов

$$p_t(x_t | Y_t^+) \propto p_t(x_t | Y_{(t)}^+) p_t(x_t | \mathbf{y}_t), \quad x_t \in \Omega, \quad t \in T,$$

$$p_t(x_t | Y_{(t)}^+) \propto \prod_s \sum_{x_s \in \Omega} p_s(x_s | Y_s^+) \frac{q_s(x_s | x_t)}{q_s(x_s)}, \quad s \in T_{(t)}^{+0},$$

где  $Y_t^+$  – поддерево с корнем в  $\mathbf{y}_t$ , включая его. Апостериорные маргинальные распределения  $p_t(x_t | \mathbf{y}_t)$  получены на этапе независимого обучения. В терминальных вершинах принимается  $p_t(x_t | Y_t^+) = p_t(x_t | \mathbf{y}_t)$ .

3. На последнем восходящем шаге распределение в корне опирается на все наблюдения  $p_t(x_{t^*} | Y_{t^*}^+) = p_{t^*}(x_{t^*} | Y)$ ,  $x_{t^*} \in \Omega$ . Такое «интерполяционное» апостериорное маргинальное распределение значений корневой переменной позволяет принять решение о классе

$$\hat{x}_{t^*}(Y) = \arg \max_{x_{t^*} \in \Omega} p_{t^*}(x_{t^*} | Y).$$

4. Нисходящим просмотром от корня для всех  $t \in T$  вычисляются интерполяционные апостериорные маргинальные распределения

$$p_s(x_s | Y) \propto \sum_{x_t \in \Omega} p_s(x_s | x_t, Y) p_t(x_t | Y), \quad x_s \in \Omega, \quad s \in T_{(t)}^{+0},$$

$$p_s(x_s | x_t, Y) \propto p_s(x_s | Y_s^+) q_t(x_t) \frac{q_s(x_s | x_t)}{q_s(x_s)}$$

и принимаются решения о классах

$$\hat{x}_s(Y) = \arg \max_{x_s \in \Omega} p_s(x_s | Y).$$

**Итерационный алгоритм.** Для заданного набора ациклических графов соседства  $G_k$ ,  $k = 1, \dots, K$ , аппроксимирующих исходный граф соседства элементов реального массива данных, итерационный алгоритм распознавания построен на основе базового алгоритма и имеет следующий вид [3].

1. Для каждого ациклического графа  $G_k$  базовым алгоритмом формируются апостериорные распределения  $p_t^k(x_t | Y)$ ,  $k = 1, \dots, K$ ,  $t \in T$ , и на их основе формируются апостериорные распределения  $p_t(x_t | Y) = \sum_{k=1}^K w_k p_t^k(x_t | Y)$ ,  $t \in T$  для заданных весов  $w_k \geq 0$ ,  $\sum_{k=1}^K w_k = 1$ .

2. Только что найденные распределения  $p_t(x_t | Y)$  принимаются за исходные вместо распределений  $p_t(x_t | \mathbf{y}_t)$  и для каждого ациклического графа соседства в отдельности базовый алгоритм применяется вновь.

3. Итерационный алгоритм останавливается, когда распределения перестают изменяться, позволяя принять решения о классах  $\hat{x}_t(Y)$ ,  $t \in T$ .

**Схема Гаусса-Зайделя.** Веса графов соседства в линейной комбинации определяются по схеме Гаусса-Зайделя, где изменение веса графа  $G_k$  в диапазоне от 0 до 1 считаем аналогом покоординатного варьирования [3].

1. На начальном шаге распределение весов всех графов изменяется от  $1/(K-1), \dots, 0, \dots, 1/(K-1)$ , когда граф  $G_k$  полностью исключен, до  $0, \dots, 1, \dots, 0$ , когда есть только граф  $G_k$ . Шаг заканчивается после варьирования всех весов и выбора того графа, чей вес обеспечил минимальное число ошибок распознавания на обучающем множестве изображений.

2. На очередном шаге варьируется вес  $0 \leq p \leq 1$  графа  $G_k$ . Нормированный вес графа  $G_k$  в линейной комбинации имеет значение  $w_k = p$ . Остальные графы имеют к данному шагу постоянные веса  $l_i$ ,  $i = 1, \dots, K$ ,  $i \neq k$ , где их сумма также постоянна  $L = \sum_{i=1}^K l_i$ ,  $i \neq k$ . Их нормированные веса  $w_i = l_i(1-p)/L$  изменяются от  $w_i = l_i/L$  до  $w_i = 0$ .

Каждое пробное варьирование проверяется однократным комбинированием уже вычисленных маргинальных распределений  $p_t^k(x_t | Y)$ ,  $t \in T$ ,  $k = 1, \dots, K$  и распознаванием с подсчетом числа ошибок для решений  $\hat{x}_t$ ,  $t \in T$ .

Шаг заканчивается после варьирования всех весов и выбора нового веса того графа, для которого получено минимальное число ошибок.

### 3. ЧАСТНАЯ МОДЕЛЬ МАРКОВСКОГО ПОЛЯ

Марковское априорное случайное поле классов  $X$  определяет его одностороннее свойство в виде условных распределений вероятностей на множестве значений переменной  $x_t \in \Omega$  относительно любой переменной  $x_r \in X_{(t)}^0$  из марковской окрестности  $x_t$ .

Для каждой пары вершин  $(r, t) \in G$ , соединенных ребром, определена пара условных распределений  $q_r(x_t | x_r)$  и  $q_t(x_r | x_t)$ . Выбор некоторой вершины в качестве корня  $t^* \in T$  задает нисходящее и восходящее направления просмотра.

Односторонняя марковская модель поля  $X$  является однородной конечной марковской цепью с неизменными условными распределениями в восходящем и нисходящем направлениях, где  $q_r(x_r | x_t) = q(x_r | x_t)$ ,  $q_t(x_t | x_r) = \bar{q}(x_t | x_r)$ , и определена двумя матрицами  $Q(m \times m)$  и  $\bar{Q}(m \times m)$  ус-

ловных вероятностей переходов. Односторонняя марковская модель поля  $X$  является эргодической неразложимой марковской цепью и имеет финальное распределение  $p(x), x \in \Omega$  в корне  $t^*$ . Для неразложимой марковской цепи матрицы  $Q$  и  $\bar{Q}$  связаны через финальное распределение вероятностей  $\bar{q}(x_t | x_r) = q(x_r | x_t) p(x_t) / p(x_r)$ . В такой модели начальное априорное маргинальное распределение вероятностей  $q_t(x_t)$  можно задать в любой вершине  $t \in T$ , тогда соответствующие маргинальные распределения оказываются известными во всех остальных вершинах дерева  $G$ , в том числе и в его корне. Корень  $t^*$  естественно связать с началом обработки, а начальное распределение сделать корневым  $q_{t^*}(x_{t^*}), x_{t^*} \in \Omega$ .

Если предположить, что априорное распределение классов в корне  $t^*$  является равномерным финальным распределением  $q_{t^*}(x_{t^*}) = p(x_{t^*}) = 1/m$ , то маргинальные априорные распределения вероятностей скрытых классов во всех остальных элементах также равномерны. В этом случае  $\bar{q}(x_t | x_r) = q(x_r | x_t)$ .

Это, в частности, позволяет выполнить приведенный выше базовый алгоритм всего за два просмотра, исключив первый просмотр.

В итоге, остается только одна симметричная и дважды стохастичная матрица переходов  $Q$ . Предполагается, что такая матрица  $Q$  имеет одинаковые диагональные элементы и одинаковые недиагональные элементы. Тогда в такой частной модели матрица  $Q$  задается только одним значением ее диагонального элемента  $q$ , который оказывается единственным марковским параметром, требующим настройки.

Заметим, что теоретически для улучшения качества распознавания по сравнению с результатом независимого распознавания необходимо обеспечить  $q > 1/m$ . Ранее эвристически предполагалось, что для распознавания протяженных областей одного типа текстуры достаточно, например, обеспечить  $0.95 \leq q < 1$ , получив относительно редко переключающуюся марковскую цепь [1–3].

#### 4. АЛГОРИТМЫ ПОИСКА МАРКОВСКИХ ПАРАМЕТРОВ

Расширим схему Гаусса-Зайделя поиска весов графов для одновременного поиска оптимального значения диагонального элемента матрицы переходов. Заметим, что при однократном распознавании шаг варьирования диагонального элемента в диапазоне  $1/m \leq q < 1$  оказывается вырожденным, т.к. при любом наборе весов ациклических графов увеличение диагонального элемента при условии  $q < 1$  монотонно уменьшает число ошибок распознавания. Чтобы задача поиска не оказалась вырожденной, в новых алгоритмах поиска будем применять многократное распознавание при оценке числа ошибок.

**Алгоритм 1. Поиск единственного диагонального элемента и весов.** Все ациклические марковские модели, соответствующие графикам  $G_k, k=1, \dots, K$ , определяются одним общим диагональным элементом  $q$ . Сначала все веса одинаковы  $\mathbf{w}^* = \{w_i = 1/K, i=1, \dots, K\}$ .

1. Проварыруем  $q$  и найдем его значение при минимальном числе ошибок  $E$ :  $q^* = \arg \min_{1/m \leq q < 1} E(\mathbf{w}^*, q)$ .

2. Шаг варьирования весов по всем графикам. Варьируется вес  $w_k$  очередного графа  $G_k$  в диапазоне  $0 \leq w_k \leq 1$  при масштабировании весов остальных графов. Каждое пробное варьирование проверяется многократным распознаванием с подсчетом числа ошибок  $E(\mathbf{w}, q^*)$ . Для каждого графа  $G_k$  определим число ошибок и оптимальный вектор весов  $\mathbf{w}_k^* = (w_1, \dots, w_k^*, \dots, w_K)$ :

$$E_k^* = \min_{0 \leq w_k \leq 1} E(\mathbf{w}, q^*), \quad \mathbf{w}_k^* = \arg \min_{0 \leq w_k \leq 1} E(\mathbf{w}, q^*).$$

3. Среди всех наборов  $\mathbf{w}_k^*, k=1, \dots, K$  найдем набор  $\mathbf{w}^* = \mathbf{w}_{k^*}^*, k^* = \arg \min_{1 \leq k \leq K} E_k^*$ , обеспечивающий наименьшее число ошибок.

4. Повторим шаги 1–3 до тех пор, пока число ошибок распознавания не перестанет изменяться.

**Алгоритм 2. Первая схема последовательного поиска диагональных элементов и весов.** Каждому графу  $G_k, k=1, \dots, K$  соответствует ациклическая марковская модель со своим диагональным элементом  $q_k, k=1, \dots, K$ . Сначала все веса одинаковы  $\mathbf{w}^* = \{w_i = 1/K, i=1, \dots, K\}$ .

1. Проварыруем одновременно все диагональные элементы  $q_k = q, k=1, \dots, K$  и найдем значение  $q^*$  при минимальном числе ошибок  $E$ :  $\mathbf{q}^* = (q^*, \dots, q^*) = \arg \min_{1/m \leq q < 1} E(\mathbf{w}^*, \mathbf{q})$ .

2. Шаг варьирования весов по всем графикам. Варьируется вес  $w_k$  очередного графа  $G_k$  в диапазоне  $0 \leq w_k \leq 1$  при масштабировании весов остальных графов. Каждое пробное варьирование проверяется многократным распознаванием с подсчетом числа ошибок  $E(\mathbf{w}, \mathbf{q}^*)$ . Для каждого графа  $G_k$  определим число ошибок и вектор весов  $\mathbf{w}_k^* = (w_1, \dots, w_k^*, \dots, w_K)$ :

$$E_k^* = \min_{0 \leq w_k \leq 1} E(\mathbf{w}, \mathbf{q}^*), \quad \mathbf{w}_k^* = \arg \min_{0 \leq w_k \leq 1} E(\mathbf{w}, \mathbf{q}^*).$$

3. Среди всех наборов  $\mathbf{w}_k^*, k=1, \dots, K$  найдем набор  $\mathbf{w}^* = \mathbf{w}_{k^*}^*, k^* = \arg \min_{1 \leq k \leq K} E_k^*$ , обеспечивающий наименьшее число ошибок.

4. Шаг варьирования всех диагональных элементов. В диапазоне  $1/m \leq q_k < 1$  варьируется диагональный элемент  $q_k$ , соответствующий графу  $G_k$ . Остальным графикам соответствуют элементы  $q_i, i=1, \dots, K, i \neq k$ , найденные до варьирования  $q_k$ , которые остаются постоянными. Найдем значение  $q_k$  при минимальном числе ошибок:

$$q_k^* = \arg \min_{1/m \leq q_k < 1} E(\mathbf{w}^*, \mathbf{q}).$$

Новое значение  $q_k^*$  применяется при варьировании остальных диагональных элементов  $q_{k+1}, \dots, q_K$ . В итоге, получим оптимальный вектор  $\mathbf{q}^*$ .

5. Повторим шаги 2–4 до тех пор, пока число ошибок распознавания не перестанет изменяться.

### Алгоритм 3. Вторая схема параллельного поиска диагональных элементов и весов.

Каждому графу  $G_k, k=1, \dots, K$  соответствует отдельная ациклическая марковская модель со своим диагональным элементом  $q_k, k=1, \dots, K$ . Сначала все веса одинаковы  $\mathbf{w}^* = \{w_i = 1/K, i=1, \dots, K\}$ .

1. Проверяируем одновременно все диагональные элементы  $q_k = q, k=1, \dots, K$  и найдем значение  $q^*$  при минимальном числе ошибок  $E: \mathbf{q}^* = (q^*, \dots, q^*) = \arg \min_{1/m \leq q \leq 1} E(\mathbf{w}^*, \mathbf{q})$ .

2. Шаг варьирования весов и диагональных элементов по всем графикам. Варьируется вес  $w_k$  очередного графа  $G_k$  в диапазоне  $0 \leq w_k \leq 1$  при масштабировании весов остальных графов. Каждое пробное варьирование проверяется многократным распознаванием с подсчетом числа ошибок  $E(\mathbf{w}, \mathbf{q}^*)$ . Определим число ошибок и вектор весов  $\mathbf{w}_k^* = (w_1, \dots, w_k^*, \dots, w_K)$ :

$$E_k^* = \min_{0 \leq w_k \leq 1} E(\mathbf{w}, \mathbf{q}^*), \quad \mathbf{w}_k^* = \arg \min_{0 \leq w_k \leq 1} E(\mathbf{w}, \mathbf{q}^*).$$

Далее варьируется элемент  $q_k$ , соответствующий графу  $G_k$ , в диапазоне  $1/m \leq q_k < 1$ . Остальным графикам соответствуют элементы  $q_i, i=1, \dots, K, i \neq k$ , найденные до варьирования  $q_k$ , которые остаются постоянными. Найдем значение  $q_k$  при минимальном числе ошибок:

$$q_k^* = \arg \min_{1/m \leq q_k < 1} E(\mathbf{w}_k^*, \mathbf{q}^*).$$

Значение  $q_k^*$  определяет вектор  $\mathbf{q}_k^* = (q_1, \dots, q_k^*, \dots, q_K)$ .

3. Среди всех пар  $(\mathbf{w}_k^*, \mathbf{q}_k^*), k=1, \dots, K$  найдем пару

$(\mathbf{w}^*, \mathbf{q}^*) = (\mathbf{w}_{k^*}^*, \mathbf{q}_{k^*}^*), k^* = \arg \min_{1 \leq k \leq K} E(\mathbf{w}_k^*, \mathbf{q}_k^*)$ , обеспечивающую наименьшее число ошибок.

4. Повторим шаги 2–3 до тех пор, пока число ошибок распознавания не перестанет изменяться.

## 5. СРАВНЕНИЕ АЛГОРИТМОВ

Решалась задача сегментации 100 модельных растровых изображений (рис. 1) размером  $201 \times 201$  пикселей.

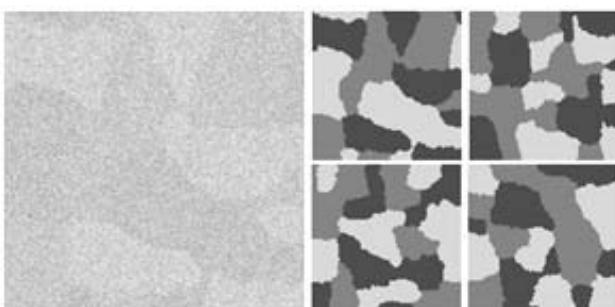


Рис 1: Пример текстуры и заданных сегментаций.

Текстуры трех классов были представлены реализациями трех нормально распределенных случайных величин с немного отличающимися средними в пространстве красной и зеленой компонент (на рис. 1 в градациях серого представлены: слева – пример первого изображения, справа – заданная сегментация текстур на первых четырех из 100 тесто-

вых изображений). Таким образом, множества точек растра трех классов очень сильно пересекались в пространстве цветовых RG-компонент.

Результаты независимого распознавания дают не менее 30% ошибок сегментирования поля растра.

На рис. 2 показаны ациклические графы соседства, которые удобно использовать для замены решеток растровых текстурных изображений.



Рис 2: Ациклические графы соседства.

Сравнивались четыре алгоритма поиска параметров ациклических марковских моделей (Г-3 – схема Гаусса-Зайделя поиска весов без поиска диагонального элемента, А1 – поиск единственного диагонального элемента и весов, А2 – первая схема поиска диагональных элементов и весов, А3 – вторая схема поиска диагональных элементов и весов) и алгоритм TRWS [5, 8, 10].

Изображения просматривались в случайном порядке, и средняя ошибка распознавания вычислялась каждый раз после добавления очередного изображения к просмотренному множеству. На рис. 3 показаны кумулятивные линии средних ошибок для одной из случайных последовательностей заданных изображений. На рис. 4 показаны линии отклонения доли средних ошибок разработанных алгоритмов относительно линии доли средних ошибок TRWS, показанной как горизонтальная координатная ось.

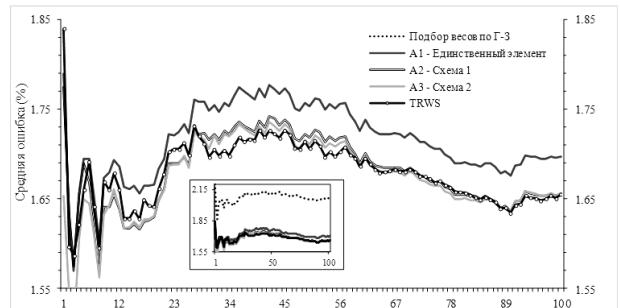


Рис 3: Среднее число ошибок (%).

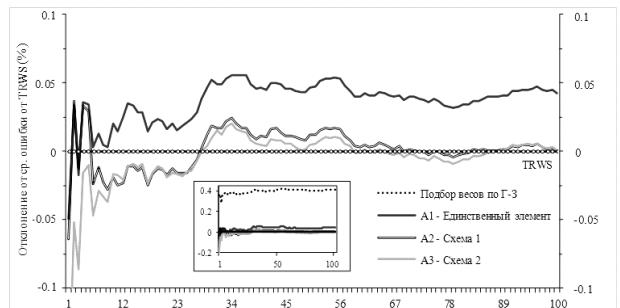


Рис 4: Отклонения линий ошибок.

Легко увидеть одинаковый характер кумулятивных линий средних ошибок для всех алгоритмов. Алгоритмы с поиском марковских параметров (A1–A3) резко улучшают качество распознавания по сравнению с алгоритмом по схеме

Гаусса-Зайделя ( $\Gamma$ -3) без такого поиска (показан на врезке). Например, алгоритм TRWS не является однозначно лучшим, т.к. на части изображений он уступает по качеству распознавания.

Очевидно, что характер поведения таких кумулятивных характеристик изменяется с увеличением размера выборки, становясь более плавным. На рис. 3 видно, что алгоритм A1 систематически проигрывает по средней ошибке остальным алгоритмам. Тем не менее, на начальном участке видно, что на каждом отдельном изображении он может выигрывать в качестве, давая меньшее число ошибок. Так и происходит: все сравниваемые алгоритмы (A1–A3, TRWS) достаточно часто уступали друг другу в качестве на различных отдельных изображениях.

В частности, при сравнении с алгоритмом TRWS оказалось, что алгоритм A1 был лучше только в 35 случаях из 100, сделав суммарно на 1721 ошибку больше; алгоритм A2 был лучше уже в 55 случаях из 100, сделав суммарно на 17 ошибок больше; алгоритм A3 был лучше в 50 случаях из 100, сделав суммарно на 43 ошибки больше.

В итоге, оказалось, что алгоритмы с поиском марковских параметров сравнимы по качеству между собой и с алгоритмом TRWS, который сегодня считается одним из эффективных алгоритмов обработки изображений. Например, при уровнях значимости  $\alpha = 0,001; 0,01; 0,05$  и, вообще, при  $\alpha < 0,3788$  (т.е. для всех значений правой границы  $Z_{kp} > 0,88$  двухсторонней критической области функции Лапласа, где  $|Z_{\text{набл}}| = 0,88$  соответствует максимальному из значений критерия, полученному для алгоритма A1) гипотеза о равенстве среднего уровня ошибок алгоритмов с поиском марковских параметров (A1 – A3) и TRWS не отвергается.

Визуально результаты сегментаций разными алгоритмами отличаются лишь мелкими деталями и выглядят практически как заданные сегментации, например, на рис. 1.

Ниже в таблице показаны параметры настройки алгоритмов на примере некоторых изображений из 100.

Параметры настройки алгоритмов						
Изоб- раже- ние	A1		A2		A3	
	Веса	q	Веса	q	Веса	q
1	0,04282	0,86	0,03947	0,90	0,02000	0,75
	0,20338		0,23684	0,89	0,09287	0,93
	0,24000		0,25000	0,91	0,36836	0,92
	0,25690		0,23684	0,90	0,25939	0,89
	0,25690		0,23684	0,89	0,25939	0,89
2	0,16894	0,89	0,17000	0,89	0,16000	0,89
	0,09034		0,17000	0,80	0,12643	0,89
	0,18863		0,17000	0,90	0,20205	0,87
	0,36209		0,32000	0,90	0,33540	0,88
	0,19000		0,17000	0,91	0,17612	0,90
3	0,06643	0,89	0,17408	0,70	0,09000	0,82
	0,19373		0,15105	0,88	0,19376	0,92
	0,30135		0,27079	0,91	0,19376	0,92
	0,23849		0,23000	0,89	0,32872	0,79
	0,20000		0,17408	0,91	0,19376	0,92
4	0,17953	0,90	0,21000	0,89	0,17225	0,90
	0,09995		0,08208	0,73	0,16000	0,88
	0,30963		0,23597	0,93	0,22966	0,91
	0,23089		0,23597	0,90	0,29854	0,90
	0,18000		0,23597	0,81	0,13955	0,89
52	0,16486	0,89	0,46511	0,37	0,15000	0,36
	0,23356		0,13000	0,61	0,21329	0,89
	0,23356		0,14111	0,90	0,21329	0,89
	0,25000		0,14111	0,92	0,21329	0,89
	0,11803		0,12267	0,82	0,21012	0,94

Легко увидеть, что марковские параметры настройки отличаются от предполагаемых по умолчанию эвристических значений. При поиске единственного диагонального элемента его значение оказывается достаточно большим, но меньше, чем обычно предполагалось эвристически.

При поиске диагональных элементов, соответствующих каждому из ациклических графов соседства, их значения, например, на первых четырех изображениях также достаточно велики, но снова меньше, чем предполагалось эвристически. Особенно это заметно, например, на изображении 52, для которого диагональный элемент, соответствующий первому графу соседства (ступенчатое дерево), оказался относительно небольшим.

## 6. ЗАКЛЮЧЕНИЕ

В данной статье для распознавания взаимосвязанных объектов, организованных в единый массив данных, применяется аппарат скрытых марковских моделей. Показано, что модель марковской цепи удаётся непосредственно применить для описания класса зависимых объектов с гораздо более широким разнообразием свойств, представленных марковскими полями, например, для растровых текстурных изображений.

Введение односторонней марковской модели для ациклических графов соседства взаимосвязанных элементов в массиве данных позволяет построить эффективный базовый неитерационный алгоритм распознавания.

Тем не менее, в реальных данных графы соседства обычно содержат циклы, как, например, решетки соседства на изображениях. Ациклическая аппроксимация исходных графов соседства неизбежно искажает характер взаимосвязей между элементами массива данных.

В данной статье предложено редуцированное множество взаимосвязей скомпенсировать расширением самого множества ациклических графов соседства. Разработаны соответствующие итерационные алгоритмы распознавания и приведены результаты экспериментов.

Следует отметить, что алгоритм TRWS решает глобальную задачу восстановления скрытого марковского поля в терминах гиббсовских энергий взаимодействия на кликах графа соседства общего вида. В случае растровых изображений это – только парные клики.

Алгоритмы A1 – A3 решают, вообще говоря, другую задачу восстановления скрытого марковского апостериорного поля в виде совокупности непосредственных численных оценок локальных апостериорных маргинальных распределений вероятностей скрытых классов. Тем не менее, показано, что, несмотря на ограниченность частной постановки такой задачи, также достигается высокое качество результатов распознавания.

## 7. БЛАГОДАРНОСТИ

Работа была выполнена при частичной поддержке грантов РФФИ №04-01-08038, 05-01-00679, 08-01-99003, 08-01-12023, 09-07-00394, 10-07-00489, гранта INTAS №04-77-7347.

## 8. ССЫЛКИ

- [1] Двоенко С.Д., Копылов А.В., Моттль В.В. Задача распознавания образов в массивах взаимосвязанных объектов. Постановка задачи и основные предположения// Автоматика и телемеханика. - 2004. - № 1. - С. 143-158.

- [2] Двоенко С.Д., Копылов А.В., Моттль В.В. Задача распознавания образов в массивах взаимосвязанных объектов. Алгоритм распознавания// Автоматика и телемеханика. - 2005. № 12. - С. 162-176.
- [3] Dvoenko S.D. Recognition of dependent objects based on acyclic Markov models// Pattern Recognition and Image Analysis. - 2012. - V. 22. № 1. Pp. 28-38. DOI: 10.1134/S1054661812010130.
- [4] Geman S., and Geman D., Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. IEEE Trans. on PAMI. **6**: 721–741 (1984).
- [5] Kolmogorov V., Convergent Tree-Reweighted Message Passing for Energy Minimization. IEEE Trans. on PAMI. **10**: 1568–1583 (2006).
- [6] Li S.Z. Markov Random Field Modeling in Image Analysis. London: Springer-Verlag, 2009.
- [7] Schlesinger M.I., and Flach B., Some Solvable Subclasses of Structural Recognition Problems. Proc. of Czech Pattern Recognition Workshop: 55–62 (2000).
- [8] Szeliski R., Zabih R., Scharstein D., Veksler O., Kolmogorov V., Agarwala A., Tappen M., and Rother C., A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors. IEEE Trans. on PAMI. **6**: 1068–1080 (2007).
- [9] Wainwright M.J., and Jordan M.I., Graphical Models, Exponential Families, and Variational Inference. Foundations and Trends in Machine Learning. **1**: 1–305 (2008).
- [10] [vision.middlebury.edu/MRF/code/MRF2.1.zip](http://vision.middlebury.edu/MRF/code/MRF2.1.zip)

### Об авторах

Сергей Д. Двоенко – профессор Тульского государственного университета. Его адрес: [dsd@tsu.tula.ru](mailto:dsd@tsu.tula.ru).  
 Шанг Динь Вьет – аспирант Тульского государственного университета. Его адрес: [dvietsang@gmail.com](mailto:dvietsang@gmail.com).

# Быстрый Алгоритм Обнаружения Пешеходов по Видеоданным

Алексей Казаков, Александр Бовырин

Факультет вычислительной математики и кибернетики

Нижегородский государственный университет им. Н.И. Лобачевского, Нижний Новгород, Россия

kazakovdz@yandex.ru, alexander.bovyrin@itseez.com

## Абстракт

В работе представлен алгоритм поиска пешеходов на изображении в реальном времени. Алгоритм основан на применении метода классификации AdaBoost совместно с HOG признаками, при этом размер окна поиска динамически изменяется в зависимости от положения линии горизонта на изображении.

**Ключевые слова:** обнаружение пешеходов, каскадный алгоритм, HOG признак, линейная интерполяция.

## 1. ВВЕДЕНИЕ

Задача поиска людей на изображении является важнейшей составляющей разнообразных прикладных систем компьютерного зрения. Подобные системы можно разделить на 3 группы: (1) системы видеонаблюдения, в которых камера зафиксирована и следит за определенным участком местности; (2) системы помощи водителю в управлении автомобилем, в котором установлена камера, направленная на дорогу по ходу движения; (3) системы поиска роботом человека, в которых камера встроена в робота и её положение относительно системы координат робота может изменяться. Далее для удобства будем использовать термин «пешеход» вместо более общего «человек», тем самым выделяя основной способ применения нашей системы, в качестве составляющей комплексов помощи водителю в управлении автомобилем.

Обнаружение пешеходов считается сложной и до сих пор не решённой задачей компьютерного зрения. Причиной этого является широкое разнообразие во внешнем виде пешеходов, зависящем от множества факторов, таких как: поза, одежда, освещение, фон. В качестве алгоритма машинного обучения в системе обнаружения пешеходов применялся AdaBoost на основе деревьев заданной высоты. В качестве тренировочных данных мы использовали базу «Daimler Chrysler Benchmark Dataset», состоящую из 15660 изображений пешеходов размером 48x96 пикселей и 6744 изображений не содержащих пешеходов. Тестовая база была представлена 27-ми минутным видео, которое было снято с камеры установленной на автомобиле и разбито на 21790 изображений 640x480. Каждому тестовому изображению соответствует запись в файле с разметкой, содержащей информацию о количестве, положениях и размерах пешеходов, присутствующих на изображении. Полноту видимые, вертикально стоящие пешеходы, на расстояниях от 12 до 37 метров, помечены в разметке как те, обнаружение которых обязательно.

## 2. ОБЗОР СИСТЕМ ОБНАРУЖЕНИЯ ПЕШЕХОДОВ

Приведём краткий обзор нескольких, на наш взгляд, наиболее значимых работ. С более полным обзором

работ по обнаружению пешеходов можно ознакомиться в статьях [2] и [4]. Кроме того, в работе [2] приводится сравнение качества и скорости работы, описанных систем на базе «Daimler Chrysler Benchmark Dataset».

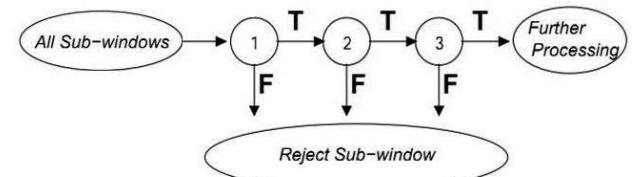
### 2.1 Каскадный алгоритм основанный на Haar wavelet признаках

Авторы работы [2] использовали для обнаружения пешеходов каскадный алгоритм Виолы - Джонса, основанный на Haar wavelet признаках. Подробное описание этого метода приведено в работе [8]. Кратко опишем суть каскадного алгоритма. В алгоритме поиск пешеходов осуществляется сканированием изображения скользящим окном (окном обнаружения), плотно заполненном Haar wavelet признаками. В алгоритме используются признаки различных типов, различных масштабов и положений (рис. 1). Внутри окна обнаружения таких признаков генерируется более 100,000.



**Рисунок 1:** Типы Haar wavelet признаков. Чёрная закраска соответствует положительному весу, белая - отрицательному.

Итоговый классификатор представляет собой цепочку уровней (рис. 2), каждый из которых тренируется с помощью алгоритма AdaBoost [3] на Haar-вейвлетах. В качестве слабого классификатора, используемого в AdaBoost, применялись деревья решений единичной высоты. Критерий качества каждого уровня – 50% ложных срабатываний при 99.5% обнаруженных пешеходов. Натренированный каскадный классификатор состоит из 15 уровней, использующих от 15 до 727 признаков.



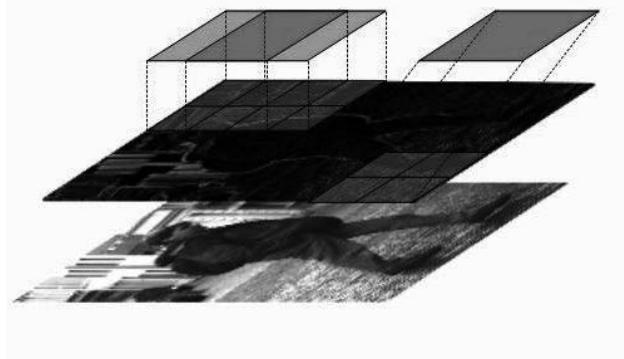
**Рисунок 2:** Схема каскадного классификатора. Каждое окно обнаружения проходит через цепочку уровней, каждый из которых отсекает большую долю негативных примеров. На следующие уровни поступают окна, не отсечённые на предыдущем. Критерии отсечения уровней задаются при тренировке.

### 2.2 HOG алгоритм

В 2004 году в работе [1] было предложено использовать HOG (Histograms of Oriented Gradients) дескрипторы в задаче обнаружения пешеходов. Идея этого подхода в том, что форма и вид объектов на изображении могут хорошо

описываться распределением относительных величин градиентов функции интенсивности, характеризующих направление границ объектов.

На практике такой подход реализуется следующим образом. Всё входное изображение разбивается на маленькие регионы - «ячейки», каждой из которых соответствует одномерная гистограмма направлений градиентов (ориентации рёбер). Каждый бин этой гистограммы соответствует определённому диапазону направлений и содержит суммарную (по пикселям ячейки) магнитуду градиентов соответствующего направления. Совокупность всех таких гистограмм образует общее представление о рёбрах на изображении. Для инвариантности к перепадам освещения производится нормировка значений в полученных гистограммах относительно некоторой области, содержащей рассматриваемую ячейку. Для этого соседние ячейки объединяются в группы - «блоки», а значения в гистограммах всех ячеек внутри блока нормируются на суммарное значение магнитуды градиентов по рассматриваемому блоку. Нормализованные блоки называются HOG дескрипторами, а множество таких блоков, локализованных в компактном регионе изображения, образуют «окно поиска». Итоговый вектор признаков образуется из конкатенации дескрипторов внутри окна поиска. В качестве алгоритма машинного обучения используется линейная машина опорных векторов (Linear SVM) [6]. Отметим, что натренированной моделью линейного SVM является плоскость, наилучшим образом разделяющая объекты от негативов в обучающей выборке.



**Рисунок 3:** Схема формирования HOG признака.

Градиенты, вычисленные на входном изображении (синего цвета) дискретизируются в гистограммы по ячейкам (жёлтого цвета). Ячейки группируются в перекрывающиеся между собой блоки (зелёного цвета). Конкатенация всех гистограмм всех блоков внутри окна образует итоговый вектор признаков.

В работе [1] рекомендуется использовать следующие параметры метода как оптимальные:  $[-1, 0, 1]$  - фильтр для вычисления градиентов на изображении; количество бинов в гистограммах – 9; размер ячейки в пикселях –  $8 \times 8$ ; количество ячеек в блоке –  $2 \times 2$ ; тип нормализации ячеек в блоке – квадратичная с усечением малых значений; шаг блоков в пикселях –  $8 \times 8$  (соответствует размеру ячейки).

Согласно обзору [2], описанный HOG алгоритм обладает самым высоким качеством обнаружения, значительно опережая каскадный алгоритм, основанный на Haar wavelet признаках

### 2.3 Каскадный алгоритм, основанный на HOG признаках

Авторы работы [8] совместили лучшие стороны, описанных выше методов (скорость работы каскадного алгоритма и качество обнаружения HOG алгоритма), в каскадном алгоритме, основанном на HOG признаках. В качестве признаков использовались блоки, состоящие из 4 ячеек, заполненных HOG признаками, дискретизированными по 9 бинам. Таким образом, каждый признак (блок) представляется 36-ти размерным вектором HOG признаков внутри блока. Для быстрого вычисления HOG признаков для каждого изображения строится интегральная гистограмма градиентов [5], позволяющая вычислять признак за несколько простейших операций. В рассматриваемом алгоритме окно обнаружения заполняется блоками трёх типов в зависимости от соотношения ширины блока к высоте (1:2, 1:1, 2:1). Размер и положение блоков сильно варьируются внутри окна обнаружения. Внутри окна обнаружения размером  $64 \times 128$  пикселей генерируется 5,031 блоков, из которых для тренировки случайным образом выбираются 5% (250) блоков.

Каждый уровень каскада тренировался (как и в каскадном алгоритме на основе Haar wavelet признаков) с помощью алгоритма AdaBoost. Однако в качестве слабого классификатора, используемого в AdaBoost, применялась линейная SVM.

Критерий качества каждого уровня – 70% ложных срабатываний при 99.75% обнаруженных пешеходов. Натренированный каскадный классификатор состоит из 30 уровней.

Применение данного подхода позволило существенно ускорить процесс обнаружения пешеходов при сохранении высокого качества обнаружения, близкого к результатам HOG алгоритма Далала.

### 3. МОДИФИКАЦИЯ КАСКАДНОГО АЛГОРИТМА

Следуя работе [8], мы попытались совместить каскадный алгоритм Виолы-Джонса с HOG признаками, внеся следующие изменения. В работе [8] в алгоритме AdaBoost в качестве слабого классификатора использовались линейные SVM. Слабый классификатор формировался по одному 36-размерному векторному признаку (блоку). Мы разбили векторные признаки на скалярные, а в качестве слабого классификатора использовали деревья решений заданной высоты. Таким образом, каждый уровень нашего каскада представляет собой ансамбль деревьев заданной высоты, каждый узел которых соответствует разбиению по одному из скалярных признаков по всем блокам. Для окна обнаружения  $48 \times 96$  пикселей генерировались блоки в диапазоне размеров от  $16 \times 16$  до  $48 \times 96$ . Шаг блоков внутри окна составлял 8 пикселей, а сами блоки согласно соотношению ширины и высоты трёх типов: вытянутые вверх (1:2), квадратные (1:1), вытянутые по ширине (2:1). Общее количество блоков – 894, а общее количество признаков – 32184 ( $894 \times 36$ ). Как и в работе [8] для ускорения вычисления HOG признаков для каждого обрабатываемого изображения строилась интегральная гистограмма [5], позволяющая вычислять признак за несколько простейших арифметических операций.

Тренировка алгоритма осуществлялась на данных, состоящих из 15660 пешеходов и 10000 фрагментов

изображений, не содержащих пешеходов. Критерий качества каждого уровня – 50% ложных срабатываний при 99.75% обнаруженных пешеходов. В алгоритме AdaBoost использовались деревья решений высоты 1(stump), 2, 3 и 4. Интегральные гистограммы содержат 9 бинов, отвечающих направлениям градиентов от 0 до 180 градусов. Натренированные каскадные классификаторы состоят из 18 уровней, использующих от 1 до 39 признаков. Реализация нашего каскадного алгоритма находится в библиотеке компьютерного зрения OpenCV [9].

#### 4. ИСПОЛЬЗОВАНИЕ ИНФОРМАЦИИ О ПОЛОЖЕНИИ ЛИНИИ ГОРИЗОНТА

В модификации каскадного алгоритма мы используем данные о положении линии горизонта на изображении, а так же считаем известными минимальный и максимальный размеры пешеходов. Мы используем допущение, что размеры пешеходов линейно растут от минимальных (на дальнем плане, вблизи линии горизонта) до максимальных (на ближнем плане) (рис. 4). Для обнаружения пешеходов различных размеров применяется масштабирование поискового окна. Пусть  $\min$ ,  $\max$  – размеры поисковых окон, соответствующих минимальному и максимальному размерам пешеходов. Поисковое окно большего размера получается умножением размеров текущего окна на величину  $\text{factor}$ , обозначающую отношение соседних по размеру окон. Количество  $n$  поисковых окон различных размеров в диапазоне  $[\min, \max]$  определяется величиной  $\lceil \log_{\text{factor}} \frac{\max}{\min} \rceil$ .

Участок входного изображения от линии горизонта ( $y_{\text{Hor}}$ ) до низа ( $H$ ) разбивается на  $n$  равных полос. Каждое  $i$  – е поисковое окно сканирует заданное количество полос ( $I$ ) изображения. Точнее,  $i$  – му окну достаётся вертикальный фрагмент изображения в диапазоне  $[y_{\text{Hor}} + i * h, y_{\text{Hor}} + (i + 1) * h]$ , где  $h = (H - y_{\text{Hor}}) / n$  высота одной полосы. Таким образом, каждая полоса сканируется с помощью серии из поисковых окон. Это необходимо для гибкости системы обнаружения к незначительным изменениям рельефа дороги.

Применение описанного подхода позволило значительно сократить количество поисковых окон. Совместно с использованием интегральных гистограмм для быстрого вычисления HOG признаков это позволило ускорить процесс обнаружения на порядок. Кроме того, за счёт уменьшения области поиска пешеходов, уменьшилось количество ложных срабатываний алгоритма, что привело к повышению качества работы системы в целом.

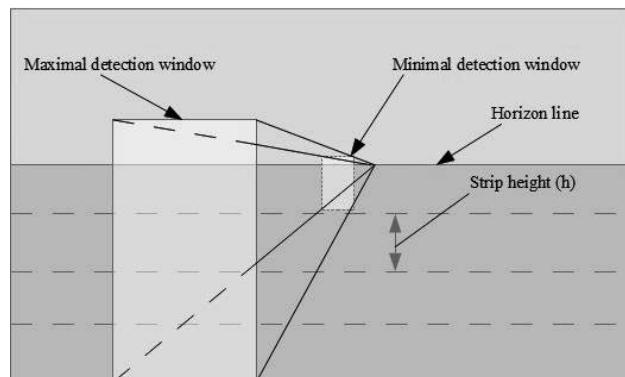


Рисунок 4. Схема поиска пешеходов использующая информацию о положении линии горизонта на

изображении. Для определения предполагаемого размера пешеходов применяется линейная интерполяция размеров пешеходов от минимального до максимального.

#### 5. ЭКСПЕРИМЕНТЫ

Для сравнения качества систем обнаружения пешеходов применяется подход, подробно описанный и использованный в работе [2]. В этом подходе к системам обнаружения предъявлены естественные требования, формирующие двухкритериальную задачу оптимизации. С одной стороны должно совершаться как можно меньшее количество ошибок второго рода, то есть, как можно большая доля объектов (detection rate - DR) должна быть обнаружена. С другой стороны должно минимизироваться количество ошибок первого рода (false positives per frame - FPPF), т.е. количество ложных срабатываний. Для визуализации качества работы системы используются ROC (Receive operating characteristic) – кривые, описывающие множество Парето, нашей двухкритериальной задачи оптимизации. Каждая точка ROC- кривой соответствует некоторой паре (FPPF, DR) при фиксированном пороге классификатора системы обнаружения. В качестве тестовых данных, как и в работе [2], использовались 21790 размеченных изображений из базы “Daimler Chrysler Benchmark Dataset”.

Все эксперименты проводились на CPU – IntelCore i7-2600 (3.4GHz), RAM - 8Gb, OS – Windows 7 Professional (64-bit).

На рисунке 4 приведено сравнение каскадных классификаторов, натренированных с использованием деревьев решений различных высот (1, 2, 3 и 4) в качестве слабых классификаторов. В таблице 1 указаны соответствующие им конфигурации параметров и скорость работы.

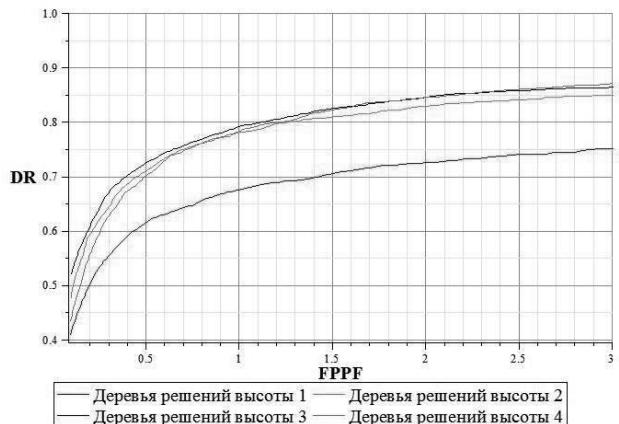


Рисунок 5: Сравнение качества работы каскадных алгоритмов на основе деревьев решений различных высот.

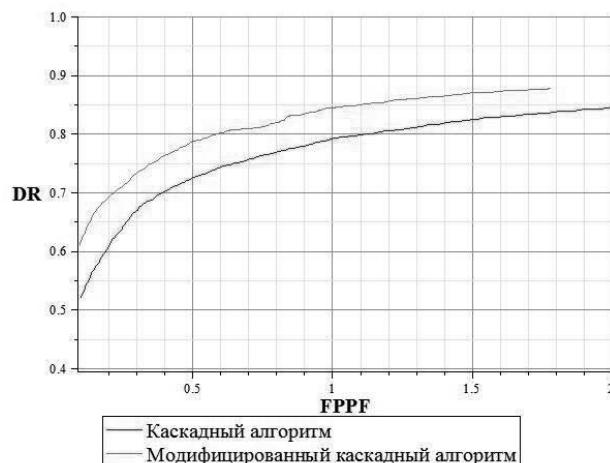
Таблица 1. Сравнение скорости работы каскадных алгоритмов на основе деревьев решений различных высот

	Деревья решений высоты 1	Деревья решений высоты 2	Деревья решений высоты 3	Деревья решений высоты 4
Шаг масштаба	1.05	1.05	1.05	1.05
Шаг по изображению	(4, 4)	(4, 4)	(4, 4)	(4, 4)

Время на обработку изображения (сек)	~0.140	~0.151	~0.156	~0.162
--------------------------------------	--------	--------	--------	--------

С увеличением высоты деревьев до 3 качество обнаружения улучшается. При дальнейшем увеличении высоты деревьев рост качества прекращается. Затрачиваемое время с ростом высоты деревьев увеличивается не значительно. По этим причинам в дальнейших экспериментах будем использовать каскадный классификатор, использующий деревья решений высоты 3.

На рисунке 5 и в таблице 2 представлено сравнение качества и времени каскадного алгоритма с его модификацией, использующей информацию о положении линии горизонта на изображениях. Линия горизонта на тестовых изображениях размечена вручную, однако допускается модификация, в которой линия горизонта задаётся одинаковой на всех тестовых данных, согласно направлению камеры.



**Рисунок 6:** Сравнение качества работы каскадного алгоритма с модифицированным каскадным алгоритмом, использующим информацию о положении линии горизонта.

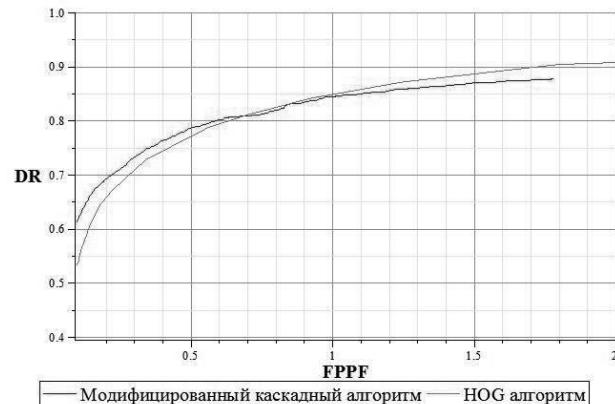
**Таблица 2.** Сравнение скорости работы каскадного алгоритма с модифицированным каскадным алгоритмом, использующим информацию о положении линии горизонта.

	Каскадный алгоритм	Модифицированный каскадный алгоритм
Шаг масштаба	1.05	1.05
Шаг по изображению (в пикселях)	(4, 4)	(4, 4)
Время на обработку изображения (сек)	~0.156	~0.034

Использование информации о положении линии горизонта не только позволило существенно ускорить процесс обнаружения, но и улучшило качество обнаружения.

На рисунке 7 и в таблице 3 приведено сравнение HOG алгоритма из работы [1] с модифицированным каскадным

алгоритмом, использующим информацию о положении линии горизонта на изображении. Для тестирования HOG алгоритма использовался программный код, реализованный авторами этого алгоритма – Н.Далалой и Б.Триггсом.



**Рисунок 7:** Сравнение качества работы модифицированного каскадного алгоритма с HOG алгоритмом в реализации Н.Далала, Б.Триггса.

**Таблица 3.** Сравнение скорости работы модифицированного каскадного алгоритма с HOG алгоритмом в реализации Н.Далала и Б.Триггса.

	Каскадный алгоритм	HOG алгоритм
Шаг масштаба	1.05	1.05
Шаг по изображению (в пикселях)	(4, 4)	(4, 4)
Время на обработку изображения (сек)	~0.034	~1

Каскадный алгоритм, использующий информацию о положении линии горизонта существенно опережает HOG алгоритм по скорости работы и сравним с ним по качеству обнаружения.

## 6. ЗАКЛЮЧЕНИЕ

Реализован каскадный алгоритм на основе HOG признаков, использующий в качестве слабого классификатора в алгоритме AdaBoost деревья решений заданной высоты. Применение информации о положении линии горизонта совместно с использованием интегральных гистограмм для ускоренного вычисления HOG признаков позволило увеличить скорость обнаружения до 30 кадров в секунду.

## 7. ССЫЛКИ

- [1] N. Dalal and B. Triggs. *Histograms of oriented gradients for human detection* // Conference on Computer Vision and Pattern Recognition (CVPR), 2005.
- [2] M. Enzweiler and Dariu M. Gavrila. *Monocular Pedestrian Detection: Survey and Experiments* // Pattern Analysis and Machine Intelligence, 2009.
- [3] Y. Freund, R.E. Schapire. *A Short Introduction to Boosting* // Journal of Japanese Society for Artificial Intelligence, 14(5): 771-780, September, 1999.

- [4] D. Geronimo, A.M. Lopez, A.D. Sappa, and T. Graf. *Survey of pedestrian detection for advanced driver assistance systems* // Pattern Analysis and Machine Intelligence, 2010.
- [5] F. Porikli. *Integral histogram: A fast way to extract histograms in Cartesian spaces* // Conference on Computer Vision and Pattern Recognition (CVPR), 2005.
- [6] V.N. Vapnik. *The Nature of Statistical Learning Theory* // Springer-Verlag, 1995.
- [7] P. Viola, M. Jones. *Rapid object detection using a boosted cascade of simple feature* // Conference on Computer Vision and Pattern Recognition (CVPR), 2001.
- [8] Qiang Zhu, Sai Avidan, Mei-Chen Yeh, and Kwang-Ting Cheng. *Fast Human Detection Using a Cascade of Histograms of Oriented Gradients* // Computer Vision and Pattern Recognition, 2006.
- [9] [www.opencv.org](http://www.opencv.org) – Open Source Computer vision library.

## Об авторах

Алексей Казаков - аспирант ВМК ННГУ имени Н.И. Лобачевского. Его адрес: kazakovdz@yandex.ru.

Александр Бовырин – директор по исследованиям и разработкам ООО «Аргус». Его адрес: alexander.bovyrin@itseez.com

# Расширенный набор характеристик Хаара

Глеб Южаков

Факультет аэрокосмических исследований

Московский физико-технический институт (ГУ), Долгопрудный, Московская область, Россия  
glebyu@gmail.com

## Аннотация

В настоящее время большую известность приобрел алгоритм быстрого детектирования объектов на ускоренном каскаде элементарных характеристических классификаторов, известный как метод Виолы и Джонса. В данной работе будет рассмотрена техника выбора наиболее информативных характеристик из расширенного набора характеристик Хаара (haar-like features), используемых в данном методе.

**Ключевые слова:** Характеристики Хаара, Метод Виолы и Джонса.

## 1. ВВЕДЕНИЕ

В 2001 году П. Виола и М. Джонс [5] представили многоэтапную процедуру классификации, которая позволила существенно сократить вычислительное время. При этом качество осталось сопоставимо со многими более медленными и сложными одноэтапными классификаторами, такими как машина опорных векторов (SVM), случайный лес (Random Forest) и нейронные сети (ANN). Этапами алгоритма Виолы и Джонса являются классификаторы бустинга [2] над решающими деревьями, использующими в качестве признаков характеристики Хаара.

Определим характеристику Хаара как функцию  $f$  от суммарной интенсивности  $I_A$  и  $I_B$  двух прямоугольных участков изображения  $A$  и  $B$ , таких что участок  $B$  вложен в участок  $A$ . Прямоугольная форма участков выбрана затем, чтобы можно было применить технику интегральных изображений [4, 5] для расчета суммарных интенсивностей в них.

В современных работах используется характеристика вида  $f(A, B) = \alpha I_A + \beta I_B$ , где  $\alpha$  и  $\beta$  – константы. Поэтому применение алгоритма Виолы и Джонса требует корректировки освещения:

$$I'(x, y) = \frac{I(x, y) - \mu}{c\sigma}, c > 0.$$

Здесь  $I(x, y)$  – интенсивность в точке  $(x, y)$ ,  $\sigma$  – оценочная дисперсия,  $\mu$  – оценочное среднее значение интенсивности по некоторой окрестности,  $c$  – положительная константа, которую обычно полагают равной двум [4].

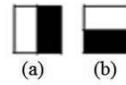
Вместо этого в данной работе для упрощения реализации алгоритма используется отношение суммарных интенсивностей:

$$f(A, B) = \frac{I_B}{I_A}.$$

Поскольку в данной статье производится сравнение нескольких подходов формирования характеристик, то эта модификация не может существенно повлиять на выводы.

Кроме того, в исходных работах [5] использовалось лишь несколько типов характеристик Хаара (рисунок 1).

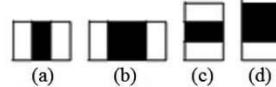
### 1. Характеристики угла



### 3. Характеристики центра



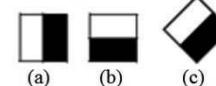
### 2. Характеристики линии



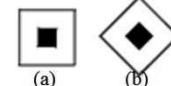
**Рис. 1:** Базовый набор характеристик Хаара.

В более поздних работах [4] набор характеристик Хаара был дополнен наклонными характеристиками (рисунок 2).

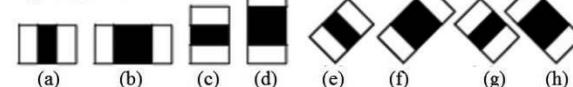
### 1. Характеристики угла



### 3. Характеристики центра



### 2. Характеристики линии



**Рис. 2:** Дополненный набор характеристик Хаара.

В данной работе предлагается отойти от идеи четкого задания нескольких типов характеристик Хаара и рассмотреть характеристики более общего вида, когда участок  $B$  имеет произвольное положение внутри участка  $A$  (рисунок 3).

## 2. ОПИСАНИЕ ХАРАКТЕРИСТИК

Рассмотрим скользящее окно, разбитое прямоугольной сеткой  $W \times H$ , внутри которого требуется проверить наличие изображения искомого объекта. Пусть имеется прямоугольник, заданный через координаты левого верхнего угла, ширину и высоту,  $r = (x, y, w, h)$ , где  $0 \leq x \leq W - w, 0 \leq y \leq H - h, 0 < w \leq W, 0 < h \leq H, I(r)$  – сумма интенсивности пикселов в прямоугольнике  $r$  исследуемого изображения. Тогда множество характеристик Хаара можно представить в виде

$$F = \left\{ \frac{I(r_i)}{I(r_j)}, r_i \in r_j \right\}.$$

Количество таких характеристик определяется по формуле

$$n = \sum_{w_A=1}^W \sum_{h_A=1}^H (W - w_A + 1)(H - h_A + 1)n(w_A, h_A),$$

$$\sum_{w_B=1}^{w_A} \sum_{h_B=1}^{h_A} (w_A - w_B + 1)(h_A - h_B + 1).$$

Отсюда

$$n = \frac{1}{576} W(W + 1)(W^2 + 5W + 6)H(H + 1)(H^2 + 5H + 6).$$

Но их слишком много, так как при размерах сетки  $24 \times 24$  число характеристик достигает значения  $n = 308002500$ . Поэтому предлагается специальная техника выбора наиболее информативных характеристик Хаара за приемлемое вычислительное время.

### 3. ВЫЧИСЛЕНИЕ ХАРАКТЕРИСТИК

Под наиболее информативными характеристиками Хаара понимаются те, на которых достигается максимум критерия, характеризующего качество разделения объектов на положительные и отрицательные.

Так как характеристик слишком много, то техника выбора наиболее информативных характеристик состоит из трех ключевых моментов, позволяющих сократить временные и вычислительные затраты:

- 1) уточнение размера характеристик;
- 2) взятие случайного подмножества исследуемых характеристик;
- 3) аппроксимацией обучающей выборки обучающим подмножеством.

Рассмотрим их более подробно.

#### 3.1 Уточнение размера характеристик

Пусть имеется две сетки, которыми разбиваются эталонные примеры:  $W \times H$  и  $2W \times 2H$ . Тогда под уточнением понимается следующая процедура:

- 1) составление множества из  $k$  наиболее информативных (грубых) характеристик для сетки  $W \times H$ ;
- 2) построение характеристик на сетке  $2W \times 2H$ , приближаемых полученными грубыми характеристиками;
- 3) выбор из этих характеристик наиболее информативных (точных) характеристик.

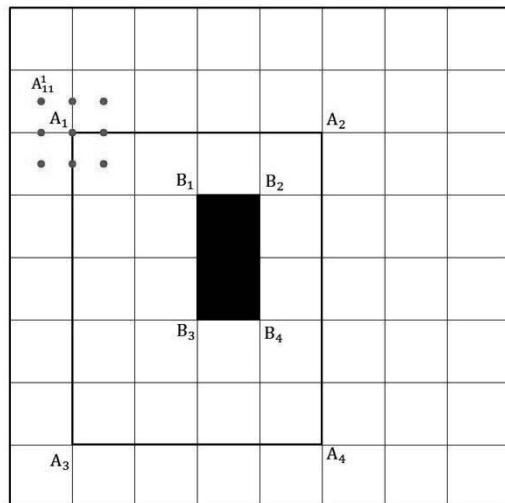


Рис. 3: Уточнение размера участков А и В.

На рисунке 3 прямоугольные участки  $A$  ( $A_1A_2A_3A_4$ ) и  $B$  ( $B_1B_2B_3B_4$ ) образуют грубую характеристику Хаара. Для уточнения рассматриваются все возможные смещения вершин составляющих грубую характеристику участков на половину размера ячейки сетки  $W \times H$ . В результате образуются участки вида  $X_{ij}^1X_{il}^2X_{kj}^3X_{kl}^4$ , где  $i, j, k, l = \{0, 1, 2\}$ ,  $X = \{A, B\}$ . Таким образом, каждая грубая характеристика образует до  $9^4 = 6561$  уточняющих ее характеристик.

Поскольку требуется, чтобы участок  $B$  был вложен в  $A$ , уточняющих характеристик будет меньше. В случае грубого размера сетки  $12 \times 12$  в среднем их число будет около 3000. При этом число различных (грубых) характеристик Хаара всего  $n = 1863225$ .

#### 3.2 Взятие случайного подмножества исследуемых характеристик

Пусть каждая из возможных характеристик берется в рассмотрение с некоторой вероятностью  $p$ . В таком случае ожидаемое количество характеристик становится меньше:  $n_p \approx rp$ . Поскольку выбор характеристик производится при каждом обучении элементарного классификатора, данная процедура в среднем не ухудшает качество классификации, но уменьшает время обучения классификатора.

#### 3.3 Аппроксимация обучающей выборки обучающим подмножеством

Еще одним методом ускорения выбора наиболее информативных характеристик является аппроксимация всей обучающей выборки некоторым его подмножеством. Так как у каждого обучающего примера имеется свой вес, то целесообразно в обучающее подмножество выбирать примеры с максимальным весом. Размер обучающего подмножества является фиксированным. В случае, когда имеется несколько примеров, претендующих попасть в обучающее подмножество и имеющих равные веса, решение об их принятии принимается случайно.

Пусть имеется положительная  $X_+$  и отрицательная  $X_-$  обучающие выборки. Опишем процедуру выбора наиболее информативных характеристик с учетом аппроксимации:

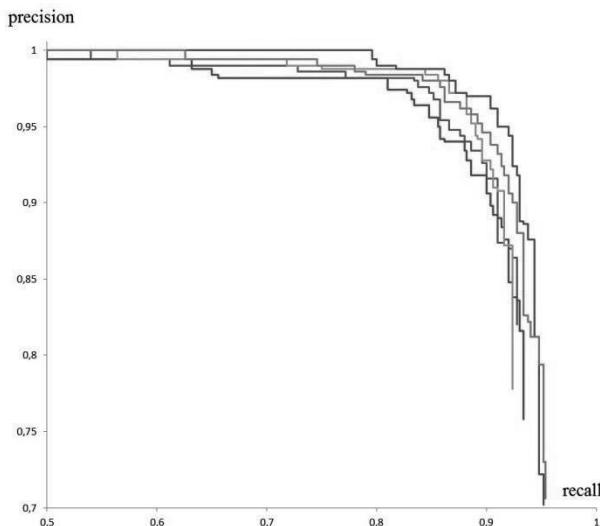
- 1) составляются обучающие подмножества  $T_+ \subset X_+$  и  $T_- \subset X_-$  определенных размеров  $n_+ = |T_+|$  и  $n_- = |T_-|$ ;
- 2) определяется  $m$  наиболее информативных характеристик Хаара на примерах из обучающих подмножеств;
- 3) выбирается несколько наиболее информативных характеристик Хаара из этих  $m$ , но уже на примерах из всего обучающего множества  $X_+$  и  $X_-$ .

Теперь остается на выбранных характеристиках Хаара построить очередной элементарный классификатор.

### 4. ЭКСПЕРИМЕНТАЛЬНЫЕ РЕЗУЛЬТАТЫ

Рассмотрим алгоритм быстрого детектирования объектов на ускоренном каскаде классификаторов бустинга над решающими деревьями, использующими характеристики Хаара. На рисунке 4 приведен сравнительный анализ качества (график точности от полноты precision – recall) работы данного алгоритма детектирования в зависимости от используемого метода выбора наиболее информативных характеристик Хаара на сетке размером  $24 \times 24$  (в случае применения уточняющего метода бралась сетка размером  $12 \times 12$ ).

В работе использовались открытые выборки LFW [3] и VOC [1]. В качестве источника положительных обучающих и тестовых примеров была взята открытая выборка LFW, содержащая лица. В качестве источника отрицательных обучающих и тестовых примеров была взята выборка VOC, не содержащая лица. Положительная обучающая выборка состояла из 1000 примеров, тестовая из 300. Обе отрицательные выборки состояли из 500 несовпадающих изображений, на которых производился поиск искомого объекта и выбросов соответственно.



**Рис. 4:** Сравнительные результаты качества работы алгоритма детектирования в зависимости от метода выбора наиболее информативных характеристик Хаара. Информация о соответствии цвета кривой и метода содержится в таблице 1.

**Табл. 1:** Сравнительные результаты алгоритмов детектирования лиц в зависимости от метода выбора наиболее информативных характеристик Хаара (названия полей таблицы уточнены ниже).

Метод	Площадь под графиком	Время обработки, мс	Время обучения, часов
Синий	0.918	25.0	6.2
Голубой	0.942	<b>22.8</b>	4.5
Красный	0.917	27.6	4.0
Зеленый	0.921	28.4	<b>3.3</b>
Фиолетовый	<b>0.945</b>	23.7	6.3

В первом столбце таблицы содержится информация о соответствии метода и цвета, соответствующего ему графику точности от полноты на рисунке 4. А именно, синему цвету соответствует поиск наиболее информативных характеристик Хаара, использующий расширенный набор с уточнением и случайным взятием (10%), голубому – расширенный набор с уточнением, красному – базовый набор, зеленому – дополненный наклонными характеристиками набор, фиолетовому – дополненный и расширенный набор характеристик Хаара с уточнением и случайным взятием.

В втором столбце дана площадь под графиком точности от полноты (precision – recall) соответствующего метода.

В третьем столбце приведено среднее время обработки одного изображения размером  $500 \times 500$ , процессор Intel Core i3 – 2120, 3.3 GHz. При этом минимальный размер искомого объекта  $96 \times 96$ .

В четвертом столбце приведено время в часах, которое требуется, чтобы алгоритм обучился.

Из таблицы 1 видно, что точность классификатора, использующего метод выбора наиболее информативных

характеристик из расширенного набора характеристик Хаара, выше, чем у классификатора, использующего при обучении дополненный и базовый набор характеристик Хаара, 0.942 в сравнении с 0.921. При этом он тратит меньше времени на обработку одного изображения, 22.8 мс в сравнении с 28.4 мс. Но зато ему требуется больше времени на обучение, 4.5 часа против 3.3.

Наилучшее качество (0.945) показал классификатор, использующий все три описанных метода выбора наиболее информативных характеристик, хотя он требует и наибольшее время на обучение (6.3 часа).

## 5. ЗАКЛЮЧЕНИЕ

Таким образом, описанная выше техника является эффективным методом выбора наиболее информативных характеристик Хаара. В сравнительном эксперименте при детектировании лиц у метода Виолы и Джонса, использующего данную технику, наблюдается более высокое качество детектирования и меньшее время, требующееся на обработку одного кадра. Единственным недостатком, но не очень существенным, является увеличение времени обучения классификатора.

## 6. ССЫЛКИ

- [1] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman, The PASCAL Visual Object Classes (VOC) Challenge, International Journal of Computer Vision, 88(2), 303-338, 2010.
- [2] Y. Freund, R. E. Schapire, A Short Introduction to Boosting, Shannon Laboratory, USA, 1999., pp. 771-780
- [3] G. B. Huang, M. Ramesh, T. Berg and E. Learned-Miller, Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. University of Massachusetts, Amherst, Technical Report 07-49, 2007.
- [4] R. Lienhart, A. Kuranov, V. Pisarevsky, Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection With an Extended Set of Haar-like Features, – Intel Technical Report, 2002.
- [5] P. Viola, M. Jones, Rapid Object Detection using a Boosted Cascade of Simple Features, – conference on computer vision and pattern recognition, 2001.

## Об авторах

Глеб Южаков – аспирант ФАКИ МФТИ, научный сотрудник ФГУП ЦНИИХМ. Его адрес: [glebyu@gmail.com](mailto:glebyu@gmail.com).

# Частичное Обучение с Учителем на Небольших Исходных Выборках

Сергей Завалишин, Илья Сафонов

Национальный Исследовательский Ядерный Университет «МИФИ», Москва, Россия

{Etti.Nekov, ilia.safonov} at gmail dot com

## Аннотация

В данной работе предлагается алгоритм частичного обучения с учителем, эффективно работающий в условиях использования экстремально малой исходной обучающей выборки, состоящей из небольшого количества заранее классифицированных объектов, и большого, но заранее неизвестного количества неклассифицированных объектов. Рассматривается проблема пошагового обучения, когда число объектов в обучающей выборке не является постоянным и увеличивается по ходу обучения. Тестирование метода производится с помощью изображений, скачанных с сервиса Flickr, которые не отбирались вручную, что ведёт к появлению ошибок в исходной обучающей выборке. Для устранения этих ошибок и повышения эффективности обучения предлагается использовать ряд специальных методик: перемаркировку голосованием, фиксацию результатов и фильтрацию данных.

**Ключевые слова:** частичное обучение с учителем, самообучение, совместное обучение, активное обучение.

## 1. ВВЕДЕНИЕ

В настоящее время персонализированный поиск всё глубже входит в нашу жизнь. Множество веб-сервисов запускают механизмы анализа пользовательских предпочтений для предоставления своим пользователям наиболее интересных материалов, в том числе и рекламного характера, что обуславливает необходимость внедрения систем автоматического обучения на основе собранных данных. В случае, когда объём этих данных достаточно велик, обучение не представляет каких-либо проблем, однако в реальности, как правило, количество доступной информации о каждом конкретном пользователе ограничено, особенно это касается новых пользователей, которые только недавно начали посещать какой-либо веб-сервис. В подобных случаях [15] эффективным является использование методов частичного обучения с учителем, когда для успешного обучения требуется наличие лишь небольшого числа заранее разбитых на классы объектов (соответствующих предпочтениям пользователя) и большое число объектов, относящихся к неизвестным классам. Очевидно, что база данных любого веб-сервиса, полностью удовлетворяет данным требованиям, что заметно упрощает процесс интеграции системы обучения подобного типа. Тем не менее, несмотря на кажущуюся простоту данного подхода, имеется и ряд проблем, которые требуют решения.

Современные алгоритмы частичного обучения с учителем рассчитаны на использование обучающих выборок заранее известного размера, однако базы данных веб-сервисов, как правило, содержат миллионы объектов, что значительно увеличивает время обучения. Можно было бы ограничиться некоторым заранее определённым числом объектов в обучающей выборке, однако исследования показывают, что с ростом числа объектов улучшается и качество работы обученного классификатора. В данной работе предлагается

пошаговый метод частичного обучения с учителем, при котором объём данных растёт постепенно, а процесс обучения заканчивается в тот момент, когда дальнейшее увеличение объёма выборки перестаёт иметь смысла.

Вторая проблема заключается в том, что исходная выборка, состоящая из промаркированных объектов и отражающая предпочтения пользователя, может иметь ошибки – промаркированные объекты, которые не относятся ни к одному из имеющихся в системе классов. Известно, что ошибки подобного рода негативно влияют на качество обучения, поэтому устранение их влияния является одной из важных задач.

Наконец, третья проблема касается объёмов исходных (промаркированных вручную человеком) обучающих выборок. Как правило, в исследовательских работах рассматривается ситуация, когда исходная выборка состоит из нескольких сотен объектов, однако в реальности она может содержать всего лишь несколько десятков объектов, что значительно затрудняет использование алгоритмов частичного обучения с учителем.

Для решения этих проблем нами предлагается ряд специальных методов: перемаркировка голосованием, фиксация результатов и фильтрация данных.

## 2. ОБЗОР СУЩЕСТВУЮЩИХ ПУБЛИКАЦИЙ

Существует множество алгоритмов частичного обучения с учителем [15]. В рамках данной работы было принято решение сфокусироваться на наиболее популярных из них: самообучении (self-training) [9] и совместном обучении (co-training) [1].

Основная идея первого из них состоит в следующем: первоначально классификатор обучается на некоторой исходной выборке, состоящей из небольшого числа промаркированных (классифицированных) человеком объектов. Затем обученный классификатор уже без помощи человека маркирует другой – больший – набор объектов, классы которых неизвестны. Два полученных набора объектов, каждый из которых теперь стал промаркированным, объединяются вместе, формируя тем самым новую обучающую выборку, на которой классификатор обучается повторно. После этого классификатор заново маркирует все объекты в объединённой обучающей выборке и снова переобучается на ней. Процесс проходит до тех пор, пока маркировка отдельных объектов не перестанет меняться.

Метод совместного обучения работает, в целом, аналогичным образом, однако вместо одного классификатора на одном и том же наборе данных параллельно обучаются два, а перемаркировка отдельных объектов осуществляется либо за счёт голосования между классификаторами, цель которого – выбор наиболее верной метки, либо через использование двух независимых обучающих выборок с разными метками, когда один классификатор маркирует обучающую выборку другого.

Одним из хорошо известных алгоритмов частичного обучения с учителем является алгоритм Яровского [5]. Существует ряд модификаций данного алгоритма, позволяющих увеличить качество обучения.

Ряд работ [4, 7] указывает на то, что увеличение количества немаркированных объектов в обучающей выборке позволяет достичь лучшего качества работы классификатора при использовании алгоритмов частичного обучения с учителем, однако данный подход эффективен только в том случае, если модель данных, полученная за счёт обучения, оказывается близка к реальным данным, чего бывает трудно достичь при использовании малой исходной выборки. Для решения данной проблемы рекомендуется использовать фильтрацию шумов.

Так же для увеличения результативности обучения часто используется метод активного обучения [12, 13], который производит перемаркировку объектов не в полностью автоматическом режиме, а с участием человека: в процессе перемаркировки для некоторых объектов, в которых классификатор не уверен, требуется ручной ввод меток классов. Данный подход хорошо согласуется с идеей пошагового обучения, поэтому он, наряду с другими методами, используется для улучшения качества классификатора.

### 3. СРЕДА ОБУЧЕНИЯ

#### 3.1. Модификация LIBSVM

Существует множество подходов к машинному обучению, таких как нейронные сети, наивный Байес и прочих, однако в рамках данной работы используется машина опорных векторов (далее SVM – Support Vector Machine), так как она одновременно обеспечивает достаточное быстродействие и точность [6] классификации. В данной работе SVM реализована с помощью открытой библиотеки LIBSVM [3].

Основная задача машины опорных векторов – это построение оптимальной разделяющей гиперплоскости с наибольшим зазором (margin) между классами в пространстве векторов с размерностью равной количеству признаков.

Для нахождения максимальной разделяющей гиперплоскости производится решение задачи оптимизации следующего вида:

$$\begin{cases} \mathcal{L}(w, w_0; \lambda) = \frac{1}{2} \langle w, w \rangle - \sum_{i=1}^l \lambda_i (y_i (\langle w, x_i \rangle - w_0) - 1) \rightarrow \min_{w, w_0} \max_{\lambda}; \\ \lambda_i \geq 0, i = 1, \dots, l; \\ \lambda_i = 0, \text{либо } \langle w, x_i \rangle - w_0 = y_i, i = 1, \dots, l \end{cases} \quad (1)$$

Где  $x_i = (x_i^1, \dots, x_i^n)$  – признаковое описание объекта  $x_i$ ;  $w = (w^1, \dots, w^n) \in \mathbb{R}^n$  и  $w_0 \in \mathbb{R}$  – параметры алгоритма;  $y_i = y^*(x_i)$  – целевая зависимость  $y^*: X \rightarrow Y$ , где  $X$  – пространство объектов, а  $Y$  – множество ответов;  $\lambda = (\lambda_1, \dots, \lambda_l)$  – вектор двойственных переменных.

Очевидно, что разделяющая гиперплоскость может быть найдена далеко не всегда, поэтому в таких случаях применяется так называемое ядерное преобразование, позволяющее строить нелинейные разделители. Основная идея состоит в замене всех скалярных произведений, использующихся при нахождении разделяющей гиперплоскости, функциями ядра – то есть скалярными произведениями в пространстве большей размерности, где уже может существовать оптимальная разделяющая гиперплоскость.

Нами используется машина опорных векторов, которая в качестве параметра использует значение веса  $C$ , влияющего на решение задачи оптимизации:

$$\min_{w, b, \xi} \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i \quad (2)$$

$$y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, \quad (3)$$

$$\xi_i \geq 0,$$

где  $\phi$  – функция, переводящая вектор  $x_i$  в пространство с большей размерностью (при этом  $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$  называется функцией ядра);  $w = (w^1, \dots, w^n) \in \mathbb{R}^n$  и  $w_0 \in \mathbb{R}$  – параметры алгоритма;  $y_i = y^*(x_i)$  – целевая зависимость  $y^*: X \rightarrow Y$ , где  $X$  – пространство объектов, а  $Y$  – множество ответов;  $\xi = (\xi_1, \dots, \xi_l)$  – вектор двойственных переменных.

Параметр  $C > 0$  определяет уровень ошибки и используется для отсечения заведомо некорректно классифицированных объектов. В рамках данной работы используется значение  $C = 1$ , так как отсечение производится другими методами.

Современные исследования показывают, что использование ядра  $\chi^2$  позволяет увеличить качество классификатора:

$$k(x, y) = 1 - \sum_{i=1}^n \frac{(x_i - y_i)^2}{\frac{1}{2}(x_i + y_i)} \quad (5)$$

При этом экспоненциальная форма данного ядра  $\chi^2_{\exp}$  оказывается ещё лучше:

$$k(x, y) = \exp \left( -\gamma \sum_{i=1}^n \frac{(x_i - y_i)^2}{\frac{1}{2}(x_i + y_i)} \right) \quad (6)$$

По умолчанию LIBSVM поддерживает несколько различных ядер, но поддержки ядра  $\chi^2$  в этой библиотеке нет. Существуют различные решения от сторонних разработчиков, добавляющие  $\chi^2$ , однако они все нацелены на использование в среде MATLAB, что требует наличия соответствующей программной среды или её компонентов. Для решения данной проблемы в библиотеку LIBSVM была включена поддержка ядер  $\chi^2$  и  $\chi^2_{\exp}$ .

Чтобы определить наиболее подходящее решаемой задаче ядро был проведён ряд тестов на выборке следующего объёма – 700 изображений для обучения и 100 для проверки, соответствующих двум классам сцен: внутри и вне помещений, скачанных с сервиса Flickr. При таком соотношении разница в результатах, показываемых различными ядрами, оказалась наиболее заметна. В качестве численных метрик качества классификации использовались следующие метрики:

$$Precision = \frac{TP}{TP+FP} \cdot 100%; \quad Recall = \frac{TP}{TP+FN} \cdot 100%;$$

$$Accuracy = \frac{TP+TN}{N} \cdot 100%; \quad (7)$$

Где  $N$  – общее число изображений,  $TP$  – число верных срабатываний, а  $FP$  и  $FN$  – число ложных срабатываний и пропусков, соответственно. При этом здесь и далее целевым классом являлся класс, соответствующий изображениям, содержащим сцены внутри помещений.

Как и ожидалось, ядро  $\chi^2_{\exp}$  показало наилучшие результаты (таблица 1).

Таблица 1: Сравнение различных ядер

Ядро	Accuracy, %	Precision, %	Recall, %
RBF	75	75	76
$\chi^2$	79	76	84
$\chi^2_{\exp}$	82	80	86

### 3.2. Распознавание сцен внутри и вне помещений

Для распознавания сцен внутри и вне помещений мы использовали метод [8], который основывается на применении низкоуровневых цветовых и текстурных признаков изображений.

Обычно в качестве гистограммных признаков используются значения гистограммы различных каналов изображения с уменьшенным числом оттенков (до 16 градаций на канал). Наиболее очевидным вариантом является использование каналов RGB, однако такой подход не всегда эффективен: в RGB яркость не отделена от цветового тона и насыщенности, что приводит к дублированию информации, так как для большого количества фотографий в этом цветовом пространстве вид каналов отличается незначительно.

Мы использовали цветовое пространство LST [11]. Преобразование пикселя из пространства RGB в LST выглядит следующим образом:

$$L = \frac{k}{\sqrt{3}}(R + G + B), \quad S = \frac{k}{\sqrt{2}}(R - B), \\ T = \frac{k}{\sqrt{6}}(R - 2G + B) \quad (8)$$

Где  $L$  – канал освещённости,  $S$  и  $T$  отражают хроматическую информацию, а  $k = 255/\max(R, G, B)$ . Особенность  $S$  и  $T$  компонент состоит в том, что их значения не меняются с изменением интенсивности источника света. При этом канал  $S$  определяет вариации цветовой температуры от дневного света до искусственного, создаваемого лампой накаливания, что очень удобно для распознавания сцен внутри и вне помещений.

Для определения текстурных признаков используется двухуровневое wavelet-преобразование канала  $L$  при помощи фильтров Добеши, определяющихся следующими коэффициентами:

$$h \leftrightarrow [-0,129, 0,224, 0,837, 0,483], \\ g \leftrightarrow [-0,483, 0,837, -0,224, -0,129]$$

Где  $h$  – высокочастотный фильтр, а  $g$  – низкочастотный.

Текстурные признаки  $e_1, e_2, \dots, e_8$  определяются по формуле (9):

$$e_k = \frac{1}{M_k N_k} \sum_{i=1}^{M_k} \sum_{j=1}^{N_k} c_k^2(i, j), \quad k = 1, 2, \dots, 8, \quad (9)$$

Где  $c_k(i, j)$  представляют собой коэффициенты двухуровневого wavelet-преобразования изображения (рис.1). Таким образом, общее количество низкоуровневых признаков изображения составляет 32: 24 цветовых признака (по 8 градаций на канал) и 8 текстурных.

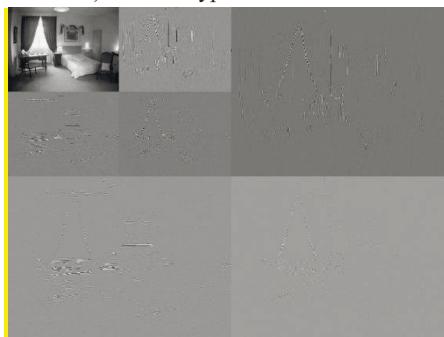


Рис 1: Пример двухуровневого wavelet-преобразования изображения.

### 4. ПРЕДЛАГАЕМЫЙ ПОДХОД К ЧАСТИЧНОМУ ОБУЧЕНИЮ С УЧИТЕЛЕМ

#### 4.1. Общий принцип

При использовании сетевых хранилищ данных число объектов в обучающей выборке заранее неизвестно, что приводит к невозможности использовать итерационное обучение на одних и тех же данных в надежде на то, что каждый шаг будет улучшать качество распознавания классификатора. Так как данные добавляются небольшими порциями, то нельзя гарантировать того, что новая порция не ухудшит результаты. Соответственно, требуется применять различные дополнительные методы для предотвращения ухудшения характеристик классификатора.

Общий алгоритм обучения при использовании сетевых хранилищ можно представить следующим образом:

**Алгоритм 1:** Предлагаемый алгоритм пошагового обучения

- 1: Дано:  $X^{(0)}, Y^{(0)}$
- 2: Для  $t \in \{0, 1, \dots\}$ :
- 3:  $X^{(t+1)} := X^{(t)} \cup Z^{(t)}$
- 4:  $\Lambda^{(t)} = \{x \in X \mid Y^{(t)} \neq \perp\}$
- 5: Тренируем классификатор  $\pi^{t+1}$  на  $(\Lambda^{(t)}, Y^{(t)})$
- 6: Для каждого вектора  $x \in X$ :
- 7:  $\hat{y} := \arg \max_j \pi_x^{(t+1)}(j)$
- 8:  $Y_x^{(t+1)} := \begin{cases} Y_x^{(0)}, & \text{если } x \in \Lambda^{(0)} \\ \hat{y}, & \text{если } x \in \Lambda^{(t)} \vee \pi_x^{(t+1)}(\hat{y}) > \frac{1}{L} \\ \perp, & \text{иначе} \end{cases}$
- 9: Если  $Y^{(t+1)} = Y^{(t)}$ , то конец, иначе шаг 2

Где  $L$  – число возможных меток (иными словами – количество классов);  $X^{(t)}$  – множество маркированных и немаркированных объектов на шаге  $t$ ;  $Z^{(t)}$  – порция немаркированных объектов, сканируемых на шаге  $t$ ;  $\Lambda^{(t)}$  – множество маркированных объектов на шаге  $t$ ;  $Y^{(t)}$  – функция, производящая маркировку каждого объекта;  $\pi_x^{(t+1)}(j)$  – уверенность в принадлежности объекта  $x$  к классу  $j$  на шаге  $t+1$ .

Можно заметить, что вероятность выполнения условия  $Y^{(t+1)} = Y^{(t)}$  достаточно мала, так как вряд ли наступит момент, когда ни одна из меток объектов на очередном шаге не изменится, поэтому данное условие имеет смысл заменить более мягким, например, когда процесс обучения заканчивается при достижении некоторого заданного процента неизменившихся меток. Можно расширить данный алгоритм для случая активного обучения, изменив шаг 8 следующим образом:

**Алгоритм 2:** Шаг, реализующий активное обучение

$$Y_x^{(t+1)} := \begin{cases} Y_x^{(0)}, & \text{if } x \in \Lambda^{(0)} \\ y, & \text{if } \pi_x^{(t+1)}(j) < \pi_{0x}, \\ \hat{y}, & \text{иначе} \end{cases}$$

где  $y$  – метка, выставляемая вручную человеком, и  $\pi_{0x}$  – минимальный порог для отступа, определяющий достаточную уверенность в метке.

#### 4.2. Перемаркировка голосованием

Изначально предполагается, что на каждом новом шаге метки объектов могут изменяться произвольным образом, в зависимости от полученной на предыдущем шаге модели данных. Однако если в случае алгоритмов обучения, использующих заранее известные выборки, данный подход

обеспечивает определённую сходимость, то при порционном скачивании сходимость не гарантируется. К тому же может наблюдаться некоторый отрицательный эффект: нередко с каждым новым шагом качество работы классификатора при таком обучении только снижается.

Для предотвращения подобных проблем целесообразно использовать своего рода «память»: при маркировке объекта запоминается не только метка, но и её отступ (margin). В данном случае под термином «отступ» понимается «степень погруженности» объекта в свой класс, соответственно, чем меньше его значение, тем выше вероятность ошибки. В англоязычной литературе так же часто используется термин *confidence* (степень доверия). В случае если на последующем шаге отступ будет ниже сохранённого, то перемаркировка не производится. Это позволяет, с одной стороны, защитить классификатор от неверной пермаркировки, но с другой – может приводить к накоплению ошибок, когда неверно принятые решения в дальнейшем закрепляются под видом верного. Тем не менее, данной ситуации можно избежать при применении более интеллектуального способа выбора меток, использующего принцип голосования.

$$y_i = \frac{\sum_{j=1}^n p_{i,j} y_{i,j} + p_{i-1} y_{i-1}}{\sum_{j=1}^n p_{i,j} + p_{i-1}}, \quad (10)$$

где  $y_i$  – новая метка объекта на  $i$ -м шаге;  $y_{i,j}$  – метка объекта на  $i$ -м шаге, выставленная  $j$ -м классификатором;  $p_{i,j}$  – отступ метки  $y_{i,j}$ ;  $y_{i-1}$  – метка объекта на  $i-1$  шаге;  $p_{i-1}$  – отступ метки  $y_{i-1}$ .

Использование подобной схемы позволяет предотвратить накопление ошибок, реализуя при этом защиту от принятия неправильных решений в ходе маркировки.

### 4.3. Фиксация результатов

К сожалению, перемаркировка голосованием в чистом виде эффективна далеко не всегда, поэтому может применяться и более «жёсткий» способ сохранения достигнутого качества распознавания: в случае, если на очередной итерации работа классификатора улучшилась, производится фиксация данных: все метки, полученные на этом шаге, жёстко фиксируются без возможности дальнейшего изменения. Алгоритмический процесс выглядит следующим образом:

**Алгоритм 3:** Алгоритм пошагового обучения с фиксацией

- 1: Дано:  $X^{(0)}, Y^{(0)}, \Lambda_0$
- 2: Для  $t \in \{0, 1, \dots\}$ :
- 3:  $X^{(t+1)} := X^{(t)} \cup Z^{(t)}$
- 4:  $\Lambda^{(t)} = \{x \in X \mid Y^{(t)} \neq \perp\}$
- 5: Тренируем классификатор  $\pi^{t+1}$  на  $(\Lambda_0 \cup \Lambda^{(t)}, Y^{(t)})$
- 6: Для каждого вектора  $x \in X$ :
- 7:  $\hat{y} := \arg \max_j \pi_x^{(t+1)}(j)$
- 8:  $Y_x^{(t+1)} := \begin{cases} Y_x^{(0)}, & \text{если } x \in \Lambda^{(0)} \\ \hat{y}, & \text{если } x \in \Lambda^{(t)} \vee \pi_x^{(t+1)}(\hat{y}) > \frac{1}{L} \\ \perp, & \text{иначе} \end{cases}$
- 9: Если классификатор  $\pi^{t+1}$  лучше, чем  $\pi_0$ , то  $\pi_0 := \pi^{t+1}$
- 10:  $\Lambda_0 := \Lambda_0 \cup \Lambda^{(t)}$
- 11: Если  $Y^{(t+1)} = Y^{(t)}$ , то конец, иначе шаг 2

где  $\Lambda_0$  – множество объектов с фиксированными метками;  $\pi_0$  – зафиксированный классификатор.

Как правило, не имеет смысла фиксировать сам классификатор, достаточно лишь сохранить параметры его

работы, чтобы их можно было сравнивать с результатами других классификаторов. Подобными параметрами могут быть, например, accuracy, precision и recall.

### 4.4. Фильтрация Данных

Известно, что SVM весьма чувствительны к шумовым выбросам (неверно промаркованных объектам), поэтому фильтрация шумов является достаточно важным процессом, позволяющим заметно повысить эффективность работы машины опорных векторов. В задачах автоматического обучения ей часто пренебрегают, так как шум не оказывает заметного влияния на результаты. Однако при использовании итерационной модели обучения шумы могут негативно влиять на обучаемый классификатор, так как присутствует эффект накопления ошибок, описанный ранее (рис. 2).

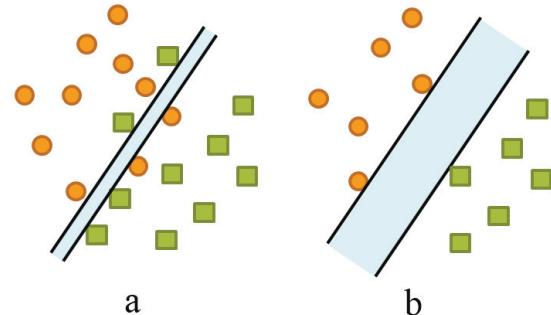


Рис 2: Фильтрация данных: а – без фильтрации, б – с фильтрацией.

Можно видеть, что при использовании фильтрации оптимальная разделяющая гиперплоскость, во-первых, будет шире, а во-вторых, число ошибочно классифицированных объектов уменьшится. Так же плюсом фильтрации является то, что увеличивается скорость обучения за счёт уменьшения числа объектов в обучающей выборке.

Наиболее простым, но при этом эффективным, методом отсечения шумов является отсечение по порогу: если отступ метки оказывается ниже некоторого минимума, то объект, соответствующий данной метке, удаляется из обучающей выборки. Однако при этом отсекается и часть полезной информации, так как низкое значение отступа лишь свидетельствует о высокой вероятности ошибки, но не гарантирует её наличия. Тем не менее, экспериментальные результаты показывают, что даже такой грубый подход оказывается лучше, чем полное отсутствие фильтрации.

## 5. ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

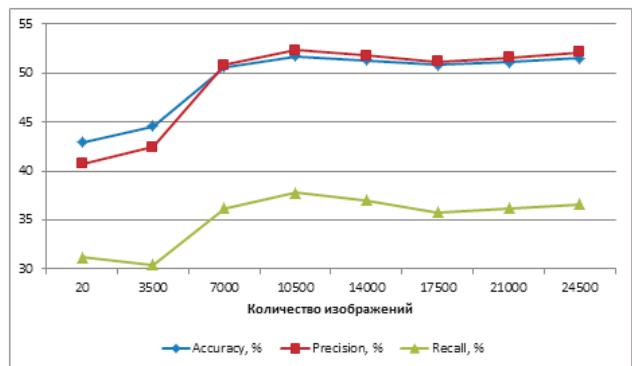
Для тестирования предложенного алгоритма пошагового обучения нами с сервиса Flickr были скачаны фотографии, соответствующие двум классам: со сценами внутри и вне помещений. Поиск фотографий производился по текстовым тегам ‘indoor’ и ‘architecture’ для изображений внутри помещений и по тегу ‘landscape’ вне помещений. Никакого ручного деления на классы в обучающей выборке не производилось. Для унификации процесса вычисления низкоуровневых признаков изображений все фотографии были масштабированы до размера 500x500 пикселов, при этом важным моментом является нормализация этих признаков: как было показано в [10], она позволяет улучшить качество распознавания. Однако так как в случае заранее неизвестного числа объектов в обучающей выборке

нормализовать все признаки достаточно затруднительно, было принято решение произвести лишь приблизительную нормализацию, когда часть значений может выходить за границы заданного диапазона [0, 1]. Для обеспечения корректности результатов тестирования нами был сформирован набор из 30,000 изображений двух классов. Исходная обучающая выборка (состоящая из фотографий, классы которых известны) состояла из 20 изображений (рисунок 4).



**Рис 3:** Изображения исходной обучающей выборки. Как можно видеть, некоторые из них (2-4) нельзя отнести ни к фотографиям, содержащим сцены внутри помещений, ни к фотографиям, содержащим сцены вне помещений.

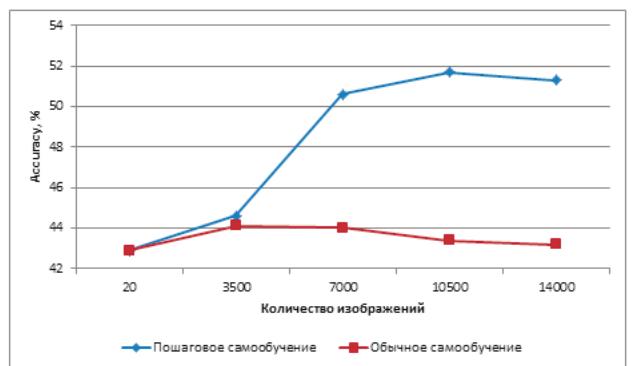
При этом в данных наборах количество изображений обоих классов было одинаковым. Размер одной порции скачанных данных на каждом шаге алгоритма составлял 3500 изображений. Для оценки качества распознавания с помощью метрик accuracy, precision и recall использовался набор из 1000 фотографий, несвязанных с основной выборкой. Параметр веса С у SVM был равен С=1. При этом использовалось ядро  $\chi^2_{exp}$  со значением параметра  $\gamma = 0.5$ .



**Рис 4:** Пошаговое самообучение. Перемаркировка голосованием, фиксация результатов и фильтрация включены. Максимальные значения: accuracy = 51.7%, precision = 52.4%, recall = 37.8%.

На первом этапе мы протестировали наш алгоритм пошагового обучения с использованием метода самообучения. Порог фильтрации при этом был равен 0.6 (фильтрация производилась по параметру probability, использующемуся в LIBSVM как мера оценки уверенности принадлежности метки какому-либо классу). На первой итерации было отфильтровано порядка 20% изображений, на последующих – около 1%. Как можно видеть на рисунке 5, пошаговое самообучение оказалось эффективно на нескольких начальных итерациях обучения, однако в дальнейшем его эффективность снизилась.

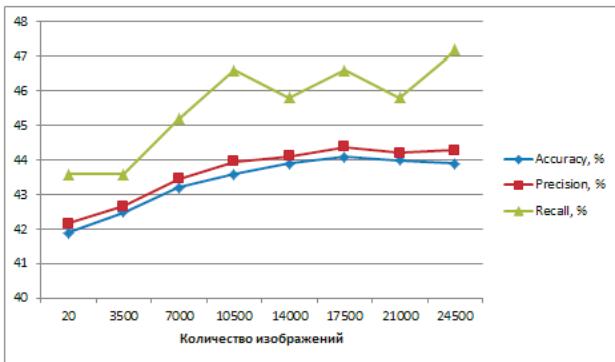
Затем мы провели сравнение результатов, показываемых предлагаемым алгоритмом пошагового самообучения и обычным алгоритмом самообучения, реализованным на основе модифицированного алгоритма Яровского. Тестирование последнего проводилось следующим образом: исходная обучающая выборка, как и в случае пошагового самообучения, состояла из 20 фотографий, к которым затем добавлялось некоторое число неклассифицированных изображений, после чего проводилось обучение. Как можно видеть на рисунке 6, обычное самообучение, в отличие от пошагового, при данных условиях оказалось полностью неэффективно.



**Рис 5:** Сравнение обычного самообучения и пошагового самообучения. Максимальные значения: Пошаговое самообучение: accuracy = 51.7%, Обычное самообучение: accuracy = 44.1%.

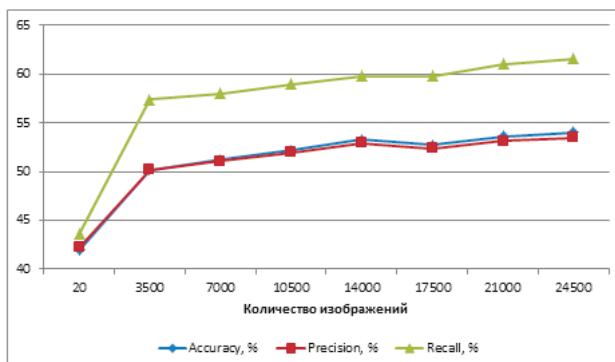
Ситуация с совместным обучением была полностью иная: результаты пошагового и обычного совместного обучения оказались достаточно близки, при этом использование

дополнительных методов улучшения качества обучения, таких как перемаркировка голосованием, фиксация результатов и фильтрация, не несла практической пользы: работа классификатора при их использовании не улучшилась. Результаты пошагового совместного обучения показаны на рисунке 7.



**Рис 6:** Пошаговое совместное обучение. Перемаркировка голосованием, фиксация результатов и фильтрация выключены. Максимальные значения: accuracy = 44.1%, precision = 44.4%, recall = 47.2%.

Наконец, мы протестировали пошаговое само- и совместное обучение, совместив его с методом активного обучения. При этом вручную марковались только изображения, которые в противном случае были бы отфильтрованы по порогу 0.6, указанному ранее. Соответственно, фильтрация не использовалась. Наибольшая эффективность при использовании активного обучения была достигнута в случае пошагового совместного активного обучения. Можно заметить, что оно, в целом, оказалось значительно более результативным, чем во всех описанных ранее случаях (рисунок 8).



**Рис 7:** Пошаговое активное совместное обучение.

Перемаркировка голосованием и фиксация результатов включены, фильтрация выключена. Максимальные значения: accuracy = 54.0%, precision = 53.5%, recall = 61.6%

Как можно видеть, в случае использования экстремально малых исходных выборок предложенные модификации алгоритмов пошагового обучения с учителем позволяют достичь лучших результатов, чем хорошо известные существующие алгоритмы.

## 6. ССЫЛКИ

- [1] Avrim Blum, Tom Mitchell. “Combining labeled and unlabeled data with co-training”, COLT' 98 Proceedings of

the eleventh annual conference on Computational learning theory, pp. 92 – 100, 1998.

- [2] Ulf Brefeld, Thomas Gärtner, Tobias Scheffer, Stefan Wrobel. “Efficient co-regularised least squares regression”, ICML '06 Proceedings of the 23rd international conference on Machine learning, pp. 137 – 144, 2006.
- [3] Chih-Chung Chang, Chih-Jen Lin. “LIBSVM – A Library for Support Vector Machines”, ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011.
- [4] George Forman, Ira Cohen. “Learning from Little: Comparison of Classifiers Given Little Training”, HP Laboratories Palo Alto, 2004.
- [5] Gholamreza Haffari, Anoop Sarkar. “Analysis of semi-supervised learning with the Yarowsky algorithm”, 2007.
- [6] Maji, S., Berg, A.C., Malik, J. “Classification using intersection kernel support vector machines is efficient”, Computer Vision and Pattern Recognition, pp. 1–8, 2008.
- [7] Perronnin F., Sanchez J. and Yan Liu. “Large-scale image categorization with explicit data embedding”, Computer Vision and Pattern Recognition (CVPR), pp. 2297 – 2304, 2010.
- [8] Ioannis Pratikakis, Basilio Gatos, Stelios C. A. Thomopoulos. “Scene categorization using low-level visual features”.
- [9] Chuck Rosenberg, Martial Hebert, Henry Schneiderman. “Semi-supervised self-training of object detection models”, Seventh IEEE Workshop on Applications of Computer Vision, January, 2005.
- [10] W. S. Sarle. “Neural Network FAQ”, 1997.
- [11] Navid Serrano, Andreas E. Savakis, Jiebo Luo. “Improved scene classification using efficient low-level features and semantic cues”, Pattern Recognition, Volume 37, Issue 9, pp. 1773–1784, 2004.
- [12] Burr Settles. “Active learning literature survey”, Computer Sciences Technical Report 1648, University of Wisconsin-Madison, Vol. 1648. 2010.
- [13] Gokhan Tura, Dilek Hakkani-Türa, Robert E. Schapire. “Combining active and semi-supervised learning for spoken language understanding”, Speech Communication, Volume 45, Issue 2, pp 171–186, 2005.
- [14] Vapnik V., Chapelle O. “Bounds on error expectation for support vector machines”, Neural Computation, Vol.12, no.9. P. 2013 – 2036, 2000.
- [15] X. Zhu. “Semi-supervised learning literature survey”, Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005/9.

## Об авторах

Сергей Завалишин – студент-дипломник НИЯУ «МИФИ». Илья Сафонов – к.т.н., доцент НИЯУ «МИФИ».

# О методах подсчета посетителей при зенитном расположении видеокамеры

Петр Харебов, Рустам Салимзибиров

Кафедра компьютерных систем и телекоммуникаций

Пермский государственный национальный исследовательский университет, Пермь, Россия

kharebov.p@yandex.ru, rustamse@gmail.com

## Аннотация

В настоящей работе сравниваются два метода автоматического подсчета посетителей при анализе видеопотока от зенитной камеры (направленной вертикально вниз). Первый метод, основанный на анализе траекторий движения, использует методы вычитания фона для выделения движущихся объектов и поиск максимально вероятного набора траекторий. Второй метод, основанный на анализе оптического потока и пространственно-временной кластеризации с помощью модифицированного алгоритма DBSCAN. Показаны границы применимости и предложен метод совместного использования обоих подходов для улучшения качества подсчета.

**Ключевые слова:** Подсчет посетителей, трекинг, оптический поток, DBSCAN.

## 1. ВВЕДЕНИЕ

Автоматический подсчет посетителей пользуется большим практическим и коммерческим интересом [4]. Подсчет посетителей позволяет контролировать посещаемость, производить мониторинг эффективности предприятия, позволяет увеличить безопасность объекта. Системы подсчета посетителей устанавливаются в торговых центрах, футбольных стадионах, метро, автобусных остановках, других местах скопления людей [2]. Модель трафика посетителей позволяет оптимизировать распределение ресурсов предприятия, может являться основанием для проведения маркетинговых мероприятий и для оперативного принятия других управленческих решений.

Существует несколько основных способов автоматического подсчета посетителей [4]: механический способ, на основе датчиков (тепловых и пересечения инфракрасного луча), и на основе анализа видеопотока. По сравнению с другими методами, анализ видеопотока выигрывает за счет того, что не задерживает посетителей как механические турникеты, и может работать при большей плотности людей, по сравнению с тепловыми и инфракрасными датчиками.



Рис 1: Зенитное расположение видеокамеры.

Задача стоит следующим образом. Видеокамера, расположенная под потолком («в зените»), направлена либо вертикально вниз, либо с достаточно небольшим отклонением от вертикали. Расположение видеокамеры в здании таково, что посетители, отображенные в видеопотоке, движутся в основном параллельно с постоянной скоростью. Необходимо подсчитать каждого посетителя и определить направление его движения – вход или выход.

На рис. 1 показан пример зенитного расположения видеокамеры; показана также заданная оператором линия, которую надо пересечь, чтобы быть зафиксированным системой подсчета и отображено количество зафиксированных посетителей. В данном случае, который является наиболее простым для видеоанализа, видеокамера расположена над входом, в кадре может быть не более одного человека, и движение происходит редко.



Рис 2: Взаимное пересечение посетителей.

Наиболее сложная проблема, которая возникает при решении данной задачи – это учет посетителей, которые перекрывают друг друга, либо движутся в группе и соприкасаются. На рис. 2 показано наклонное положение камеры, в таком случае видеоанализ усложняется из-за частого пересечения посетителями друг друга. В случае же зенитного расположения камеры данная проблема стоит менее остро.

## 2. СУЩЕСТВУЮЩИЕ ПОДХОДЫ

Можно выделить два подхода к решению задачи подсчета прошедших людей – прямой и косвенный [9,14].

При прямом подходе к решению данной задачи люди непосредственно выделяются и отслеживаются, строится траектория их движения. Данный метод применим при небольшом взаимном перекрытии движущихся объектов, когда их можно легко отделить друг от друга.

Косвенные же методы не строят траекторию непосредственно. Они используют вычисление характеристик особенностей изображения (края, углы, особые точки, информация о текстуре и т.д. [9]), или определение характеристик оптического потока. Данные

методы применимы для плотного потока людей, когда традиционные методы отслеживания движущихся объектов (трекинга) неприменимы. Лучший результат достигается, когда плотность потока людей приблизительно однородна, иначе, возникают большие ошибки оценивания.

## 2.1 Прямой подход

От выбора метода зависит, в каких условиях сможет работать система подсчета посетителей, и как может быть расположена видеокамера. В случае использования зенитных камер, направленных вертикально вниз, можно добиться хорошего отделения людей друг от друга. В таком случае, хорошие результаты будут давать классические методы трекинга, основанные на методе вычитания фона [21,22] и простого связывания ближайших выделенных областей переднего плана на последовательных видеокадрах. Т.к. перекрытия людей в данном случае быть практически не может, можно задать типичные размеры человека, таким образом, решая задачу определения количества людей в группе.

Сегментация группы людей для подсчета каждого посетителя возможно не только с помощью разделения, учитывающего площадь, занимаемую одним человеком [4,7], но и с помощью алгоритмов кластеризации, например, k-means [2].

Авторы [12] показывают, что точность подсчета прошедших людей достигает 99%. В более сложных условиях, когда камера расположена не строго сверху, и люди могут перекрывать друг друга, используются более сложные методы отслеживания. Например: поиск и отслеживание определенных частей человеческого тела (головы и плеч [16,17], головы [15]), отслеживание особых точек [14], и другие методы, используемые для решения задачи трекинга движущихся объектов, ознакомиться с которыми можно в работе [19].

## 2.2 Косвенный подход

Используя информацию о величине и направлении векторов оптического потока на последовательности видеокадров, можно определить направление движения и количество посетителей [5, 13].

Определение статистических характеристик особенностей изображения может дополнять анализ оптического потока [1], либо использоваться независимо [3]. Устойчивость данного подхода позволяет приблизительно подсчитать количество людей в толпе [3,14]. При определении статистических характеристик особенностей изображения считается, что количество людей в определенной области зависит от количества выделенных особенностей изображения. Обычно [1,18], хотя и не всегда, количество людей прямо пропорционально количеству особенностей. В работах [9,10] авторы предлагают учет перспективы, учет уменьшения количества детектированных особенностей при увеличении плотности потока людей и др.

В настоящей работе сравниваются два метода подсчета посетителей. Прямой метод, с помощью отслеживания движущихся объектов, основанный на методе «вычитании фона», описывается в главе 3. Косвенный метод, основанный на анализе оптического потока, описывается в главе 4. Показаны границы применимости обоих методов. В главе 5 предложен метод совместного использования обоих подходов для улучшения качества подсчета.

Данные метода были выбраны, т.к. они не требовательны к вычислительным ресурсам и могут давать высокую точность, около 89-99% при зенитном расположении камеры [12,13]. Кроме того, данные методы более

устойчивы в случае зенитного расположения камеры, а для определения особенностей изображения желательно наклонное расположение.

## 3. ПОДСЧЕТ ПОСЕТИТЕЛЕЙ С ПОМОЩЬЮ ТРЕКИНГА

Выделение и отслеживание движущихся объектов с помощью методов вычитания фона является стандартной практикой [21,22]. Методы вычитания фона не позволяют отслеживать перекрывающиеся объекты, для этого существует множество алгоритмов [19], но не существует универсального. В нашем варианте случаи пересечения объектов обрабатываются с помощью поиска максимально вероятного набора траекторий, данная задача сводится к поиску минимального потока в сети [20]. При этом используется информация о размере, положении объектов, направлении движения до и после пересечения.

Особенностью данной задачи, в отличие от общей задачи трекинга, является «стерильность» условий – стабильность заднего плана: искусственное освещение, отсутствие изменения геометрии (нет оставленных и унесенных объектов). Поэтому подсчет на основе трекинга дает точность, близкую к 100% при достаточно редком потоке посетителей, когда они движутся отдельно. Кроме того, учет истории движения (анализ траектории объекта) позволяет отфильтровывать блики и шумы. Анализ траектории также позволяет отфильтровывать ситуации, когда один человек стоит под камерой и пересекает линию несколько раз туда-обратно.

Точность подсчета сильно зависит от положения камеры и величины потока посетителей. Для достижения максимальной точности видеокамера должна быть строго сверху, чтобы минимизировать взаимные перекрытия движущихся объектов друг другом, а поток людей должен быть достаточно редким.

## 4. АНАЛИЗ ОПТИЧЕСКОГО ПОТОКА

Анализ оптического потока позволяет производить подсчет посетителей в условиях плотного движения.

В данной работе используется метод минимизации суммы абсолютной разности блоков для вычисления оптического потока [6]. Оптический поток вычисляется для линии пересечения [5,8,13], как показано на рис. 3.

На рис. 3 показаны вектора перемещения при движении человека снизу вверх.

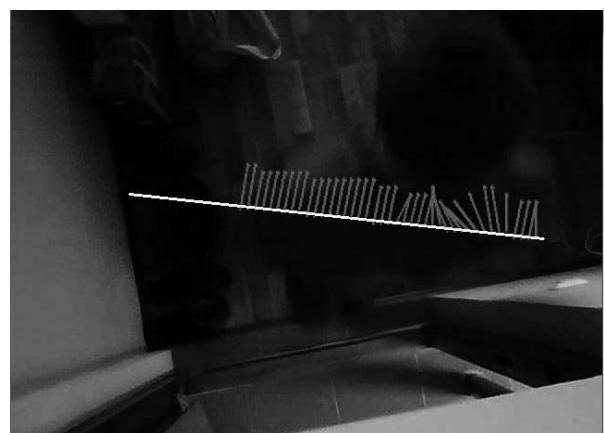


Рис 3: Вектора перемещения.

Анализ векторов перемещения, расположенных только на линии пересечения, позволяет значительно экономить на

вычислительных ресурсах, т.к. вычисление оптического потока для всего изображения – это вычислительно сложная задача и, как правило, она не может выполняться в режиме реального времени.

Пусть  $L^t$  – линия пересечения в момент времени  $t$ . Разобьём всю линию на  $N$  равноудаленных друг от друга точек и будем производить расчет векторов перемещения оптического потока для каждой точки.

Таким образом,  $\tilde{l}_i^t$  – вектор перемещения на линии с индексом  $i$ ,  $i \in (1..N)$ .

Вычисление векторов перемещения для линии производится для двух соседних видеокадров со временами  $t$  и  $t-1$ .

Рассмотрим упорядоченную по времени последовательность линий  $L^t, L^{t+1}, L^{t+2}, L^{t+3}$  и т.д. При пересечении посетителем линии подсчета будет наблюдаться кластер, состоящий из пикселов со сходным направлением векторов перемещения (см. рис. 4).

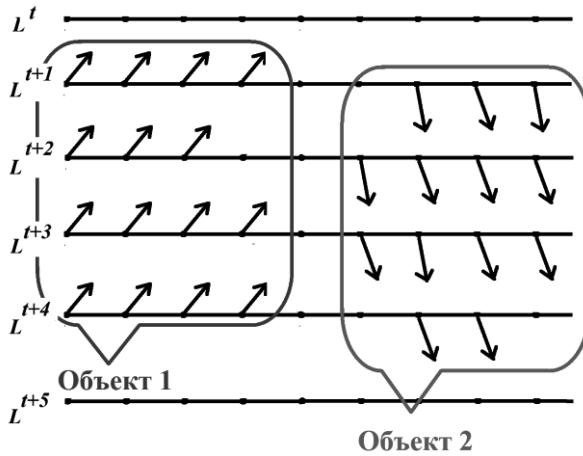


Рис 4: Кластеризация векторов перемещения для последовательности линий пересечения.

Задачу подсчета посетителей сведем к задаче кластеризации векторов перемещения.

Посетители движутся в основном перпендикулярно линии пересечения, и основной интерес представляет проекция на перпендикуляр к линии пересечения  $l_{\perp i}^t$  (см. рис. 5).

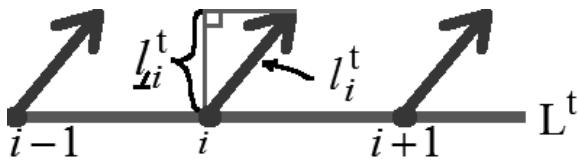


Рис 5: Проекции вектора перемещения.

Площадью кластера будем считать сумму проекций векторов перемещения, составляющих кластер:

$$S = \sum_{i,t} l_{\perp i}^t, \text{ где } i,t \text{ - принадлежат кластеру.}$$

В основу алгоритма кластеризации был положен плотностный алгоритм кластеризации пространственных данных (DBSCAN) [11]. Алгоритм DBSCAN выбран, т.к. он не требует априорного знания количества кластеров и

обладает высокой устойчивостью к шуму [11]. Шум присутствует в значениях векторов перемещения, возникающий из-за неточности алгоритма оптического потока и не строгой направленности векторов перемещения движущегося объекта (не все части движущегося объекта движутся однородно и с одинаковой скоростью в момент пересечения линии).

Использование алгоритма DBSCAN ко всему набору анализируемых данных на каждом кадре невозможно из-за его большой вычислительной сложности. Поэтому он был изменен так, что обрабатывается только небольшая часть новых данных при учете данных уже проанализированных.

#### 4.1 Алгоритм кластеризации

В классическом алгоритме DBSCAN [11] для отнесения рассматриваемой точки к кластеру необходимо сравнить данную точку со всеми другими. Если найдено не менее  $M$  близких точек (с расстоянием менее  $\varepsilon$ ), то группа точек считается кластером.

Введем функцию расстояния между двумя векторами перемещения:

$$d(l_{i1}^{t1}, l_{i2}^{t2}) = \begin{cases} \infty, & \text{если } l_{\perp i1}^{t1} l_{\perp i2}^{t2} \leq 0 \\ \sqrt{\alpha \cdot (t1 - t2)^2 + \beta \cdot (i1 - i2)^2}, & \text{если } l_{\perp i1}^{t1} l_{\perp i2}^{t2} > 0, \end{cases} \quad (1)$$

где  $\alpha, \beta$  – параметры алгоритма. Произведение проекций  $l_{\perp i1}^{t1} l_{\perp i2}^{t2}$  больше нуля, если проекции векторов одного знака.

Вектора  $l_{i1}^{t1}$  и  $l_{i2}^{t2}$  будем называть близким, если

$$d(l_{i1}^{t1}, l_{i2}^{t2}) < \varepsilon, \quad (2)$$

где  $\varepsilon$  – параметр алгоритма.

Тогда, из (1) и (2):

$$\sqrt{\alpha \cdot (t1 - t2)^2 + \beta \cdot (i1 - i2)^2} < \varepsilon \quad (3).$$

Заметим, что в неравенстве (3) при любых значениях  $t1, t2, i1, i2$  всегда выполняются следующие условия:

$$|t1 - t2| < \frac{\varepsilon}{\alpha} \text{ и } |i1 - i2| < \frac{\varepsilon}{\beta} \quad (4).$$

Из (4) следует, что вектора перемещения могут быть близкими только при достаточно близких значениях  $t1, t2$  и  $i1, i2$ , а это означает, что можно ограничить область поиска окном размером  $\frac{2\varepsilon}{\alpha}$  на  $\frac{2\varepsilon}{\beta}$ .

Для каждого вектора перемещения  $l_i^t$  на линии  $L^t$  будем искать близкие вектора в окне размером  $\frac{2\varepsilon}{\alpha}$  на  $\frac{2\varepsilon}{\beta}$ .

При анализе каждый вектор перемещения может находиться в двух состояниях: не принадлежащий ни одному кластеру и принадлежащий кластеру с индексом  $C_i$ . Анализ только части векторов (по сравнению со стандартным алгоритмом DBSCAN) приводит к возможности появления двух близкорасположенных

кластеров, а настоящий модифицированный алгоритм предусматривает их слияние.

Если анализируемый вектор имеет не менее  $M$  близких векторов, и анализируемый вектор принадлежит существующему кластеру с индексом  $C_i$ , то все близкие вектора так же отмечаются принадлежащими кластеру  $C_i$ . Если обнаруживается близкий вектор, принадлежащий кластеру  $C_j$ , то производится слияние двух кластеров.

Если анализируемый вектор не принадлежит ни к одному кластеру, то создается новый кластер с индексом  $C_i$  и все его близкие вектора так же относятся к этому кластеру. Также, если обнаруживается близкий вектор, принадлежащий другому кластеру с индексом  $C_j$ , то производится слияние двух кластеров.

## 5. КОМБИНИРОВАННЫЙ МЕТОД

При комбинации двух предложенных методов можно увеличить точность подсчета посетителей. При анализе траекторий с помощью трекинга, описанного в 3 главе, используется информация об оптическом потоке следующим образом. Если площадь выделенной области переднего плана превышает заданный порог, то для подсчета посетителей используется анализ оптического потока, а информация о траекториях не используется. Кроме того, анализ траектории используется для определения площади кластера, соответствующей одному посетителю.

## 6. РЕЗУЛЬТАТЫ

Подсчет посетителей тестирулся в двух вариантах – с помощью зенитных камер и с помощью наклонных, как показано на рис. 1 и 2. Для каждого случая было проанализировано 4 видеоролика от разных камер длительностью от 2 до 10 минут.

Для каждого случая была вычислена точность подсчета, равная отношению количества верно подсчитанных посетителей к количеству реально прошедших. Данная величина была усреднена по роликам.

В таблице 1 показаны результаты подсчета с помощью метода, описанного в 3 главе. В зависимости от условий, метод, основанный на анализе траекторий, может иметь точность  $>99\%$ . Для достижения такой точности пересечения объектов в кадре должны быть исключены. С другой стороны, данный метод дает очень низкую точность при частом появлении групп людей, когда алгоритм трекинга не может корректно найти траектории движения.

Условия	Расположение камеры	Точность
Редкое появление посетителей, групп людей нет.	Зенитная	$>99\%$
	Наклонная	95%
Нечастое появление групп людей.	Зенитная	75%
	Наклонная	60%
Частое появление групп людей.	Зенитная	25%
	Наклонная	15%

Таблица 1: Анализ траекторий.

Результаты подсчета посетителей при анализе оптического потока, показаны в таблице 2. Видно, что в случае редкого прохождения посетителей, данный метод имеет меньшую точность, по сравнению с методом анализа траекторий. Это связано с неточной работой метода нахождения

оптического потока. Но данный метод выигрывает в случае, если посетители ходят группами.

Условия	Расположение камеры	Точность
Редкое появление посетителей, групп людей нет.	Зенитная	85%
	Наклонная	80%
Нечастое появление групп людей.	Зенитная	80%
	Наклонная	70%
Частое появление групп людей.	Зенитная	50%
	Наклонная	30%

Таблица 2: Анализ оптического потока.

Комбинированный метод демонстрирует результаты, превосходящие оба предложенных метода по отдельности. Рост точности подсчета людей в случае появления групп людей связан с тем, что периодически проходили и одиночные посетители, когда подсчет производился с помощью анализа траекторий. В случае же редкого появления посетителей использовался метод анализа траекторий, поэтому точность не изменилась (ср. с таблицей 1).

Условия	Расположение камеры	Точность
Редкое появление посетителей, групп людей нет.	Зенитная	$>99\%$
	Наклонная	95%
Нечастое появление групп людей.	Зенитная	80%
	Наклонная	75%
Частое появление групп людей.	Зенитная	55%
	Наклонная	35%

Таблица 3: Комбинированный метод.

## 7. ЗАКЛЮЧЕНИЕ

В данной работе предложен метод подсчета посетителей, объединяющий прямой и косвенный подходы, что позволило добиться улучшения точности подсчета. Предложенный метод позволяет обрабатывать 16 камер с частотой 30 кадров/с на компьютере с процессором Intel i5-2400.

Показано, что метод, основанный на прямом выделении объектов, работает лучше или сопоставимо с анализом оптического потока при небольшом количестве посетителей, движущихся в группах. В иных случаях, анализ оптического потока работает корректнее.

В дальнейшем планируется улучшить анализ траекторий посетителей с помощью поиска и отслеживания головы и плеч людей, как наиболее отличительной части человеческого тела [17], что предположительно, может привести к большей точности подсчета в случае плотных групп посетителей.

## 8. ССЫЛКИ

- [1] Albiol A., Mar'ia Julia Silla, Alberto Albiol and Jose Manuel Mossi, Video Analysis Using Corner Motion Statistics, *Performance Evaluation of Tracking and Surveillance workshop at CVPR 2009* (2009), pp. 31-37, 2009

- [2] Antic, B.; Letic, D.; Culibrk, D.; Crnojevic, V. K-Means Based Segmentation For Real-Time Zenithal People Counting. *Image Processing (ICIP), 2009 16th IEEE International Conference*, pp 2565 – 2568, 2009
- [3] Antoni B. Chan Zhang-Sheng John Liang Nuno Vasconcelos, Privacy Preserving Crowd Monitoring: Counting People without People Models or Tracking. *IEEE Conf. on Computer Vision and Pattern Recognition, Anchorage, 2008*
- [4] Barandiaran, J.; Murguia, B.; Boto, F. Real-Time People Counting Using Multiple Lines. *Image Analysis for Multimedia Interactive Services*, pp. 159 – 162, 2008.
- [5] Benabbas Y., Nacim Ihaddadene, Tarek Yahiaoui, Thierry Urruty and Chabane Djeraba, Spatio-Temporal Optical Flow Analysis for People Counting, *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference*, pp. 212 – 217, 2010
- [6] Catalano G; Alessio Gallace; Bomi Kim; Sergio Pedro; Optical Flow, 2009
- [7] Chao-Ho Chen, Tsong-Yi Chen, Da-Jinn Wang, and Tsang-Jie Chen, A Cost-Eective People-Counter for a Crowd of Moving People Based on Two-Stage Segmentation, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 3, No. 1, pp. 12-23, January 2012
- [8] Cong Y., H. F. Gong, S. C. Zhu, Y. D. Tang Flow Mosaicking: Real-time Pedestrian Counting without Scene-specific Learning, *CVPR 2009*, pp. 1093-1100, 2009
- [9] Conte, D. Foggia, P. Percannella, G.; Tufano, F.; Vento, M. A method for counting people in crowded scenes, *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference*, pp. 225 - 232, 2010
- [10] Conte D., Pasquale Foggia, Gennaro Percannella, Francesco Tufano, Mario Vento, Counting Moving People in Videos by Salient Points Detection, *Pattern Recognition (ICPR), 2010 20th International Conference*, pp. 1743-1746, 2010
- [11] Ester M., Hans-Peter Kriegel, Jrg Sander, Xiaowei Xu, A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, *KDD-96 Proceedings*, pp. 226 - 231, 1996
- [12] Khemlani A, Kester Duncan, and Sudeep Sarkar, People Counter: Counting of Mostly Static People in Indoor Conditions
- [13] Lamovsky D, R. Sadykhov, Method Of Pedestrians Traffic Assessment Based On Analysis Of Video Data In Surveillance Systems, *MIPRO, 2010 Proceedings of the 33rd International Convention*, pp. 704 – 706, 2010
- [14] Rabaud V., S. Belongie, Counting Crowded Moving Objects, *CVPR '06 Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, pp. 705 – 711, 2006
- [15] Rodriguez M., I. Laptev J. Sivic JY. Audibert, Density-aware person detection and tracking in crowds, *Proceedings of the IEEE International Conference on Computer Vision*, 2011
- [16] Sidla, O.; Lypetskyy, Y.; Brandle, N.; Seer, S. Pedestrian Detection and Tracking for Counting Applications in Crowded Situations, *Video and Signal Based Surveillance, 2006. AVSS '06. IEEE International Conference*, 2006
- [17] Xing J., Haizhou Ai, Liwei Liu, Shihong Lao, Robust Crowd Counting Using Detection Flow, *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pp. 2061 - 2064 , 2011
- [18] Yang T., Yanning Zhang, Dapei Shao, Ying Li, Clustering method for counting passengers getting in a bus with single camera, *Optical Engineering, Volume 49, Issue 3, pp. 037203-037203-10*, 2010
- [19] Yilmaz A, Object Tracking: A Survey, *ACM Computing Surveys, Vol. 38, No. 4, Article 13, Publication, 2006*
- [20] Zhang Li, Yuan Li and Ramakant Nevatia, Global Data Association for Multi-Object Tracking Using Network Flows, *CVPR 2008*
- [21] Салимзубаров Р., Разумков А., Коробков А. Адаптивный метод выделения движущихся объектов. *Труды конференции Графикон, СПб., 2010.* 333 с.
- [22] Харебов П., Новиков С. Проблемы выделения объектов в компрессированном потоке изображений. *Труды конференции Графикон, 2009.*

## Об авторах

Петр Харебов – к.ф.-м.н., старший преподаватель ККСиТ ПГНИУ, ведущий разработчик ООО «Сателлит». Его адрес: [kharebov.p@yandex.ru](mailto:kharebov.p@yandex.ru).

Рустам Салимзубаров – аспирант ККСиТ ПГНИУ, ведущий разработчик ООО «Сателлит». Его адрес: [rustamse@gmail.com](mailto:rustamse@gmail.com).

# Визуализация сейсмических данных на виртуальном глобусе

Александр Бобков, Андрей Леонов, Виктор Чебров

Московский физико-технический институт,

Институт истории естествознания и техники им. С.И. Вавилова РАН,

Камчатский филиал Геофизической службы РАН

alexander.e.bobkov@gmail.com, a.leonov@ihst.ru, chebr@emsd.ru

## Аннотация

В статье рассматривается трехмерная визуализация сейсмических данных на основе полупрозрачного виртуального глобуса: гипоцентры<sup>1</sup>, зоны субдукции<sup>2</sup>, сейсмические регионы, данные сейсмотомографии и др. Предложен метод рендеринга и фильтрации массивов точечных данных (каталогов гипоцентров) на GPU, методы рендеринга объемных данных (сейсмотомографии) с учетом кривизны Земли, методы корректного рендеринга полупрозрачного рельефа согласованно с другими объектами, методы визуализации и анимации различных подземных процессов. Разработано приложение для одновременной 3D-визуализации перечисленных выше сейсмических данных в едином контексте с возможностью гибкой настройки режимов отображения глобуса. Поддержано отображение в различных стереорежимах (горизонтальная стереопара, квадробуфер).

**Keywords:** виртуальный глобус, виртуальное окружение, геофизика, землетрясения, сейсмотомография.

## 1. ВВЕДЕНИЕ

В настоящее время в геофизике активно развиваются методы трехмерного моделирования глубинных структур, визуального анализа данных, прежде всего, полученных на основе сетей инструментальных наблюдений [1][2]. Растет число сейсмостанций, существенно возрастает объем регистрируемых данных. Это связано как с развитием техники регистрации, так и с увеличением мощности программно-аппаратных средств обработки данных. Однако в сфере представления данных по-прежнему используются в основном консервативные методы: плоские карты поверхности и сечения Земли.

В то же время, в последние годы значительное развитие получило направление виртуальных глобусов, которые могут использоваться для трехмерной визуализации геопространственных данных. В 2000 году была начата работа над виртуальным глобусом Keyhole, который в 2005 году был выпущен в свободном доступе как Google Earth и стал факто первым всемирно известным виртуальным глобусом [3]. Позже появился целый ряд аналогичных продуктов.

Развитие виртуальных глобусов связано с тремя основными причинами:

- Появлением данных дистанционного зондирования Земли (ДЗЗ) высокого разрешения (спутниковых снимков, цифровых моделей рельефа), в том числе, свободно доступных в интернете;
- Распространением скоростного доступа к интернету, что дало возможность в реальном времени передавать с удаленных серверов на локальный компьютер данные ДЗЗ, объемы которых измеряются терабайтами;
- Развитием технических и алгоритмических средств компьютерной визуализации динамических данных,

<sup>1</sup>Центральная точка очага землетрясения

<sup>2</sup>Место, где океаническая кора погружается в мантию

что позволяет интерактивно отображать на экране модель глобуса с текстурами и рельефом высокого разрешения и различными эффектами с наложением геоданных и возможностью демонстрации их пространственно-временной структуры.

Растет интерес к новым подходам в визуализации геоданных. Так в 2008 году вышла монография, посвященная вопросам визуализации в географии: *Geographic Visualization: Concepts, Tools and Applications* [4]. В ней рассматриваются современные тренды в визуализации геопространственных данных, значение Google Earth для географии, польза 3D и стерео-3D технологий, использование многоэкранных проекционных установок для визуального анализа данных. В статье [5] рассматривается применение виртуальных глобусов в геологии. В журнале *Computers & Geosciences* вышел целый номер полностью посвященный виртуальным глобусам [6].

В 2011 году вышла первая монография, полностью посвященная разработке виртуальных глобусов: *3D Engine Design for Virtual Globes* [7]. В ней рассматриваются математические основы глобуса, визуализация векторных данных на поверхности глобуса, визуализация самого глобуса на основе спутниковых снимков и цифровых моделей рельефа, рассматриваются проблемы при визуализации глобусов и способы их решения.

Самое популярное приложение из разряда виртуальных глобусов, Google Earth, не предназначено для просмотра данных под поверхностью Земли, и не поддерживает стереоскопический режим отображения. Также эти функции не поддерживаются большинством современных виртуальных глобусов. Усилия разработчиков в этой области сейчас направлены главным образом на усовершенствование визуализации данных на поверхности Земли, повышение реалистичности, развитие эффектов атмосферы, воды, растительности и т. п.

Некоторые специализированные приложения поддерживают показ проекций подземных данных на поверхность Земли (например, эпицентры землетрясений). Есть специальное приложение для визуализации землетрясений на глобусе — *Earthquake3D* [8]. Оно поддерживает автоматическую загрузку оперативного каталога землетрясений с сайта USGS. Но оно тоже показывает землетрясения на поверхности глобуса. Для объемного рендеринга данных (в частности подземных данных) существует множество различных приложений, но все они работают без привязки к глобусу.

Трехмерная визуализация данных под поверхностью Земли на виртуальных глобусах, тем более в стерео-режиме, незаслуженно обойдена вниманием. Авторам известно только одно приложение для работы с подземными данными на виртуальном глобусе - *GA World Wind Suite* [9], разработанное в Австралии на основе *NASA World Wind Java SDK*.

В статье описано применение виртуального глобуса для показа сейсмических данных под поверхностью Земли. Предложен метод рендеринга и фильтрации массивов точечных данных на GPU, методы рендеринга объемных данных с учетом кривизны Земли, методы корректного рендеринга

полупрозрачного рельефа согласованно с другими объектами, методы визуализации и анимации различных подземных процессов. Разработано приложение для 3D-визуализации сейсмических данных с возможностью гибкой настройки режимов отображения глобуса. Поддержано отображение в различных стерео-режимах (горизонтальная стереопара, квадробуфер).

Приложение может использоваться для демонстрации массивов сейсмических данных, их визуального анализа, в том числе комплексного анализа различных наборов данных в едином контексте.

## 2. МЕТОДЫ ВИЗУАЛИЗАЦИИ И ПРИМЕРЫ ДАННЫХ

### 2.1 Построение виртуального глобуса

Существуют данные дистанционного зондирования Земли свободно доступные в интернете. Сюда можно отнести спутниковые снимки Landsat-7 с разрешением 15м в нескольких спектральных диапазонах, цифровую модель рельефа SRTM 90м и очищенную и обработанную текстуру Земли Blue Marble Next Generation с разрешением 500м.

Разрабатывается открытое программное обеспечение для построения глобуса по данным ДЗЗ. В 2008 году началась разработка библиотеки osgEarth на языке C++ на базе популярного графического инструментария OpenSceneGraph, которая активно развивается до сих пор. На основе этих инструментов возможна разработка своего глобуса для специализированных задач. В нашем проекте использовалась именно библиотека osgEarth.

Существует множество методов рендеринга рельефа, но в случае глобуса рекомендуется использовать метод CLOD (Chunked LOD). Он описан в книге [7] и реализован в osgEarth. Этот метод не требует предобработки сырых геопривязанных изображений, удобно ложится на эллипсоид, обеспечивает приемлемую точность в рамках типа float, и работает даже на старом аппаратном обеспечении начала 2000х годов. Метод заключается в использовании квадропада из тайлов, выстроенных вдоль меридианов и параллелей. Тайл — элемент поверхности, который имеет свою геометрию и текстуру и относится к некоторому уровню детализации. При приближении к тайлу он заменяется на 4 тайла большей детализации. На стыках между тайлами разных уровней детализаций могут возникать щели. Это решается добавлением по периметру каждого тайла вертикальной полоски треугольников, которая называется "юбка". При этом тайлы могут рендериться в произвольном порядке.

### 2.2 Полупрозрачный рельеф

Для просмотра данных под землей нужно сделать рельеф полупрозрачным. При этом важным являются порядок и режимы отрисовки разных слоев объектов в приложении. В случае глобуса это — рельеф, объекты на рельефе, атмосфера и звезды. В обычной ситуации порядок отрисовки следующий: звезды с отключенной записью в буфер глубины, потом атмосфера также с отключенной записью в буфер глубины, потом рельеф, потом все остальные надземные объекты.

При полупрозрачном рельефе начинают просвечивать звезды и атмосфера. Также видны объекты с противоположной стороны планеты, что вносит путаницу. Поэтому было предложено изменить порядок отрисовки. Во-первых, рисуется черный непрозрачный эллипсоид с центром в начале мировой системы координат, который меньше земного на 1000 км, поскольку максимальная глубина используемых на данный момент в приложении сейсмических данных составляет 700 км. Черный эллипсоид позволяет впоследствии отбросить объекты с дальней стороны Земли на этапе теста глубин.

Далее рисуются подземные объекты, далее рельеф, звезды и атмосфера.

При использовании эффекта прозрачности при визуализации рельефа возникают побочные эффекты. Картинка на экране становится чувствительной к порядку отрисовки тайлов и к повороту камеры относительно тайла: при определенных углах поворота камеры сквозь гористые участки рельефа начинают просвечивать другие части рельефа. Начинают просвечивать и мешаться юбки.

Для устранения этих эффектов было решено юбки отключить (для полупрозрачной поверхности проблема щелей не стоит так остро), а рельеф рендерить в 2 прохода. В первый проход происходит запись в буфер глубины при отключенной записи в цветовой буфер. Во второй проход происходит запись в буфер цвета только тех фрагментов, глубина которых совпадает с глубиной в буфере глубины. Это позволяет выводить на экран только ближайшие к камере полигоны рельефа.

При смешивании цвета рельефа с цветом подземных объектов в цветовом буфере предлагается использовать следующую функцию:

$$C = C_u + C_t * A_t,$$

где  $C$  — итоговый цвет фрагмента,  $C_u$  — цвет фрагмента в буфере глубины от подземного объекта,  $C_t$  — цвет фрагмента рельефа,  $A_t$  — прозрачность поверхности рельефа, которую можно регулировать. В этом случае цвет рельефа не вносит сильного искажения в цвет подземных объектов.

### 2.3 Режимы отображения глобуса

Для более наглядного представления данных реализованы различные режимы отображения глобуса:

- С атмосферой/без атмосферы
- С текстурами/без текстур
- С растяжением по высоте/без растяжения

Режимы могут произвольно комбинироваться. Также настраивается прозрачность поверхности Земли.

Для атмосферы использовалась модель О'Нила [10]. На данный момент она не самая реалистичная, но и сильно не загружает видеокарту, ресурсы которой более нужны для сейсмических данных.

Растяжение по высоте производится в вершинном шейдере. Вычисление прямо в шейдере высоты вершины над эллипсоидом на основе ее декартовых координат — это довольно медленная операция и может приводить к неточности из-за использования типа float. Поэтому в шейдер просто передается дополнительный вершинный атрибут с высотой.

### 2.4 Гипоцентры

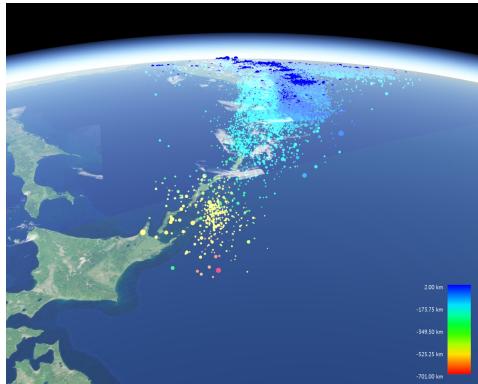
Одним из важнейших типов сейсмических данных являются каталоги гипоцентров землетрясений. Гипоцентр — это центральная точка очага землетрясения. Проекция гипоцентра на поверхность Земли называется эпицентром. Каталоги гипоцентров свободно доступны на специализированных сайтах, например, на сайте Геологической службы США (USGS) [11], на сайте Камчатского филиала Геофизической службы РАН [12], и др. На сайте USGS информация о новых землетрясениях выкладывается через несколько минут для Калифорнии и через полчаса для всего мира. Данные о гипоцентрах с сайта USGS можно получить в разных форматах: Atom, CSV, GeoJSON(P) и KML. При загрузке данных с сайта можно выбрать либо глобальный каталог, либо каталог землетрясений для заданной области. Полный каталог землетрясений для всей Земли с 1900 года включает около 1 млн. событий.

Строка каталога с описанием гипоцентра содержит следующие поля данных: широта, долгота, глубина, дата, время (час, мин, сек), магнитуда. При визуализации каталогов землетрясений возникает задача фильтрации по разным параметрам: по магнитуде, по дате, по региону [13].

В традиционном подходе данные фильтруются на центральном процессоре методом перебора либо с использованием специальных структур данных. Но для визуального анализа нет необходимости хранить результаты фильтрации. Фильтрацию можно осуществлять на графическом процессоре на этапе рендеринга. При нынешних мощностях видеокарт отрисовка и выполнение некоторых действий даже над миллионом вершин происходит очень быстро при условии, что это происходит за один вызов функции отрисовки. Это позволяет значительно упростить код для хранения гипоцентров и их фильтрации. Благодаря параллелизму фильтрация на графическом процессоре выполняется быстрее, чем на центральном процессоре.

Координаты гипоцентров конвертируются из географической системы координат в декартову систему координат виртуального глобуса и помещаются в массив вершин с набором дополнительных вершинных атрибутов: глубина, магнитуда, дата события, номер региона. Через uniform-переменные в шейдер передаются максимальные и минимальные значения для глубины, магнитуды, даты.

В вершинном шейдере производится только преобразование систем координат, а во фрагментном шейдере — выявление переменной `gl_FragColor`. Основные расчеты производятся в геометрическом шейдере. Вычисляется размер точки, ее цвет, и принимается решение отбрасывать точку точку или нет (рис. 1, 2).

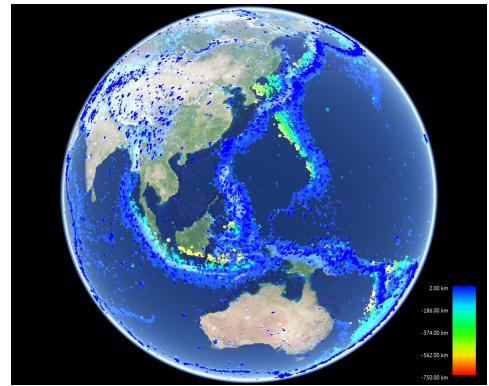


**Figure 1:** Региональный каталог сейсмических событий в районе Курило-Камчатской дуги, по данным КФ ГС РАН [12]

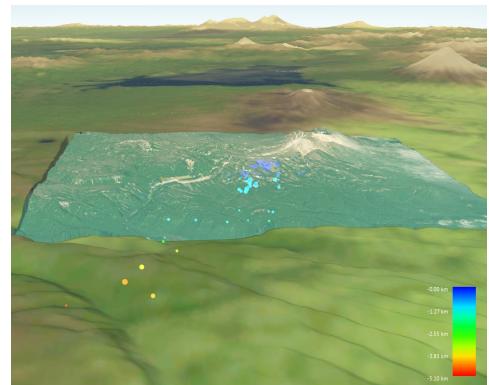
Глубина задается своим реальным значением (от 5км до -700км) с типом `float` и используется для вычисления цвета гипоцентра. Цветовое кодирование позволяет лучше определить глубину гипоцентра при обычном (нестереоскопическом) просмотре, а также является традиционным обозначением в геофизике. Цвет берется из линейной палитры от синего на малых глубинах до красного на больших глубинах.

Магнитуда также задается реальным значением от 0 (если не определена) до 9.5 с типом `float`. Если магнитуда землетрясения меньше минимального заданного значения или больше максимального, то точка отбрасывается. Значения магнитуды от минимального до максимального отображаются на диапазон размеров точек от 1го до 6ти пикселей.

Фильтрация по датам происходит более сложно. Для хранения дат был использован тип `boost::ptime` из библиотеки `boost`. Для передачи в шейдер дата приводится к диапазону



**Figure 2:** Мировой каталог сейсмических событий USGS за 1900-2011 гг, по данным USGS [14], в редакции А. В. Ланьера (МИТП РАН)



**Figure 3:** Локальная сейсмичность в районе Долины гейзеров на Камчатке в 2008-2009 гг, по данным Ю. А. Кугаенко и др., КФ ГС РАН [15]

[0; 1], где 0 — минимальная дата среди всех гипоцентров, а 1 — максимальная дата. Для вычисления значения даты для конкретного гипоцентра используется формула: (разница между датой гипоцентра и минимальной датой в секундах) / (разница между максимальной и минимальной датой в секундах). Максимальная и минимальная даты фиксированы для каждого каталога. Для фильтрации гипоцентров по дате задается минимальное и максимальное значение фильтра в диапазоне [0; 1] и передается в шейдер в виде uniform-переменных. Далее в геометрическом шейдере производится проверка значения даты на соответствие заданному диапазону.

Однако у этого подхода есть два недостатка. Во-первых, из-за особенностей реализации типа `boost::ptime` разница дат не может превышать примерно 60 лет. Для каталогов землетрясений с диапазоном дат в 100 лет приходится применять обходной маневр. Каталог разбивается на 2 части по датам. Первая половина отображается на интервал [0; 0.5], а вторая — на [0.5; 1]. Во-вторых, тип `float`, который используется для хранения даты, имеет точность — 7 десятичных разрядов. При интервале дат в 100 лет это означает, что шейдер не различает гипоцентры в пределах 315 секунд.

Также происходит фильтрация гипоцентров по регионам. Для каждого гипоцентра хранится идентификатор региона, к которому относится гипоцентр. При выборе алгоритма фильтрации учитывались следующие условия: (1) регионы не пересекаются и (2) одновременно можно выбирать несколько регионов.

Для реализации фильтрации возможны 2 подхода. Первый — использование битовых флагов. В этом походе идентификатор региона является степенью двойки, т.е. каждому региону соответствует свой бит. Идентификаторы всех выбранных регионов складываются (то же самое, что и операция "побитовое или"). Полученное число передается в виде uniform-переменной. В шейдере применяется операция "побитовое и" для принятия решения: оставить или отбросить вершину. Преимущество этого подхода — большая скорость работы, недостаток — ограничение на число регионов.

Второй подход — использование uniform buffer. В этом подходе номера выбранных регионов передаются в виде массива в шейдер. Далее для каждой вершины в цикле проверяется, что ее идентификатор присутствует в массиве. Преимущество этого метода — поддержка большого числа регионов, минус — меньшая производительность.

Нами тестирулся набор из 10 регионов для Курило-Камчатской дуги и поэтому был использован метод на основе битовых флагов.

Приложение позволяет загружать из файлов несколько каталогов одновременно. Каталоги хранятся отдельно в разных массивах вершин. Это сделано, чтобы можно было легко включать и отключать отдельные каталоги. Для каждого каталога отдельно хранятся диапазоны глубин и дат. И отдельно хранятся глобальные диапазоны для всех каталогов и других видов сейсмических данных. Эти глобальные значения передаются в шейдеры, чтобы все данные отображались согласованно.

При показе данных локальной сейсмичности глобус представляет возможность визуализации локального рельефа с высокой детальностью (рис. 3).

## 2.5 Зоны субдукции

Еще один важный вид подземных данных — это форма зон субдукции (рис. 4). Зоны субдукции — места, где литосферные плиты погружаются в мантию. Это места концентрации землетрясений. Форма зон субдукции выложена на сайте USGS. Данные представлены в виде двумерной матрицы глубин с шагом 0.2 градуса по долготе и широте. Пустые поля соответствуют отсутствию зоны субдукции в данных точках. По набору точек можно легко восстановить трехмерную поверхность из треугольников. Для каждой вершины задается атрибут — глубина по аналогии с гипоцентрами. Рендеринг производится в шейдере, где на основе глубины вычисляется цвет. В шейдер также передаются глобальные значения максимальной и минимальной глубины в виде uniform-переменных, чтобы цвет гипоцентров и зон субдукции был одинаковым на одинаковых глубинах.

Для подчеркивания сложной формы зон субдукций используется затенение с помощью источника света, привязанного к камере. Поскольку расчет освещения происходит в шейдере в видовой системе координат, то вектор направления на источник света берется просто  $(0, 0, -1)$ .

## 2.6 Сейсмическая регионализация

Землетрясения, происходящие в районе Курило-Камчатской дуги, имеют различную тектоническую природу. Разнообразие тектонических обстановок требует строгого разделения существующих и непрерывно развивающихся каталогов землетрясений с возможно более точной идентификацией природы каждого сейсмического события. С этой целью на основе накопленной на сегодняшний день информации о пространственном распределении и механизмах землетрясений и с учетом современных представлений о новейшей тектонике региона проведена регионализация сейсмоактивного объема Камчатки и прилегающих областей.

Согласно представлениям А. В. Ландера (МИТП РАН), каж-

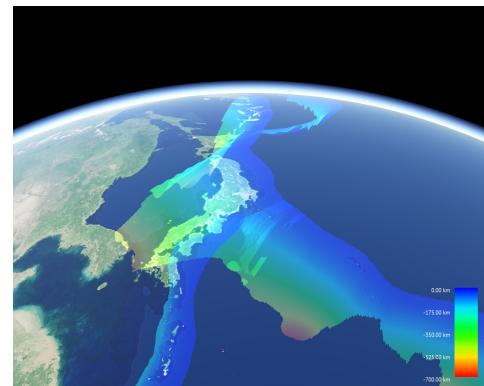


Figure 4: Форма зоны субдукции в районе Японии, по данным USGS [16]

дый регион задается призмами сложной формы. Каждая призма имеет 2 крышки (верхнюю и нижнюю) в географических координатах. Такие призмы называются базовыми фигурами. Реальные регионы являются результатом булевских операций над базовыми фигурами. Фигуры можно объединять, вычитать и брать их пересечение.

Помимо задачи определения принадлежности гипоцентра конкретному региону встает задача визуализации границ самого региона. Если показать базовую фигуру — достаточно простая задача, то логические операции представляют определенные сложности.

Существует два основных подхода к булевским операциям: основанный на геометрии и основанный на изображении.

Подход, основанный на геометрии, генерирует полигональную 3D-модель результирующей фигуры после применения булевских операций. Преимуществом этого подхода является возможность сохранения полученной модели в файл для последующего использования. Минусы подхода: большая сложность и медленность алгоритмов. Возможно появление артефактов в сложных случаях. Большую роль играют ошибки округления.

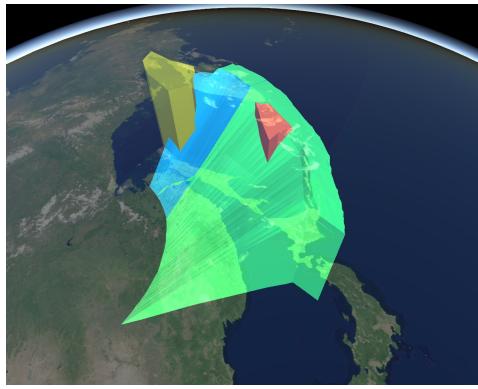
Подход, основанный на изображении, генерирует изображение результирующей фигуры на экране без использования отдельной геометрии. Рендерятся только базовые фигуры в несколько проходов с разными параметрами отрисовки (используется буфер трафарета). Преимуществом данного подхода является большая простота (хотя для вогнутых фигур алгоритм существенно усложняется). Также появляется возможность динамического изменения базовых фигур. Это особенно актуально в случае глобуса, т.к. необходимо изгибать крышки призм в соответствии с поверхностью глобуса. При приближении к поверхности можно производить динамическую тесселяцию крышек.

Для реализации булевских операций с вогнутыми фигурами нами был модифицирован алгоритм Голдфезера с применением шейдеров, основанный на изображении [17]. В этом алгоритме каждый регион сначала рендерится в текстуру с помощью фреймбуфера, а потом результат переносится в основной буфер (рис. 5).

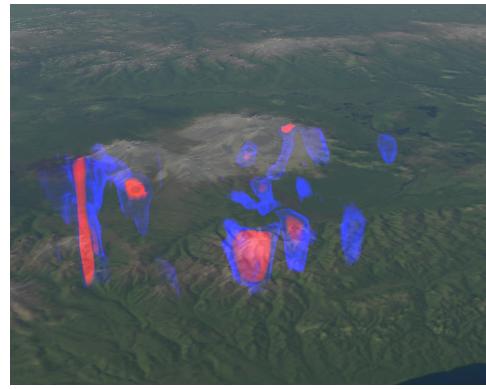
## 2.7 Данные сейсмомографии

В настоящее время активно развивается направление сейсмомографии [18][19]. По аналогии с медицинской томографией сейсмомография позволяет получить распределение некоторых пространственных характеристик земной коры путем анализа данных с сети сейсмостанций, регистрирующих сейсмические волны от землетрясений.

Землетрясения создают продольные и поперечные волны,



**Figure 5:** Границы сейсмических регионов в районе Курило-Камчатской дуги, по данным А. В. Ландера (МИТП РАН)



**Figure 6:** Визуализация результатов сейсмической томографии в районе Ключевской группы вулканов на Камчатке за 2004 год, по данным И. Ю. Кулакова (ИНГ СО РАН)

которые обозначают буквами Р и S соответственно. Скорости распространения сейсмических волн различны и зависят от модуля упругости и плотности среды. Это позволяет идентифицировать некоторые характеристики земной коры на пути следования сейсмических волн. Результатом обработки множества данных является трехмерная матрица (воксельный массив), в ячейках которой записано значение отношения скоростей Р-волн и S-волн в данной точке. Каждая ячейка матрицы (воксель) имеет координаты: долгота, широта, глубина.

Для визуализации этих данных был использован прямой объемный рендеринг [20]. Для хранения объемных данных на GPU используется 3D-текстура. В ней хранится не цвет, а условная "плотность" — значение параметра  $Vp/Vs$ . Отдельно задается передаточная функция (transfer function), которая ставит в соответствие плотности цвет и прозрачность.

Для небольших объемов можно считать Землю локально плоской и выводить объем в виде куба. При этом был использован метод обратной трассировки лучей. Для каждого пикселя экрана пускается луч через объем. С некоторым шагом из объема выбираются значения плотности, вычисляется и суммируется цвет на основе передаточной функции.

Использованы различные модификации метода: показ изоповерхностей, проекции максимальной интенсивности. Для анализа взаимного расположения подземных структур с различными соотношениями  $Vp/Vs$  добавлена поддержка двух изоповерхностей.

Рендеринг тестировался на объемных данных под Ключевской группой вулканов (рис. 6). Размер матрицы 121x81x46, шаг по XY — 1 сек, по Z — 1 км. Объем расположен не полностью под поверхностью эллипсоида, но поднимается над ним на высоту 5 км. Большая часть надземного объема заполнена пустыми значениями. Меньшая часть содержится внутри вулкана. Реализовано корректное отображение рельефа поверх объема.

Для больших объемов уже необходимо учитывать кривизну Земли. Для этого метод обратной трассировки лучей подходит хуже. Поэтому был использован метод слоев. Каждый слой является частью эллипсоида, параллельного земному эллипсоиду.

В ходе сейсмических и вулканических процессов характеристики и структура земной коры меняются. Эти изменения можно наблюдать, сравнивая данные сейсмомагнитографии за разные промежутки времени. Особенно явно видны изменения в области активной вулканической деятельности. Нами поддержана плавная анимация объемной картинки во времени.

## 2.8 Геологические модели и анимация извержений

Комплексная визуализация сейсмических данных, связанных с вулканической активностью, может быть дополнена показом геологических моделей (границы тектонических плит, структуры очага и каналов) и анимацией извержений. Движение магмы к земной поверхности, сопровождается ролями землетрясений, изменениями скоростей распространения сейсмических волн под вулканическим комплексом. Стартованный показ всех этих данных дает возможность комплексного анализа динамики процесса.

Подземная часть системы состоит из канала, по которому поступает магма. Канал задается полигональной моделью. Поскольку реальная форма канала не известна, то можно сгенерировать произвольный канал путем фрактальной деформации конуса или цилиндра. Показ движения магмы в канале реализован на уровне шейдеров. В вершинном шейдере срезается часть канала, чтобы было видно сечение, и рассчитываются нужные цвета сечения, исходя из параметров, которые передаются в виде uniform-переменных. Надземная часть извержения визуализируется в виде системы частиц.

## 3. ВЫВОДЫ

Разработано приложение для визуализации различных видов сейсмических данных и сейсмических явлений на базе виртуального глобуса. Приложение использует свободную библиотеку osgEarth для генерации и рендеринга самого глобуса на основе геопривязанных изображений. Были внесены изменения в код osgEarth для поддержки прозрачного рельефа и переключения различных режимов отображения глобуса во время работы приложения (атмосфера, растяжение по высоте, отключение текстур).

Протестированы региональные и мировые каталоги землетрясений, содержащие до 1 миллиона событий. Разработаны средства поддержки для показа зон субдукции, выделения объемных зон землетрясений с типичными тектоническими характеристиками, границ тектонических плит. Реализована визуализация данных сейсмомагнитографии с тестированием на примере района Ключевской группы вулканов с помощью прямого объемного рендеринга. Реализована визуализация геологических моделей и анимации. Поддержана стереоскопическая визуализация.

Отображение разных видов сейсмических данных в едином контексте на базе глобуса позволяет анализировать зависимости между данными. При этом оказалось очень важным использование стереоскопического просмотра. Это позволяет

ет визуально отделять группы гипоцентров между собой и подземные объекты от объектов, расположенных на поверхности Земли.

Разработанное приложение используется для демонстрации результатов геофизических исследований на постоянно действующей выставке РАН. Планируется дальнейшее развитие приложения путем добавления новых видов данных и инструментов анализа данных.

#### 4. БЛАГОДАРНОСТИ

Авторы выражают благодарность: Кугаенко Ю.А. (КФ ГС РАН), Кулакову И.Ю. (ИНГГ СО РАН), Ландеру А.В. (МИТП РАН) за предоставленные исходные данные и научные консультации; Батурину Ю.М. (ИИЕТ РАН), Гордееву Е.И. (ИВиС ДВО РАН), Клименко С.В. (ИФТИ), Маловичко А.А. (ГС РАН) за поддержку.

#### 5. СПИСОК ЛИТЕРАТУРЫ

- [1] Гордеев Е.И., Чебров В.Н., Левина В.И., Сенюков С.Л., Шевченко Ю.В., and Ящук В.В., “Система сейсмологических наблюдений на Камчатке,” *Вулканология и сейсмология*, vol. 3, pp. 6–27, 2006.
- [2] Гордеев Е.И., Чебров В.Н., Левина В.И., Бахтиярова Г.М., Сенюков С.Л., and Пантиухин Е.А., “Банк сейсмологических данных Камчатки,” *Открытое образование*, vol. 4, pp. 16–22, 2008.
- [3] Michael F. Goodchild, “What does google earth mean for the social sciences?,” in *Geographic Visualization: Concepts, Tools and Applications*. Wiley, 2008.
- [4] Martin Turner Martin Dodge, Mary McDerby, *Geographic Visualization: Concepts, Tools and Applications*, Wiley, 2008.
- [5] T. Paraskevas, “Virtual Globes and Geological Modeling,” *International Journal of Geosciences*, vol. 2, no. 4, pp. 648–656, 2011.
- [6] Aijun Chen and John Bailey, Eds., *Virtual Globes in Science*, vol. 37, 2011.
- [7] Patrick Cozzi and Kevin Ring, *3D Engine Design for Virtual Globes*, CRS Press, 2011.
- [8] Earthquake 3D, ,” <http://www.wolton.net/quake.html>.
- [9] GA World Wind Suite, ,” <https://github.com/ga-m3dv/ga-worldwind-suite>.
- [10] S. O’Neal, “Accurate atmospheric scattering,” *GPU Gems*, vol. 2, pp. 253–268, 2005.
- [11] USGS, ,” <http://usgs.gov>.
- [12] КФ ГС РАН, ,” <http://www.emsd.ru/>.
- [13] Белосохов Д.Е., Бобков А.Е., and Леонов А.В., “Возможности 3D-визуализации для эффективного представления результатов научных исследований,” in *Проблемы комплексного геофизического мониторинга Дальнего Востока России. Труды Третьей научно-технической конференции. Петропавловск-Камчатский. 9-15 октября 2011 г.*, Чебров В.Н., Ed., 2011, pp. 347–351.
- [14] Каталог землетрясений USGS, ,” <http://earthquake.usgs.gov/earthquakes/feed/>.
- [15] Кугаенко Ю.А., Салтыков В.А., and Коновалова А.А., “Локальная сейсмичность района Долины Гейзеров по данным полевых наблюдений 2008-2009 гг.,” *Вестник КРАУНЦ. Науки о земле*, vol. 1, 2010.
- [16] Зоны субдукции USGS, ,” <http://earthquake.usgs.gov/research/data/slab/>.
- [17] Jack Goldfeather, S. Monar, Greg Turk, and Henry Fuchs, “Near real-time CSG rendering using tree normalization and geometric pruning,” *IEEE Computer Graphics and Applications*, vol. 9, pp. 20–28, 1989.
- [18] Кулаков И.Ю., “Геодинамические процессы в коре и верхней мантии земли по результатам региональной и локальной сейсмомагнитографии,” ИГМ СО РАН, 2007, Диссертация на соискание ученой степени доктора геолого-минералогических наук.
- [19] I. Koulakov, E.I. Gordeev, N.L. Dobretsov, V.A. Vernikovsky, S. Senyukov, and A. Jakovlev, “Feeding paths of the Kluchevskoy volcano group (Kamchatka) from the results of local earthquake tomography,” *Geophys. Res. Lett.*, vol. 38, 2011.
- [20] Klaus Engel, Markus Hadwiger, Joe M. Kniss, Aaron E. Lefohn, Christof Rezk Salama, and Daniel Weiskopf, “Real-time volume graphics,” 2004.

#### ОБ АВТОРАХ

Александр Бобков — аспирант Московского физико-технического института, его адрес: [alexander.e.bobkov@gmail.com](mailto:alexander.e.bobkov@gmail.com)

Андрей Леонов — к. ф.-м. н., руководитель Центра виртуальной истории науки и техники Института истории естествознания и техники им. С. И. Вавилова РАН, его адрес: [a.leonov@ihst.ru](mailto:a.leonov@ihst.ru)

Виктор Чебров — к. т. н., директор Камчатского филиала Геофизической службы РАН, г. Петропавловск-Камчатский, его адрес: [chebr@emsd.ru](mailto:chebr@emsd.ru)

# Высококачественная объемная визуализации в реальном времени

Денис Боголепов<sup>1</sup>, Илья Бугаев<sup>1</sup>, Дмитрий Сопин<sup>1</sup>, Данила Ульянов<sup>2</sup>, Вадим Турлапов<sup>1</sup>

<sup>1</sup>Нижегородский государственный университет им. Н.И. Лобачевского

<sup>2</sup>Нижегородский государственный технический университет им. Р.Е. Алексеева

denisbogol@gmail.com, sopindm@gmail.com, danila-ulyanov@ya.ru

## Аннотация

В настоящей работе рассматривается алгоритм прямой визуализации объема, построенный на методе испускания лучей и адаптированный для программируемой графической аппаратуры. Алгоритм использует прединтегрированную классификацию, которая обеспечивает высокое качество визуализации при относительно небольших вычислительных затратах. Для работы алгоритма практически не требуется предварительных вычислений, что позволяет его встраивать в системы, допускающие модификацию передаточных функций в реальном времени. Предлагаются оптимизационные стратегии для повышения качества и скорости работы. Приводятся оценки производительности.

**Ключевые слова:** *direct volume rendering, ray casting, pre-integrated classification, GPU, GPGPU, OpenGL, GLSL.*

## 1. ВВЕДЕНИЕ

Объемная визуализация – метод формирования изображения, который для трехмерного набора данных отображает не только общий вид, но и внутреннее строение трехмерного объекта. В качестве входного набора данных часто выступает множество плоских изображений слоев, полученных с помощью компьютерной или магнитно-резонансной томографии. Обычно слои имеют равную толщину и одинаковое число пикселей на каждый слой. Таким образом, входные данные можно представить регулярной пространственной сеткой, каждому узлу которой ставится в соответствие значение некоторого скалярного поля.

Существует большое разнообразие методов визуализации объема, однако наилучшие результаты обеспечивает алгоритм трассировки лучей, который естественным образом вычисляет интеграл объемного рендеринга. Значительное влияние на результирующее изображение также оказывает метод классификации, посредством которого конкретным значениям скалярного поля ставится в соответствие набор оптических свойств. Высокое качество обеспечивает так называемая прединтегрированная классификация, которая была предложена в работе [1]. Однако доступная на тот момент аппаратура не поддерживала программируемый графический конвейер, поэтому метод был реализован с помощью техники слоев [2]. С ростом возможностей программируемых графических процессоров появились интерактивные реализации прямого объемного рендеринга на основе испускания лучей [3]. В публикации [4] авторы представили программный каркас системы объемной визуализации с использованием прединтегрированной классификации и такими эффектами, как объемные тени и освещение. Исследование различных методов интегрирования посвящена работа [5], в которой авторы предложили использовать интерполяцию второго порядка для аппроксимации скалярного поля в пределах интервала дискретизации. Наряду с этим следует отметить работы, направленные на повышение наглядности визуализации и упрощение анализа изображения. В частности, рассматривалась визуа-

лизация в широком диапазоне (HDR), объемные тени, различные модели освещения [6]–[7].

Настоящая работа посвящена реализации высококачественного объемного рендеринга, в основе которого лежит метод испускания лучей. Рассматриваются особенности реализации прединтегрированной классификации на графическом процессоре и исследуются ее преимущества по сравнению с традиционной пост-классификацией. Предлагается ряд оптимизационных стратегий, направленных на улучшение качества изображения и повышение скорости работы. Приводится сравнительный анализ различных вариантов реализации метода.

## 2. ОСНОВНЫЕ МОДЕЛИ И МЕТОДЫ

### 2.1 Интеграл объемной визуализации

Предположим, что луч  $\mathbf{x}(t)$  параметризован расстоянием  $t$  от объектива виртуальной камеры, а излучаемая яркость  $\text{intensity}(\mathbf{x})$  и коэффициент затухания  $\text{extinction}(\mathbf{x})$  могут быть вычислены в любой точке  $\mathbf{x}$  пространства. Интеграл объемной визуализации [1] описывает процесс накопления яркости вдоль луча с учетом ее затухания в зависимости от расстояния до точки наблюдения:

$$I = \int_0^D \text{intensity}(\mathbf{x}(t)) \times \text{extinction}(\mathbf{x}(t)) \times \exp\left(-\int_0^t \text{extinction}(\mathbf{x}(t')) dt'\right) dt \quad (1)$$

Здесь через  $D$  обозначено расстояние, на которое луч проникает в объем (в большинстве случаев определяется временем выхода из ограничивающей оболочки).

### 2.2 Скалярное поле

На практике скалярное поле  $s(\mathbf{x})$  часто задается регулярной трехмерной сеткой, каждому узлу  $\mathbf{v}_i$  которой ставится в соответствие значение поля  $s_i$ . Для вычисления поля в произвольной точке пространства выполняется интерполяция. Порядок данной интерполяции существенно влияет на качество генерируемого изображения. Современные графические процессоры аппаратно поддерживают трилинейную фильтрацию. Лучшие результаты обеспечивает трикубическая фильтрация, которая может быть реализована через 8 операций трилинейной фильтрации [9].

### 2.3 Передаточные функции

Преобразование скалярного поля  $s(\mathbf{x})$  в яркость излучаемого света и коэффициент затухания называется классификацией. Данное преобразование выполняется путем задания передаточных функций для вычисления излучаемой яркости  $L(s)$  и коэффициента затухания  $\tau(s)$ . При наличии передаточных функций интеграл (1) записывается в виде:

$$I = \int_0^D L(s(\mathbf{x}(t))) \times \tau(s(\mathbf{x}(t))) \times \exp\left(-\int_0^t \tau(s(\mathbf{x}(t'))) dt'\right) dt \quad (2)$$

В ряде случаев для вычисления яркости  $L(s)$  в конкретной точке луча  $\mathbf{x}(t)$  могут использоваться алгоритмы затенения, основанные на различных моделях освещения (таких как модели Блинна-Фонга или Эми Гуч).

## 2.4 Пред- и пост-классификация

В зависимости от способа вычисления передаточных функций выделяют различные типы классификации. В случае пост-классификации передаточные функции применяются после интерполяции скалярного поля  $s(\mathbf{x})$ . В случае пред-классификации излучаемая яркость и коэффициент затухания вычисляются на этапе препроцессирования для каждого узла сетки  $\mathbf{v}_i$  и используются для интерполяции в произвольную точку пространства. На практике данный подход практически не применяется.

## 2.5 Численное интегрирование

В большинстве случаев для оценки интеграла (2) используется простой метод прямоугольников. Будем считать, что отрезок интегрирования разбивается на  $n$  элементарных отрезков равной длины  $d = D / n$ . Тогда справедливы следующие квадратурные формулы:

$$\begin{aligned} \exp\left(-\int_0^t \tau(s(\mathbf{x}(t'))) dt'\right) &\approx \exp\left(-\sum_{i=0}^{t/d} \tau(s(\mathbf{x}(id))) d\right) \\ &= \prod_{i=0}^{t/d} \exp(-\tau(s(\mathbf{x}(id))) d) = \prod_{i=0}^{t/d} (1 - \alpha_i) \end{aligned} \quad (3)$$

Здесь через  $\alpha_i$  обозначена непрозрачность  $i$ -го сегмента луча. Разложение экспоненты в ряд Тейлора позволяет дополнительно упростить выражение:

$$\alpha_i = 1 - \exp(-\tau(s(\mathbf{x}(id))) d) \approx \tau(s(\mathbf{x}(id))) d \quad (4)$$

Данная формула предпочтительна в задачах визуализации реального времени, поскольку вычисление экспоненциальной функции существенно снижает производительность. Аналогичную оценку можно получить для яркости, излучаемой  $i$ -ым сегментом луча:

$$C_i \approx L(s(\mathbf{x}(id))) \tau(s(\mathbf{x}(id))) d \quad (5)$$

В результате, получаем следующую аппроксимацию для интеграла (2):

$$I \approx \sum_{i=0}^n C_i \prod_{j=0}^{i-1} (1 - \alpha_j) \quad (6)$$

Полученная аппроксимация сходится при  $d \rightarrow 0$ . На практике для точной оценки непрерывного подынтегрального выражения необходимо рационально выбрать шаг дискретизации. Для понимания сути проблемы уместно обратиться к теореме отсчетов (Котельникова), согласно которой корректное восстановление аналогового сигнала возможно только по дискретным отсчетам, взятым с частотой строго большей частоты Найквиста (удвоенной верхней частоты сигнала). Скачкообразный характер передаточных функций приводит к значительному повышению частоты дискретизации, поскольку частота Найквиста у подынтегрального выражения в худшем случае оценивается произведением данных частот у поля  $s(\mathbf{x})$  и передаточных функций  $L(s)$  и  $\tau(s)$ . В итоге корректное вычисление интеграла может потребовать сотен тысяч интервалов дискретизации для каждого луча. Для работы алгоритмов визуализации в реальном времени число интервалов ограничивается несколькими тысячами, что приводит к различным артефактам в результате изображении.

## 2.6 Прединтегрированная классификация

Эффективным способом борьбы с высокими частотами Найквиста является прединтегрированная классификация, которая подробно рассматривается в работе [1]. Основная идея сводится к декомпозиции вычислений на интегрирование скалярного поля  $s(\mathbf{x})$  и интегрирование передаточных функций  $L(s)$  и  $\tau(s)$ . Для этого на этапе препроцессирования генерируются таблицы, с помощью которых можно получить яркость и непрозрачность произвольного сегмента луча во время визуализации. В общем случае для обращения к таблицам используется три аргумента: значение поля в начальной точке сегмента  $s_f = s(\mathbf{x}(id))$ , значение поля в конечной точке сегмента  $s_b = s(\mathbf{x}(id + d))$  и длина сегмента  $d$ . Если же лучи разбиваются на сегменты равной длины, то число аргументов сокращается до двух. Непрозрачность  $i$ -го сегмента выражается формулой:

$$\begin{aligned} \alpha_i(s_f, s_b, d) &= 1 - \exp\left(-\int_{id}^{id+d} \tau(s(\mathbf{x}(t'))) dt'\right) \\ &\approx 1 - \exp\left(-\int_0^1 \tau((1-\lambda)s_f + \lambda s_b) d d\lambda\right) \\ &\approx \int_0^1 \tau((1-\lambda)s_f + \lambda s_b) d d\lambda \end{aligned} \quad (7)$$

Аналогично оценивается яркость  $i$ -го сегмента луча:

$$\begin{aligned} C_i(s_f, s_b, d) &\approx \int_0^1 L((1-\lambda)s_f + \lambda s_b) \times \tau((1-\lambda)s_f + \lambda s_b) \\ &\quad \times \exp\left(-\int_0^\lambda \tau((1-\lambda')s_f + \lambda' s_b) d d\lambda'\right) d d\lambda \end{aligned} \quad (8)$$

Таким образом, прединтегрированная классификация позволяет не увеличивать частоту дискретизации для скачкообразных передаточных функций. В итоге имеется потенциал для улучшения качества визуализации и уменьшения числа интервалов дискретизации на этапе визуализации объема.

## 2.7 Упрощенное прединтегрирование

Основной недостаток прединтегрированной классификации связан с необходимостью вычисления таблиц, с помощью которых каждому набору параметров  $s_f$ ,  $s_b$  и  $d$  ставится в соответствие яркость сегмента и его непрозрачность. Указанные таблицы зависят от передаточных функций, поэтому требуют повторного построения при их модификации. Данное построение может быть эффективно выполнено на графическом процессоре, что позволит модифицировать передаточные функции в реальном времени.

Альтернативным вариантом является аппроксимированная прединтегрированная классификация [1], которая без видимого снижения качества позволяет ограничиться быстрым предварительным вычислением двух первообразных функций:

$$T(s) = \int_0^s \tau(\lambda) d\lambda \quad (9)$$

$$K(s) = \int_0^s L(\lambda) \tau(\lambda) d\lambda \quad (10)$$

С использованием данных функций формулы (7) и (8) могут быть преобразованы следующим образом:

$$\alpha_i(s_f, s_b, d) \approx \frac{d}{s_b - s_f} [T(s_b) - T(s_f)] \quad (11)$$

$$C_i(s_f, s_b, d) \approx \frac{d}{s_b - s_f} [K(s_b) - K(s_f)] \quad (12)$$

Выражение (12) предполагает отсутствие затухания в пределах одного сегмента луча, что оправданно при малом интервале дискретизации.

## 2.8 Стратегии вычисления цвета

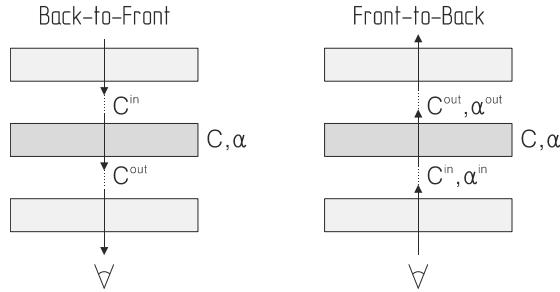


Рис. 1: Стратегии накопления цвета.

На основе аппроксимации (6) строится алгоритм *обратного* (*back-to-front*) вычисления цвета пикселя:

$$C^{out} \leftarrow C + (1 - \alpha)C^{in} \quad (13)$$

Алгоритм *прямого* (*front-to-back*) вычисления цвета также возможен, однако требует дополнительных ресурсов для поддержки коэффициента непрозрачности:

$$\begin{aligned} C^{out} &\leftarrow C^{in} + (1 - \alpha^{in})C \\ \alpha^{out} &\leftarrow \alpha^{in} + (1 - \alpha^{in})\alpha \end{aligned} \quad (14)$$

Несмотря на использование дополнительных регистров и большее число операций в цикле трассировки, алгоритм прямого вычисления цвета часто более предпочтителен за счет возможности ранней остановки луча (early ray termination). В настоящей работе реализованы обе стратегии трассировки.

## 2.9 Стратегии генерации лучей

В типовой реализации метода объемного рендеринга цвет каждого пикселя вычисляется с помощью одного первичного луча, для оценки яркости которого используется  $S$  интервалов дискретизации. В качестве альтернативного подхода можно предложить генерацию  $R$  лучей на пиксель, начальные точки которых последовательно смещаются в пределах интервала дискретизации, а число интервалов сокращается до  $S / R$ . Очевидно, что вычислительные затраты в обоих случаях одинаковы, однако второй подход имеет ряд преимуществ.

Во-первых, программный код имеет большую степень параллелизма, поскольку лучи в пределах пикселя обрабатываются независимо. Шейдерный компилятор получает возможность эффективного распараллеливания инструкций, что ведет к более полному использованию ресурсов ГПУ.

Во-вторых, данный подход позволяет реализовать практически бесплатное полноэкранное сглаживание (FSAA). Для этого каждый из  $R$  лучей необходимо дополнительно смещать в пределах пикселя экранной плоскости. При этом эффективно устраняется не только эффект ступенчатости изображения, но и мелкие (сопоставимые с размерами пикселя) артефакты визуализации.

В-третьих, лучи в пределах пикселя характеризуются высокой пространственной когерентностью, что позволяет дополнительно повысить быстродействие. Трассировка первого луча через центр пикселя в качестве побочного результата дает оценку расстояния, которое луч преодолел в пустом пространстве. Данная информация позволяет трассировать остальные лучи сразу со значимой области исследуемого объема. В ряде случаев такая оптимизация увеличивает скорость работы в несколько раз, поскольку значительно сокращается число наиболее трудоемких операций – выборок из трехмерной текстуры скалярного поля (оценки производительности приводятся далее).

## 3. РЕАЛИЗАЦИЯ АЛГОРИТМА НА ГПУ

Несмотря на широкую доступность таких универсальных инструментов программирования ГПУ как NVIDIA CUDA и OpenCL, в настоящей работе для этой цели использовался интерфейс OpenGL и его шейдерный язык GLSL (OpenGL Shading Language). Богатый набор встроенных типов данных и широкие функциональные возможности позволяют в естественном виде записывать алгоритм испускания лучей. К достоинствам подхода следует также отнести поддержку графических ускорителей от различных производителей. Интерфейс OpenGL и его язык шейдеров являются межплатформенным стандартом, что позволяет использовать единый программный код для поддержки самых различных платформ.

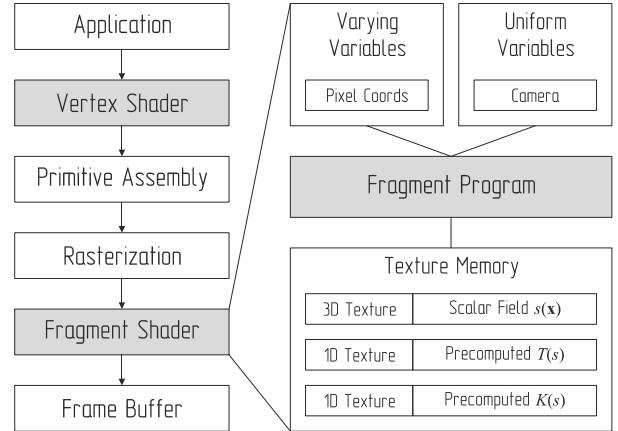


Рис. 2: Схема работы алгоритма визуализации.

Существует стандартный подход к отображению алгоритма испускания лучей на программируемый графический конвейер. Для инициирования вычислений следует установить параллельную проекцию и нарисовать прямоугольник, заполняющий всю область видимости в окне размера  $N \times M$  пикселей. На этапе растеризации данного прямоугольника будет сгенерировано ровно  $N \times M$  фрагментов, соответствующих пикселям в буфере кадра. Каждый сгенерированный фрагмент обрабатывается фрагментным шейдером, на вход которому передаются координаты пикселя и данные для визуализации. Располагая всей необходимой информацией, фрагментный шейдер генерирует первичный луч (несколько лучей) и трассирует его через объем согласно изложенному выше алгоритму. Шейдерный язык GLSL поддерживает директивы препроцессора, с помощью которых можно быстро переключаться между блоками кода, реализующими различные методы визуализации. В тестовой версии программы пользователь может выбирать между пост-классификацией и прединтегрированной классификацией, а также между прямым и обратным алгоритмом прохода по лучу.

## 4. АНАЛИЗ РЕЗУЛЬТАТОВ

Для оценки качества и скорости работы рассмотренных алгоритмов использовались широко известные данные для объемной визуализации [10]. Расчеты производились на графическом процессоре начального уровня NVIDIA GeForce 560 1 Гб под управлением ОС Ubuntu Linux 10.10 и видеодрайвера NVIDIA Linux Graphics Driver 275.36. Визуализация выполнялась в режимах пост-классификации и прединтегрированной классификации, при этом сравнивались прямой и обратный алгоритм вычисления цвета. Для обработки каждого пикселя генерировалось 5 лучей, каждый из которых разбивался не более чем на 120 интервалов. Таким образом, общее число дискретных отсчетов на пиксель не превышало 600, что является скромным показателем для современной графической аппаратуры. Для каждого режима визуализации выполнялось два замера: без использования и с использованием когерентности субпиксельных лучей. В следующей таблице представлены характеристики входных данных и результаты эксперимента.

**Таблица 1:** Результаты замера производительности (кадров / секунду, окно  $512 \times 512$  пикселей).

Description	Dimensions	Post-classification		Pre-integrated classification	
		Bits	Front-to-Back	Back-to-Front	Front-to-Back
Bucky Ball	$32 \times 32 \times 32$ 8 bit	120 / 142	82 / 119	150 / 152	75 / 140
Daisy Pollen Grain	$192 \times 180 \times 168$ 8 bit	59 / 100	58 / 98	66 / 108	66 / 112
Engine Block	$256 \times 256 \times 256$ 8 bit	71 / 118	67 / 130	74 / 110	71 / 130
Bonsai Tree	$512 \times 512 \times 154$ 8 bit	61 / 128	63 / 89	66 / 114	51 / 90
Stanford Bunny	$512 \times 512 \times 361$ 16 bit	36 / 92	38 / 79	38 / 47	34 / 44
Orange	$256 \times 256 \times 64$ 8 bit	61 / 86	75 / 98	84 / 100	60 / 120

Экспериментальные данные показывают, что прединтегрированная классификация практически не вносит дополнительных затрат на этапе визуализации по сравнению с пост-классификацией. В данном случае скорость работы ограничивается производительностью текстурных модулей, которые интенсивно используются для выборки и фильтрации значений скалярного поля. Во всех режимах работы число данных операций остается неизменным.

Прединтегрированная классификация (рис. 3) обеспечивает значительно более высокое качество изображения при использовании всего 600 интервалов дискретизации на каждый пиксель. Таким образом, в особо трудных случаях (при работе с высокочастотными передаточными функциями и скалярным полем) остается значительный ресурс повышения качества за счет увеличения числа интервалов (вплоть до нескольких раз при сохранении реального времени).

Без использования когерентности субпиксельных лучей наилучшие результаты дает метод прямого накопления цвета с прединтегрированной классификацией. Очевидно, что данный режим работы обеспечивает наилучшие условия для ранней остановки луча. Использование когерентности для отсечения пустого пространства позволяет значи-

тельно повысить производительность. Величина ускорения существенно зависит от входных данных и передаточных функций, однако в ряде случаев наблюдается прирост до двух и более раз. Кроме того, использование когерентности зачастую выводит метод обратного накопления цвета в число наиболее эффективных. Поскольку выбрать единственный оптимальный подход не представляется возможным, в прикладных программах целесообразно поддерживать различные режимы визуализации.

## 5. ЗАКЛЮЧЕНИЕ

В настоящей работе был реализован алгоритм прямой объемной визуализации для программируемой графической аппаратуры на базе метода испускания лучей. Алгоритм использует прединтегрированную классификацию, которая обеспечивает высокую скорость работы и качество визуализации даже для нелинейных передаточных функций с высокими частотами. Для работы алгоритма практически не требуется предварительных вычислений, что позволяет модифицировать передаточные функции в реальном времени. Предложены подходы экономичного полноэкранного сглаживания и отсечения пустого пространства от начальной точки луча до значимой области объема. Реализована тестовая версия алгоритма на базе интерфейса OpenGL и шейдерного языка GLSL, которые обеспечивают высокую производительность на различных программных и аппаратных платформах.

## 6. ЛИТЕРАТУРА

- [1] K. Engel, M. Kraus, T. Ertl. *High-quality pre-integrated volume rendering using hardware-accelerated pixel shading* // In Proc. ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware (HWWS '01), 9-16.
- [2] J. Edward Swan, Roni Yagel. *Slice-Based Volume Rendering* // Ohio State University Technical Report OSU-ACCAD-1/93-TR1, January 1993.
- [3] J. Krüger, R. Westermann. *Acceleration Techniques for GPU-based Volume Rendering* // In Proc. 14th IEEE Visualization 2003 (VIS'03), 287-292.
- [4] J. E. Vollrath, D. Weiskopf, and T. Ertl. *A generic software framework for the gpu volume rendering pipeline* // In Proc. Vision, Modeling, and Visualization 2005, 391-398.
- [5] J.-F. El Hajjar, S. Marchesin, J.-M. Dischler, C. Mongenet. *Second Order Pre-Integrated Volume Rendering* // In Proc. Visualization Symposium 2008 (PacificVIS '08), 9-16.
- [6] A. Guetat, A. Ancel, S. Marchesin, J.-M. Dischler. *Pre-Integrated Volume Rendering with Non-Linear Gradient Interpolation* // IEEE Trans Vis Comput Graph, 16 (6), 2010, 1487-94.
- [7] D. S. Ebert and P. Rheingans. *Volume illustration: Non-photorealistic rendering of volume models* // In Proc. Visualization '00 (VIS '00), 195-202.
- [8] Nelson Max, Pat Hanrahan, and Roger Crawfis. *Area and volume coherence for efficient visualization of 3D scalar functions* // In Proceedings of the 1990 workshop on Volume visualization (VVS '90), 27-33.
- [9] Christian Sigg, Markus Hadwiger. *Fast Third-Order Texture Filtering* // In GPU Gems 2, Volume 2, 313-318.
- [10] The Volume Library by Stefan Roettger: <http://www9.informatik.uni-erlangen.de/External/vollib>

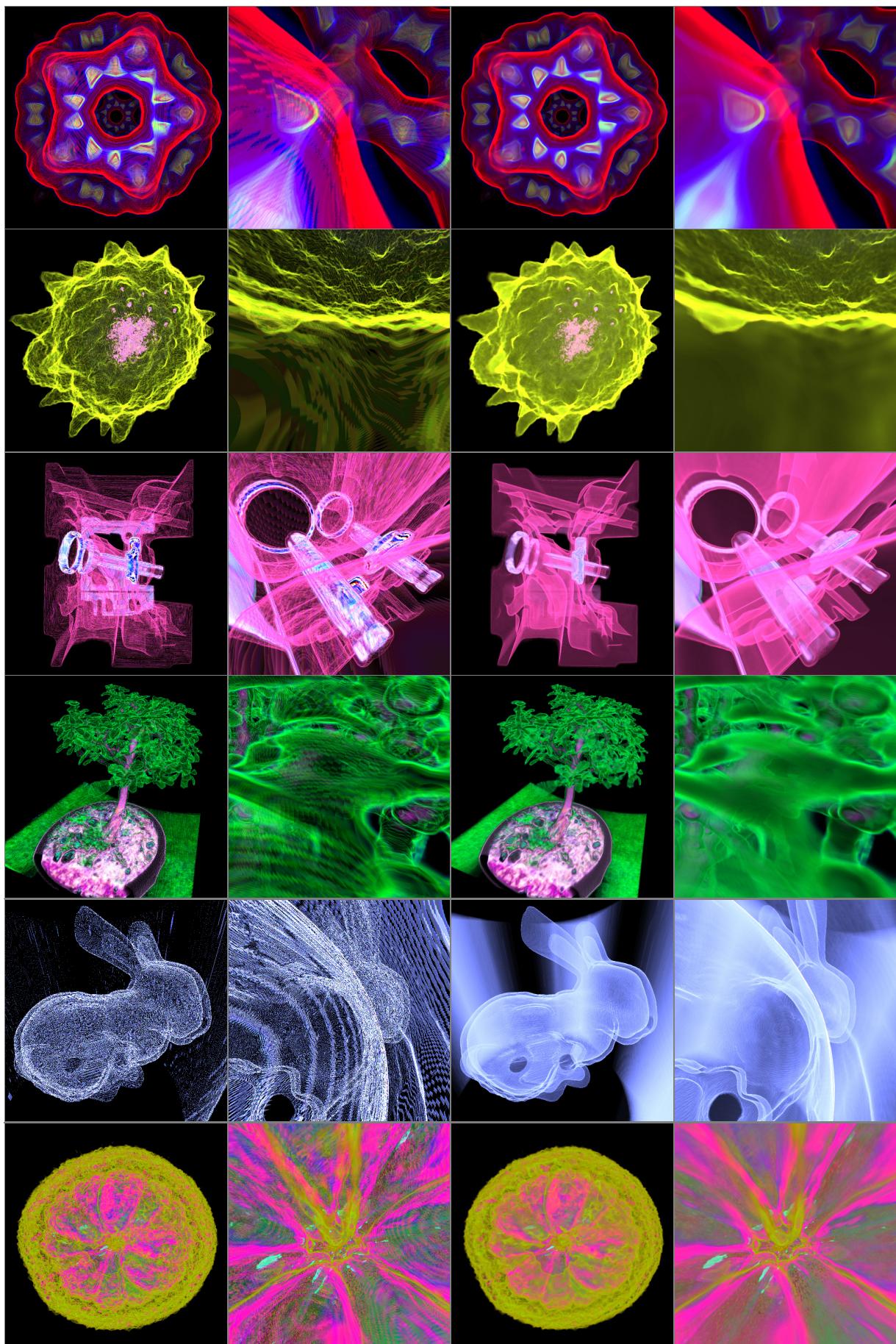


Рис. 3: Пост-классификация (первые два столбца) и прединтегрированная классификация (последние два столбца). Входные данные (сверху вниз): Bucky Ball, Daisy Pollen Grain, Engine Block, Bonsai Tree, Stanford Bunny, Orange.

# Качественная объёмная визуализация гигавоксельных массивов в блочном представлении на примерах данных из медицины

Николай И. Гаврилов, Вадим Е. Турлапов

факультет Вычислительной Математики и Кибернетики

Нижегородский государственный университет, Н.Новгород, Россия

[gavrilov86@gmail.com](mailto:gavrilov86@gmail.com); [vadim.turlapov@cs.vmk.unn.ru](mailto:vadim.turlapov@cs.vmk.unn.ru)

## Аннотация

Обсуждаются подходы, применённые авторами для достижения производительности и качества объёмного рендеринга на GPU методом Volume Raycasting, достаточных для интерактивной визуализации гигавоксельных томограмм. Изложены особенности разбиения на блоки, порядок трассировки блоков для лучшей производительности, а также методы подавления артефактов рендеринга и повышения качества визуализации: зашумление стартовых позиций лучей, трикубическая интерполяция, прединтегрированный рендеринг, адаптивное изменение и дробление шага луча. Представлены результаты вычислительных экспериментов на видеокартах среднего и высокого класса и обсуждаются их результаты.

**Keywords:** GPU-Based, Volume Raycasting, split into blocks, pre-integrated volume rendering, adaptive ray-step.

## 1. ВВЕДЕНИЕ

С ростом возможностей источников данных и графических процессоров постепенно сложились следующие изменения в развитии объемной визуализации: 1)внимание авторов обратилось на гига- и мультигига-воксельные объекты; 2)основным подходом к решению стала декомпозиция исследуемого объема; 3)ведущим методом стал Volume Raycasting; 4)ведущей проблемой стала проблема обеспечения высокого качества и функциональности изображения при сохранении интерактивности; 5)меняются отношения в решении задачи между центральным и графических процессором.

Так, один из признанных авторитетов в области качества визуализации, компания Fovia, реализовала коммерческую библиотеку для объёмного рендеринга High Definition Volume Rendering® (<http://www.fovия.com/>), которая обеспечивает визуализацию в реальном времени для массива  $4096^3$  или 64 гигавокселя. Один из секретов высокого качества визуализации - адаптивное изменение шага луча (<http://www.vizworld.com/2010/10/high-definition-volume-rendering-fovия-cpu/>). Авторы работ [9] и [3] рассматривают задачи с массивами данных объемом в  $8192^3$  и даже  $16384^3$  вокселей соответственно.

В случае использования GPU идеальным хранилищем объёмных данных является трёхмерная текстура, доступная при использовании расширений библиотеки OpenGL. Однако, существуют ограничения на размеры текстуры:  $512^3$  вокселя. Использование блочного представления данных позволяет обойти это ограничение, хотя в целом проблема ограниченности памяти, доступной GPU, конечно остаётся. Разумеется, что CPU тоже будет успешно оперировать лишь с объемом данных, помещающимся в оперативную память. Поэтому идея блочного представления в том или ином виде используется практически во всех работах последних лет и максимально плодотворна и на

CPU. Кроме декомпозиции на прямоугольные блоки, используются также блоки с полигональными границами [4]: даже невыпуклая полигональная граница способна повысить производительность вычислений. Есть попытки провести декомпозицию одновременно объема и окна визуализации согласованную с направлением лучей, получившее название slab-based рендеринга [8], чтобы уменьшить кэш-промахи для блоков, лежащих вдоль пакета лучей, находящегося в разделяемой памяти, стартующего с малого фрагмента окна. Показано также и непостоянство выигрыша и его ограниченная амплитуда (порядка 30%) и, в конечном счете, сложность лавирования между плюсами и минусами метода. В ряде работ используются октодеревья [3],[6]: в работе [3] - с целью организовать уровни разного разрешения (подобно [1]) для гигавоксельного объема, а в работе [6] - для организации сложной полигонально-воксельной сцены.

Метод Volume Raycasting показал себя как успешный инструмент и удаления пустого пространства на пути луча [13], и получения за один проход сразу несколько полезных для диагностики типов визуализации [2], и в визуализации данных смешанного разрешения [1].

Проблема обеспечения высокого качества и функциональности изображения при сохранении интерактивности включает в себя два направления: 1)обеспечение качества визуализации за счет минимизации шумов и артефактов, в том числе за счет различных модификаций метода *pre-integrated volume rendering* [11] и сплайн-интерполяции исходных данных; 2)обеспечение многообразного и качественного освещения реконструируемого объекта, обсуждаемого в работах [10], [7], [5]. Очередной виток интереса к освещению в 2011-2012 не случаен: он следствие того, что каждый из перечисленных выше методов вносит дополнительную сложность в реализацию для него полноценного освещения трехмерной сцены, отнимает заметный кусок интерактивности, требует дополнительной оптимизации вычислений.

## 2. МЕТОДЫ И АЛГОРИТМЫ

В нашей работе также используется Volume Raycasting и блочная декомпозиция данных, как единственная и эффективная возможность работы с гигавоксельными данными. При декомпозиции блоки данных записываются в разные трёхмерные текстуры. Текстуры имеют одинаковые размеры, за исключением текстур пограничных блоков. Чтобы избежать артефактов в местах соединения, соседние блоки должны перекрываться на толщину как минимум в один воксель (рис.1). Этого будет достаточно, если в алгоритме рендеринга при выборках значений из данных используется только трилинейная интерполяция. В нашей реализации блоки перекрываются на три вокселя, поскольку, во-первых, для реализации локального

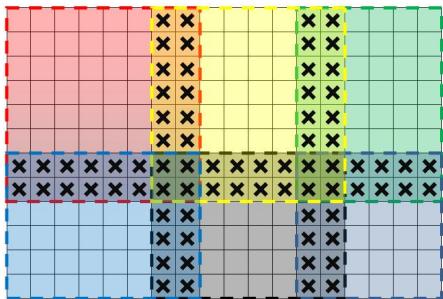


Рис.1: Иллюстрация декомпозиции данных на блоки.

освещения по Фонгу вычисляем градиент, и, во-вторых, мы используем трикубическую интерполяцию выборки вместо трилинейной, в том числе и при вычислении градиента.

## 2.1 Плюсы и минусы декомпозиции

### Плюсы:

- 1) Возможность загружать большие массивы данных, ограниченные вместимостью видеокарты (при 1 Гб можно визуализировать массив данных размером 512x512x2048).
- 2) Декомпозиция значительно повышает производительность визуализации, поскольку при рендеринге блока выборка идет из текстуры малого размера, что значительно быстрее выборки из текстуры большого размера. Блоки выводятся в порядке от наблюдателя: часть блоков экранируется пройденными ранее, что вызывает раннее завершение луча.
- 3) Пространство трассировки квантовано размерами блоков. Стратегия пропуска «пустых» областей реализуема еще производительнее. Например, для КТ томограмм обычно можно отбросить порядка 40% блоков, как не содержащих полезной информации.

### Минусы:

- 1) Более высокая сложность алгоритма рендеринга – главный недостаток блочного представления. При рендеринге ничего не известно о данных из соседних блоков за исключением слоя перекрытия с соседними блоками. Усложняется реализация отбрасывания теней, различных нелокальных техник освещения, и вообще всех техник, требующих генерирования вторичных лучей. Также усложняется рендеринг при использовании алгоритмов, в которых луч может нести некую добавочную информацию: в технике MIDA [2] для каждого луча, кроме накопленных цвета и непрозрачности, хранится максимальная интенсивность.
- 2) Усложнение алгоритма мульти-объёмного рендеринга, при котором нужно производить совместный рендеринг двух или более перекрывающихся в пространстве массивов объёмных данных, каждый из которых имеет блочное представление.
- 3) Перекрытие блоков означает, что приграничные воксели будут продублированы, поэтому при слишком мелком разбиении (в эксперименте - при размере блоков меньше 32<sup>3</sup>) экономия памяти GPU сведётся на нет.
- 4) При слишком мелком разбиении на блоки заметно увеличивается время, затрачиваемое на “переключение” между блоками. Результат рендеринга очередного блока необходимо скопировать из текстуры, в которую производился рендеринг, в текстуру, из которой будет читаться достигнутые значения (производительность значительно падает уже при размере регулярного блока 32<sup>3</sup> вокселей).

## 2.2 Пропуск пустых областей

Метод пропуска пустых областей применяется наиболее часто в CPU реализациях объёмного рендеринга. Используются оптимизирующие структуры: октодеревья, kd-деревья, полигональные сетки, ограничивающие трассируемый объём. В нашем случае для любой позиции камеры блок задаётся ограничивающим параллелепипедом, который определяет начальные и конечные точки движения лучей внутри него. Размер пограничных блоков минимизируется под размер наблюдаемых данных (рис.2).

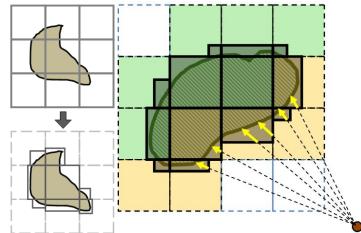


Рис 2: Подгонка ограничивающих параллелепипедов к областям данных.

Подгонка производится автоматически при любом изменении точки наблюдения и передаточной функции (transfer function). Для реализации раннего завершения луча при каждом изменении точки наблюдения для блоков производится разметка порядка трассировки (рис.3).

Результат трассировки очередного блока передается в следующий блок через текстуру. Библиотека OpenGL обеспечивает средства для рендеринга в текстуру через экранный буфер (Frame Buffer Objects). Используются две текстуры, в одну из которых производится рендеринг блока (текстура для записи), а другая хранит результат рендеринга предыдущих блоков (текстура для чтения).

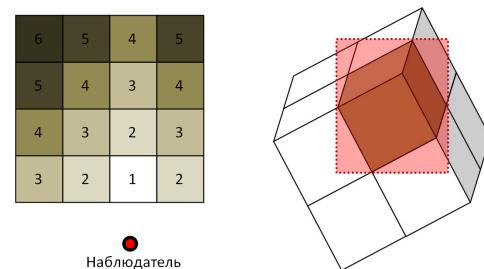


Рис. 3: Очерёдность вывода блоков (слева) и область экрана, копируемая в текстуру для чтения после рендеринга очередного блока данных (справа).

Время переключения между блоками в основном определяется временем копирования из текстуры для записи в текстуру для чтения. Для сокращения времени этого копирования мы копируем только ту часть экрана, в пределах которой производился рендеринг (рис.3).

## 2.3 Повышение качества визуализации

### 2.3.1 Подавление артефактов DVR

В методе бросания лучей луч движется с неким шагом через объём, делая на каждом шаге выборку значения из данных с использованием какой-либо интерполяции. Текущему значению данных передаточная функция ставит в соответствие оптические свойства, как правило, цвет и прозрачность, которые вносят вклад в искомый цвет

пикселя: луч пошагово “накапливает” цвет и непрозрачность. Благодаря возможной нелинейности передаточной функции, на интервале между двумя соседними отсчетами данных, цветовая интерпретация данных интерполированных передаточной функцией может существенно отличаться по оптическим свойствам от точек на концах отрезка. Это приводит к артефактам постклассификации. Из-за регулярности расположения стартовых позиций лучей эти артефакты имеют вид концентрических замкнутых линий (рис.4). Один из реализованных методов борьбы с данным артефактом – сдвиг стартовых позиций лучей на сонаправленный случайный вектор: накопление зашумлённых изображений даст усреднённое изображение без шума. Накопление кадров целесообразно применять при отсутствии мощного графического процессора, который был бы способен произвести рендеринг в реальном времени с достаточно малым шагом луча. Обычно для данных томографии при шаге менее 1/8 от размера вокселя артефакты уже не видны (рис.6).

Для устранения артефактов, связанных с недостаточной гладкостью исходных данных (в виде “ступенек”), реализована трикубическая интерполяция исходных данных «налету». Для этого опубликованный алгоритм бикубической фильтрации [12] расширен на трёхмерный случай. Для выполнения одной выборки с такой фильтрацией необходимо выполнить 8 выборок с трилинейной фильтрацией (см. рис.4).

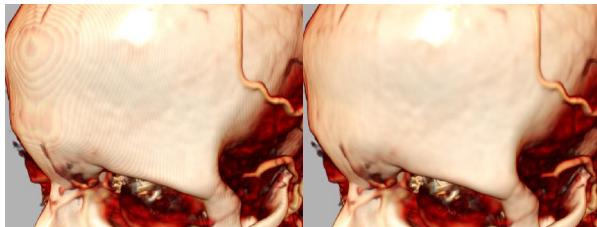


Рис. 4: Сравнение результатов рендеринга КТ томограммы при использовании обычной трилинейной интерполяции (слева) и трикубической интерполяции (справа).

### 2.3.2 Адаптивное изменение шага луча

Несмотря на возможность компенсировать недостаточно большой шаг луча техникой накопления кадров, более правильным подходом для рендеринга на видеокартах высокого класса является рендеринг одного изображения высокого качества, т.е. с малым шагом луча – объёмный рендеринг высокого разрешения (High definition volume rendering). Современные графические процессоры могут обеспечивать визуализацию в реальном времени больших данных с частотой выборки 8 на один воксель. Однако, производительность такого рендеринга можно значительно повысить, изменения длину шага луча аддитивно в зависимости от значения данных в текущей позиции, не ухудшая при этом качества рендеринга.

Предлагаемый нами алгоритм иллюстрирован на Рисунке 5. Мы трассируем луч с неким большим шагом (0,5 от размера вокселя) и на каждом шаге делаем выборку из данных в текущей позиции луча. В случае если мы попали в область видимых данных (т.е. значение текущей выборки не соответствует полной прозрачности), то мы сдвигаем наш луч на шаг назад и начинаем идти с шагом 1/8 от стандартного шага. При этом делается 8 шагов и накапливается цвет и непрозрачность так же, как в обычном рендеринге с постоянным шагом луча. На

видеокарте GeForce GTX 580 при шаге луча 1/8 такая оптимизация ускоряет рендеринг в среднем в 3 раза, при шаге 1/16 – в 6.2 раза.

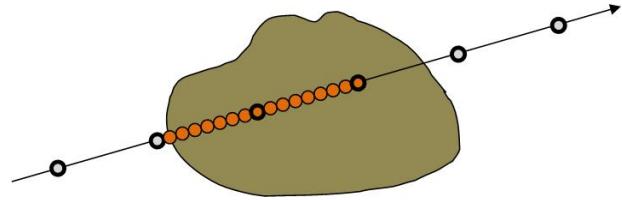


Рис. 5: Иллюстрация дробления шага луча в зависимости от значения выборки из данных. Луч двигается с большим шагом (жирные точки) и в случае нахождения в видимой области производится серия из 8 выборок на отрезке от предыдущего шага до текущего (восьмая выборка совпадает с позицией текущего шага).

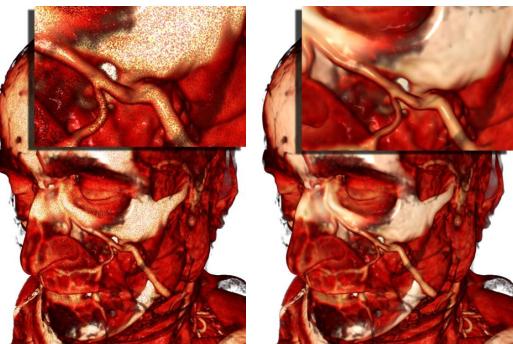


Рис. 6: Сравнение качества рендеринга (слева направо) при шаге луча в 1 и 1/8 от размера вокселя.

### 2.3.3 Дробление шага луча без увеличения количества выборок из данных

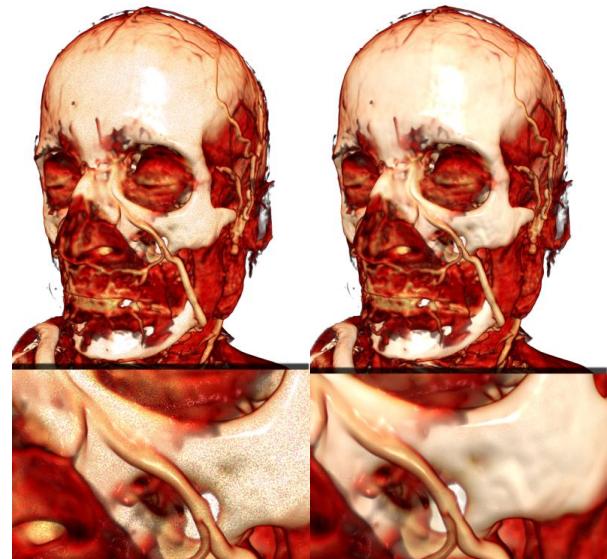


Рис. 7: Сравнение качества обычного (а) (4.6 fps) и метода разбиения шага луча (б) (4.2 fps). Шаг луча – 0,5 от размера вокселя.

Уменьшение шага луча хорошо устраниет артефакты, но число выборок при этом значительно снижает производительность. Предложен и исследован метод дробления шага луча без увеличения количества выборок

из данных. Шаг луча уменьшен в 20 раз, но выборка из данных выполняется только каждые 20 шагов. Внутри шага выборки используются интерполированные значения. Уже линейная интерполяция даёт значительное улучшение качества (рис.7).

#### 2.3.4 Прединтегрированный объёмный рендеринг (Pre-Integrated Volume Rendering)

Довольно эффективным методом устранения артефактов, связанных с “пролётом” луча мимо видимых особенностей в пространстве, является прединтегрированный объёмный рендеринг [11]. Идея объёмного рендеринга состоит в вычислении интеграла рендеринга вдоль луча для каждого пикселя экрана методом прямоугольников. Идея метода прединтегрированного рендеринга состоит в замене метода прямоугольников на метод более высокого порядка. Уже метод трапеций даёт значительное улучшение качества. Прединтегрированный рендеринг особенно хорошо (рис.8) оказывается на визуализации тонких слоев, которые могут быть пропущены в обычном рендеринге (избавиться от этого иногда не помогает и зашумление позиции старта луча - артефакты остаются даже при усреднении по десяткам кадров).

Данный метод отличается от метода разбиения шага луча тем, что здесь мы стараемся точно вычислить интеграл на отрезке между соседними выборками на луче (рис.9). В данном методе используется прединтегрированная таблица, которая для любых двух значений данных ( $a, b$ ) хранит цвет и непрозрачность, которые должен накопить луч, при проходе по отрезку, в начальной точке которого выборка из данных была равна  $a$ , в конечной –  $b$ . При изменении передаточной функции таблица пересчитывается.

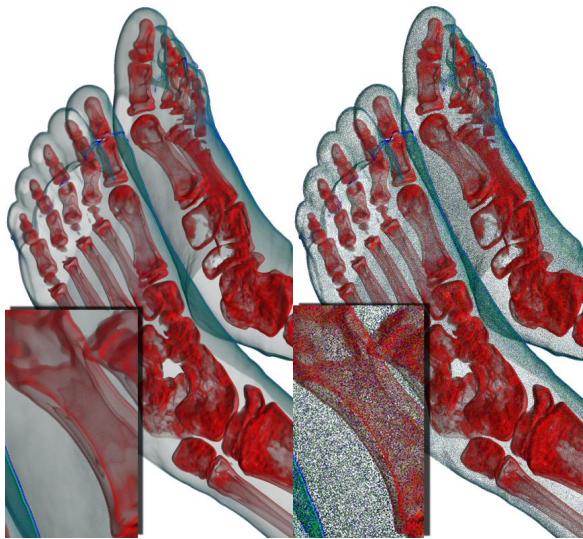


Рис. 8: Сравнение качества обычного (справа) и прединтегрированного рендеринга (слева). Шаг луча – 0,5 от размера вокселя.

Недостаток прединтегрированного рендеринга – необходимость вычислять прединтегрированную таблицу всякий раз при изменении передаточной функции. На практике изменение передаточной функции происходит в процессе интерактивной визуализации постоянно. Причём, в подходах, использующих методы интегрирования более высоких порядков размерность таблицы увеличивается. Так, для метода парабол нужна трёхмерная таблица, которая будет хранить оптические свойства отрезков путей

луча, которые будут определяться уже тремя значениями [11].

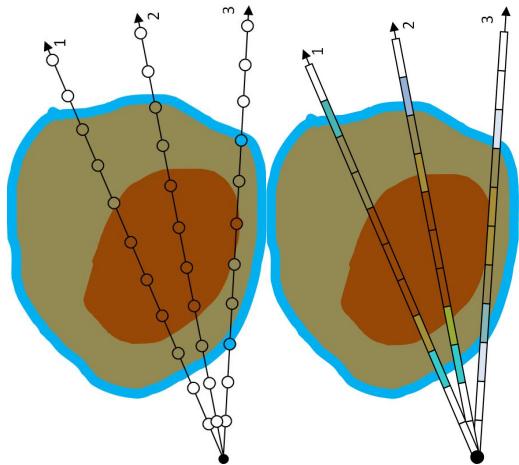


Рис. 9: Иллюстрация обычного рендеринга, при котором часть лучей не попадают в синюю область (слева) и прединтегрированного рендеринга (справа).

#### 2.3.5 Артефакты на границах блоков

Как было описано выше, для устранения видимых швов на границах соседние блоки должны перекрываться на границах, в нашей реализации толщина перекрытия соседних блоков составляет три вокселя, что позволяет правильно вычислять на границе блока градиент, даже при использовании трикубической интерполяции. Однако, для устранения артефактов также необходимо, чтобы для каждого пикселя искомого изображения путь луча был непрерывен. При рендеринге очередного блока лучи должны стартовать в позициях, бывшими “следующими” для соответствующих лучей при рендеринге предыдущих блоков. Единственная доступная информация о луче-предшественнике – это координаты соответствующего пикселя и накопленные цвет и прозрачность соответствующего пикселя. Координаты пикселя используются для генерирования случайного числа, которое используется при случайном сдвиге стартовой позиции луча для подавления артефактов. Т.к. для лучей из разных блоков, но находящихся на одной линии, координаты пикселя одни и те же, то и случайный сдвиг для них будет неизменен, что позволит продолжать движение луча без разрывов.

### 3. ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ

#### 3.1 Производительность для видеокарт потребительского класса

Ниже представлены результаты экспериментов по замеру производительности рендеринга при выборе различных размеров блоков, на которые мы делим массив данных. В нашем случае используются кубические блоки размерами стороны 32, 48, 64, 96, 128 и 256. Для выбранных тестовых массивов данных и видеокарты GeForce GTS 250 оптимальными размерами оказались 64 и 128 в зависимости от данных. Использовано 5 тестовых массивов данных, которые общедоступны в сети. Кроме того, мы используем три различных техники визуализации:

- 1) Объёмный рендеринг без затенения;

- 2) Объёмный рендеринг с затенением по локальной модели освещения Фонга;
- 3) Объёмный рендеринг без затенения с трикубической выборкой вместо трилинейной.

В данных экспериментах мы перекрываем наши блоки на толщину в два вокселя. Размеры окна вывода: 800x600 пикселей. Шаг луча в алгоритме Ray Casting: 0,34 от длины диагонали вокселя. Использованная видеокарта: GeForce GTS 250.



Рис. 10: Тестовые данные: A) hand 244x124x257 (КТ данные); B) Beetle 832x832x494; C) melanix 512x512x1203 (КТ данные); D) x-mas 512x512x999 (КТ ёлки); E) vessels 512<sup>3</sup> (МРТ томограмма сосудов головного мозга).

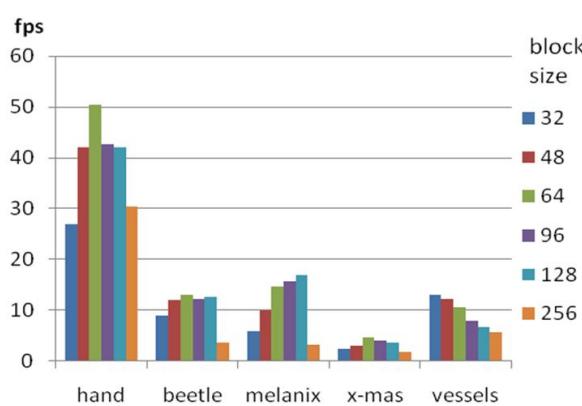


Диаграмма 1: Средняя по различным методам рендеринга производительность рендеринга при различных размерах блока разбиения данных.

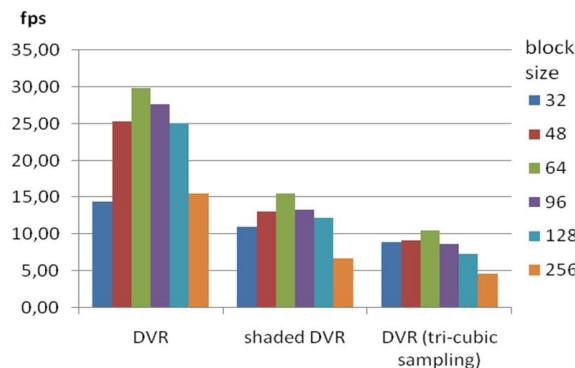


Диаграмма 2: Средняя по тестовым данным производительность рендеринга при различных размерах блока разбиения.

### 3.2 Ускорение оптимизированных подходов для видеокарты высокого класса

Для следующих экспериментов использовалась видеокарта GeForce GTX 580, размер блока данных был выбран равным 256<sup>3</sup>. При более мелком разбиении количество блоков становится довольно большим, что сильно увеличивает суммарное время на переключение между

блоками, особенно при их большом количестве. Так, массив данных multi2 был разбит на 88 блоков размером 256<sup>3</sup> вокселя каждый, не считая блоков меньших размеров на границах массива. Учитывая, что данные хранились на GPU в 12-битном формате (типичном для медицинских данных), весь массив занял 2Гб памяти GPU.

Для получения тестовых данных больших размеров в один массив были искусственно объединены различные томограммы. На диаграммах 3-4 представлены результаты замеров производительности рендеринга с переменным шагом луча и ускорение производительности от этой оптимизации.

В данных экспериментах мы перекрываем наши блоки на толщину в три вокселя, т.к. помимо трёх режимов рендеринга, использованных ранее, мы использовали режим рендеринга с локальным освещением с трикубической интерполяцией (режим cubic sDVR на диаграммах 3-4). Размеры окна вывода: 1920x1018 пикселей. Шаг луча в алгоритме Ray Casting: 1/8 от длины стороны вокселя. Использованная видеокарта: GeForce GTX 580 3Gb.

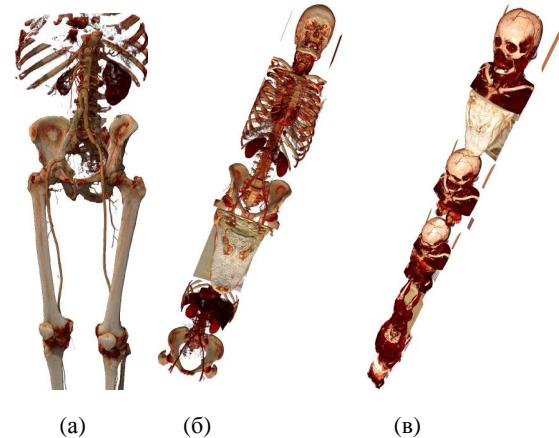


Рис. 11: Тестовые данные: а) amnesix 512x512x1624; б) multi1 512x512x1981; в) multi2 512x512x5382.

Благодаря технике изменения шага луча производительность рендеринга гораздо менее чувствительна к уменьшению длины шага луча. Дело в том, что регулярный шаг луча мы оставляем неизменным, но мы увеличиваем в два раза число выборок только в областях пространства, где данные видимы. Тестовые данные и настройки визуализации таковы, что луч обычно очень скоро накапливает непрозрачность и останавливается.

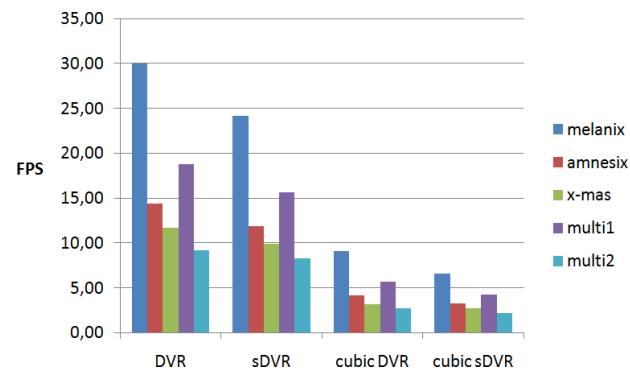
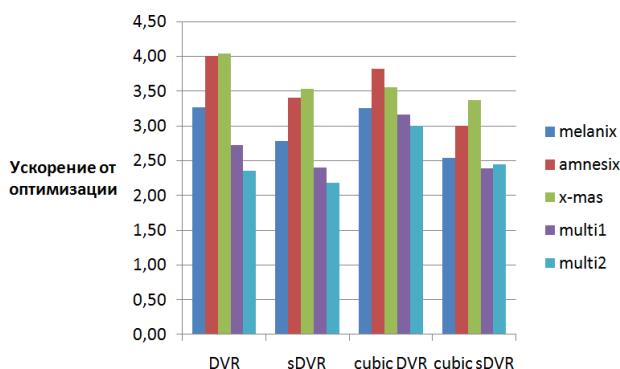


Диаграмма 3: Производительность рендеринга с адаптивным шагом луча



**Диаграмма 4:** Ускорение рендеринга с адаптивным изменением шага луча по сравнению с неоптимизированной версией при тех же условиях.

Однако, для случая среды с низкой непрозрачностью, в которой луч может трассироваться большое число шагов и на каждом из них накапливать цвет и непрозрачность, понадобится другой способ подстройки шага, при котором длина шага будет меняться более плавно при изменении оптических свойств среды.

Массив данных x-mas отличен от остальных тем, что имеет сложную топологию видимой области пространства, что затрудняет использование регулярных ускоряющих структур и ограничивающей геометрии для пропуска пустых областей. Техника изменения шага луча не восприимчива к топологии данных и является достаточно универсальной.

## 4. ЗАКЛЮЧЕНИЕ

Рассмотрены современные тенденции в развитии объемной визуализации и детали комплексного решения для интерактивного визуализатора гигавоксельных томограмм. Основу составила межплатформенная реализация на GPU блочной декомпозиции гигавоксельных данных для метода Volume Raycasting. Изложены особенности разбиения на блоки в условиях моделирования освещения реконструируемого объекта, устранение нежелательных артефактов, порядок рендеринга блоков для лучшей производительности, методы подавления артефактов рендеринга и повышения качества визуализации: трикубическая интерполяция, реализация прединтегрированного рендеринга, методы дробления и адаптивного изменения шага луча.

Результаты экспериментов показали высокую производительность рендеринга для видеокарты GeForce GTX 580 3Gb, достаточную для визуализации гигапиксельных томограмм.

Благодаря методам устранения артефактов и увеличения частоты выборок при рендеринге качество интерактивной визуализации сравнимо с качеством CPU-реализации от компании Fovia.

С ростом объемов данных и усложнением сцен продолжит расти значение оптимизирующих структур и методов, как и массивно-параллельных устройств, обеспечивающих потоковые вычисления нижнего уровня. Следует ожидать повышения роли одновременно и CPU, и GPU, и, особенно, их совместного применения.

## 5. ЛИТЕРАТУРА

- [1] J.Beyer et al,2008, *Smooth Mixed-Resolution GPU Volume Rendering*, IEEE International Symposium on Volume and Point-Based Graphics (VG '08); 2008. pp. 163–170.
- [2] S.Bruckner et al, 2009, *Instant volume visualization using maximum intensity difference accumulation*. Computer Graphics Forum, 28(3):775–782, June 2009.
- [3] C.Crassin et al, 2009, *GigaVoxels: Ray-Guided Streaming for Efficient and Detailed Voxel Renderin*. ACM SIGGRAPH Symp. on Interactive 3D Graphics and Games (I3D) - feb 2009, 8p.
- [4] B.Kainz et al, 2009. *Ray Casting of Multiple Volumetric Datasets with Polyhedral Boundaries on Manycore GPUs*, Proceedings of ACM SIGGRAPH Asia 2009, Volume 28, No 152.
- [5] J.Kronander et al, 2012, *Efficient Visibility Encoding for Dynamic Illumination in Direct Volume Rendering*. IEEE TVCG, Volume 18, Number 3, 2012, pp. 447-462
- [6] S.Laine, T.Karras (NVIDIA), *Efficient Sparse Voxel Octrees*, IEEE Transactions on Visualization and Computer Graphics, vol. 17(8), 2011, pp. 1048-1059.
- [7] F.Lindemann, T.Ropinski, 2011, *About the Influence of Illumination Models on Image Comprehension in Direct Volume Rendering*. IEEE TVCG. V.17(12), Dec.2011, pp. 1922-1931
- [8] J.Mensmann, et al, 2011, *Slab-Based Raycasting: Exploiting GPU Computing for Volume Visualization*, Springer, Communications in Computer and Information Science (CCIS), V. 229. - 2011. pp. 246-259.
- [9] E.Gobbetti et al, 2008, *A single-pass GPU ray casting framework for interactive out-of-core rendering of massive volumetric datasets*. - 8p.
- [10] M.Hadwiger. et al, 2006, *GPU-Accelerated Deep Shadow Maps for Direct Volume Rendering*, Proc.of the 21st ACM SIGGRAPH/EUROGRAPHICS symposium on Graphics hardware, Sep.03-04, 2006, Vienna, Austria, pp. 49-52
- [11] J.E.Hajjar et al, 2008, *Second order pre-integrated volume rendering* // IEEE Pacific Visualization Symposium, March 2008, pages: 9 – 16.
- [12] D.Ruijters et al, 2008. *Efficient GPU-Based Texture Interpolation using Uniform B-Splines*, In IEEE Transactions on Journal of Graphics, GPU, & Game Tools, Vol. 13, No. 4, pp 61-69.
- [13] V.Vidal(Inria) et al, 2008, *Simple Empty-Space Removal for Interactive Volume Rendering*, Journal of Graphics, GPU, and Game Tools, Volume 13, Issue 2, pp. 21-36

# Симуляция динамики жидкости и её взаимодействия с твёрдой поверхностью средствами графических ускорителей.

Юлия Хохлова

Физический факультет

Новосибирский государственный университет, Новосибирск, Россия

girl.from.ff.nsu@gmail.com

Денис Гладкий

ООО «Онлайнка», Новосибирск, Россия

generalgda@gmail.com

Станислав Кузиковский, Игорь Белаго

Институт автоматики и электрометрии СО РАН, Новосибирск, Россия

{stas, bel}@sl.iae.nsk.su

## Аннотация

В данной статье предлагается метод моделирования динамики жидкости и её взаимодействия с окружающим виртуальным миром в интерактивных графических приложениях. Такой подход применим для анимации разлива воды на дороге или попадания объекта под струю водопада. Объём жидкости представляется набором взаимодействующих друг с другом частиц, поверхность, формируемая окружающими жидкость объектами, задаётся картой высот и нормалей, а поверхность жидкости генерируется и отображается как полигональная сетка. Высокая скорость расчёта кадра, позволяющая производить анимацию в реальном времени, достигается за счёт использования возможностей эффективных параллельных вычислений общего назначения на современных графических процессорах с помощью программного интерфейса DirectX 11.

**Ключевые слова:** Моделирование Жидкости GPU, Карта Высот, Динамика Жидкости в Реальном Времени.

## 1. ВВЕДЕНИЕ

В современных приложениях виртуальной реальности, требования к которым по качеству генерируемых изображений постоянно растут, моделирование и визуализация жидкостей является актуальной задачей. Использование программируемого конвейера видеокарты в качестве математического сопроцессора – один из немногих способов получения интерактивных сцен с присутствующей на них жидкостью.

Полная реализация симуляции на графическом процессоре позволяет избежать многократного относительно медленного копирования данных между системной памятью и памятью видеокарты.

Эффективность моделирования взаимодействия жидкости с твёрдой поверхностью окружающего виртуального мира может быть повышена применением техники, не зависящей от способа представления объектов сцены.

## 2. ДОСТИЖЕНИЯ В ПРЕДМЕТНОЙ ОБЛАСТИ

В настоящее время в игре Alice: Madness Return [5] реализовано моделирование подобных явлений с использованием до 10000 частиц, однако отображение жидкости проводится обработкой проекции модели на экран. Множество реализаций моделирования жидкости

набором частиц представлено в работах по исследованию возможностей параллельных вычислений на видеокартах с использованием CUDA [4]. Однако в таких задачах не изучается ни вопрос взаимодействия жидкости с окружающим миром, ни применение различных методов визуализации результатов моделирования на экране.

В статье [1] предлагается метод, при котором высота столба жидкости не влияет на объём вычислений, поскольку детальное внимание сосредоточено на верхних слоях жидкости, но при симуляции процессов разлива жидкости по поверхности преимущества такого подхода не проявляют себя. В работе [2] представлена техника, комбинирующая моделирование большого объёма жидкости частицами и полем высот, при этом твёрдая поверхность окружающего мира задаётся полигональной моделью. В [7] описывается алгоритм, позволяющий представлять поверхность жидкости в виде набора треугольников, который на каждом шаге модифицируется, а не генерируется заново, но реализации метода была произведена только на центральном процессоре и дала частоту кадров анимации 6 кадр/с.

## 3. ВЫБОР ТЕХНОЛОГИЙ

В настоящее время существует множество реализаций моделирования жидкости набором частиц на графическом процессоре. В задачах компьютерной графики главная цель состоит в реалистичном визуальном результате, что подразумевает не только моделирование динамики жидкости, но и отображение её поверхности.

В отличие от широко применяемой в научных кругах технологии CUDA или OpenCL, разработка игр ведётся преимущественно на языке C++ с использованием программного интерфейса DirectX. Поэтому было решено использовать возможности Shader Model 5.0 DirectX 11, в частности, Compute Shader 5.0 для вычислений общего назначения, а также Geometry Shader 4.0 для генерации геометрии на видеокарте по произвольным данным, описывающим модель виртуального мира.

## 4. ДИНАМИКА ЖИДКОСТИ

### 4.1 Гидродинамика сглаженных частиц

Для моделирования жидкости предлагается использовать представление её объёма в виде системы частиц, что позволяет использовать достаточно простую форму уравнения Навье-Стокса

$$\rho \frac{\partial \mathbf{v}}{\partial t} = -\nabla p + \mu \Delta \mathbf{v} + \rho \mathbf{g}. \quad (1)$$

Влияние каждой частицы на свойства оценивается в соответствии с расстоянием до интересующей точки. Дискретные частицы имеют характерный радиус  $h$ , на котором их свойства сглаживаются радиально симметричной функцией ядра  $W(|r|, h)$ . Значение физической величины  $A$  в точке  $\mathbf{r}$  описывается выражением

$$A(\mathbf{r}) = \int A(\mathbf{r}') W(|\mathbf{r} - \mathbf{r}'|, h) d\mathbf{r}', \quad (2)$$

причём функция  $W$  должна удовлетворять следующим условиям:

1.  $\int_{R^3} W(|\mathbf{r}|, h) d\mathbf{r} = 1$  – условие нормировки.
2.  $\int_{R^3} A(\mathbf{r}') \lim_{h \rightarrow 0} W(|\mathbf{r} - \mathbf{r}'|, h) d\mathbf{r}' = A(\mathbf{r})$  – условие вырождения, т.е. при  $h \rightarrow 0$  величина определена в одной точке пространства.

В методе гидродинамики сглаженных частиц для каждого элемента жидкости отслеживаются координаты, скорость и плотность. Значение величины в конкретной точке получается суммированием по всем окружающим частицам

$$A(\mathbf{r}) = \sum_j A_j \frac{m_j}{\rho_j} W(|\mathbf{r} - \mathbf{r}_j|, h), \quad (3)$$

а значение, например, плотности в интересующей точке составляет

$$\rho(\mathbf{r}) = \sum_j m_j W(|\mathbf{r} - \mathbf{r}_j|, h). \quad (4)$$

Пренебрегая влиянием удалённых частиц, выбирается функцию ядра такая, что  $W(|\mathbf{r}|, h) = 0$  при  $|\mathbf{r}| > h$ . При этом отпадает необходимость проводить суммирование по всем частицам, ограничивших лишь близлежащим, что может значительно сэкономить время вычислений.

Преимущество гидродинамики сглаженных частиц состоит в простоте оценивания фигурирующих в уравнении (1) частных производных, поскольку их вычисление сводится к дифференцированию ядра:

$$\frac{\partial}{\partial x} A(\mathbf{r}) = \sum_j A_j \frac{m_j}{\rho_j} \frac{\partial W(|\mathbf{r} - \mathbf{r}_j|, h)}{\partial x}, \quad (5)$$

$$\nabla A(\mathbf{r}) = \sum_j A_j \frac{m_j}{\rho_j} \nabla W(|\mathbf{r} - \mathbf{r}_j|, h), \quad (6)$$

$$\Delta A(\mathbf{r}) = \sum_j A_j \frac{m_j}{\rho_j} \Delta W(|\mathbf{r} - \mathbf{r}_j|, h). \quad (7)$$

## 4.2 Моделирование сил

Перейдём к формулам расчёта сил, действующих на частицу в текущий момент времени, и приобретаемого при этом частицей ускорения.

$$\mathbf{a}_i = \frac{\mathbf{f}_i}{\rho_i} \quad (8)$$

$$\mathbf{f}_i = \rho_i \mathbf{g} + \mathbf{f}_i^{pressure} + \mathbf{f}_i^{viscosity} \quad (9)$$

Согласно формуле (3) при использовании метода гидродинамики сглаженных частиц сила давления находится по формуле (10)

$$\mathbf{f}_i^{pressure} = -\nabla p(\mathbf{r}_i) = -\sum_j m_j \frac{p_j}{\rho_j} \nabla W(|\mathbf{r}_i - \mathbf{r}_j|, h) \quad (10)$$

Однако в такой ситуации сила давления будет несимметрична. При взаимодействии двух частиц, частица  $i$  учитывает влияние только частицы  $j$  и наоборот, а полученные значения сил могут быть различны по величине, что противоречит третьему закону Ньютона. В литературе описаны различные решения этой проблемы, согласно [3] приемлема с точки зрения скорости

вычислений и стабильности формула моделирования силы давления (11)

$$\mathbf{f}_i^{pres} = -\nabla p(\mathbf{r}_i) = -\sum_j m_j \frac{p_j + p_i}{\rho_j} \nabla W(|\mathbf{r}_i - \mathbf{r}_j|, h). \quad (11)$$

Давление в точке предлагается моделировать простой формулой (12), где  $\rho_0$  – плотность жидкости в свободном состоянии.

$$p = k(\rho - \rho_0) \quad (12)$$

Сила вязкости определяется соотношением (13)

$$\mathbf{f}_i^{visc} = \mu \Delta p(\mathbf{r}_i) = \mu \sum_j m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \Delta W(|\mathbf{r}_i - \mathbf{r}_j|, h). \quad (13)$$

Опять же, для симметричности силы вводится модифицированная формула 14:

$$\mathbf{f}_i^{visc} = \mu \Delta p(\mathbf{r}_i) = \mu \sum_j m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \Delta W(|\mathbf{r}_i - \mathbf{r}_j|, h). \quad (14)$$

Подробные обоснования выбора моделей действующих сил приводятся в [3, 8].

## 4.3 Функции ядра

За время применения гидродинамики сглаженных частиц было предложено множество функций, удовлетворяющих требованиям, описанным в пункте 4.1. При моделировании жидкости в задачах компьютерной графики наибольшее распространение получили следующие функции, которые используются при реализации алгоритма.

1. Для вычисления плотности частиц используется

$$W_{poly6}(r, h) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - r^2)^3, & \text{если } 0 \leq r \leq h \\ 0, & \text{иначе} \end{cases} \quad (15)$$

2. Для расчёта силы давления жидкости, действующей на частицу жидкости, применяется функция

$$W_{spiky}(r, h) = \frac{15}{\pi h^6} \begin{cases} (h - r)^3, & \text{если } 0 \leq r \leq h \\ 0, & \text{иначе} \end{cases} \quad (16)$$

3. Для учёта вязкости в жидкости служит функция

$$W_{visc}(r, h) = \frac{15}{2\pi h^3} \begin{cases} -\frac{r^3}{2h^3} + \frac{r^2}{h^2} + \frac{h}{2r} - 1, & 0 \leq r \leq h \\ 0, & r > h \end{cases} \quad (17)$$

Подробное обоснование выбора ядер можно найти [10].

## 5. ВЗАИМОДЕЙСТВИЕ С ПОВЕРХНОСТЬЮ

### 5.1 Представление поверхности

Окружающий виртуальный мир, в котором симулируется жидкость, может быть представлен самым различными моделями данных. При этом учёт каждой из них при взаимодействии с элементами жидкости создаёт необходимость специальной подготовки дополнительных данных художниками, их хранения в памяти и специальная обработка, что неэффективно. По этой причине предлагается получать данные о строении поверхности путём отрисовки модели окружающего мира в карту высот. На современных видеокартах эта операция производится достаточно быстро [11]. Более того, карта высот зачастую генерируется для использования в других алгоритмах, таких как освещение или обнаружение столкновений. Также для расчёта взаимодействия частиц с полем высот требуется соответствующая поверхности карта нормалей.

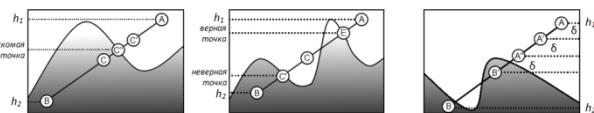
### 5.2 Поиск точки контакта

Для обнаружения столкновения частицы с поверхностью и обработки отражения был адаптирован алгоритм отображения рельефных текстур [9].

Траектория движения частицы между положениями в двух последовательных моментах времени является прямой. На

каждом шаге моделирования для каждой частицы происходит проверка на попадание под поверхность. Если частица оказывается ниже поверхности, то необходимо произвести поиск пересечения отрезка траектории с картой высот. Для этого используется метод дихотомии.

Чтобы избежать ошибки этого метода при многократном пересечении траектории с поверхностью (см. рисунок № в центре), сначала выполняется линейный поиск с постоянным шагом  $\delta$ , размер которого зависит от угла между направлением движения частицы и горизонтальной плоскостью. При обнаружении двух точек, одна из которых над поверхностью, другая - под ней, запускается алгоритм двоичного поиска.

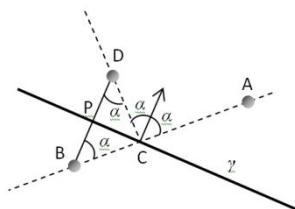


**Рисунок 1:** Поиск пересечения отрезка с полем высот методом дихотомии (слева). Возможные ошибки прямого применения метода дихотомии (в центре). Выполнение линейного поиска для исключения ошибки метода дихотомии (справа).

Пусть А и В – начальное и конечное положение частицы, причём А находится над поверхностью, В – под поверхностью. Берём точку С – среднюю точку между А и В и проверяем её положение относительно поверхности. Если точка С оказывается над поверхностью, повторяем действия для отрезка СВ, иначе, для отрезка АС. Схема работы алгоритма представлена на рисунке № слева.

### 5.3 Отражение частицы

Вблизи точки столкновения элемент поверхности задаётся плоскостью, ориентированной нормалью. Отрезки траектории частицы до и после отражения лежат в одной плоскости. Если частица пересекла поверхность в точке С, то после упругого удара она должна оказаться в точке D, симметричной точке В относительно прямой  $\gamma$ , по которой пересекаются плоскости падения-отражения и поверхности.



**Рисунок 2:** Схема отражения частицы от плоскости

Пусть О – начало координат, тогда  $\vec{OA}$  и  $\vec{OB}$  – трёхмерные векторы координат точек А и В соответственно. Описанным выше алгоритмом находятся координаты точки С. Далее необходимо вычислить положение  $\vec{OD}$  частицы после отражения. Итак,

$$\vec{OD} = \vec{OB} + \vec{D} = \vec{OB} + 2(\vec{C}, \vec{\gamma}) \cdot \vec{\gamma}$$

Между тем,

$$\vec{\gamma} = C \cos \alpha \cdot \vec{\gamma} = (\vec{C}, \vec{\gamma}) \cdot \vec{\gamma},$$

тогда

$$\vec{OD} = \vec{OB} + 2(\vec{C}, \vec{\gamma}) \cdot \vec{\gamma}$$

Аналогичными рассуждениями находится изменение направления движения частицы при сохранении модуля скорости.

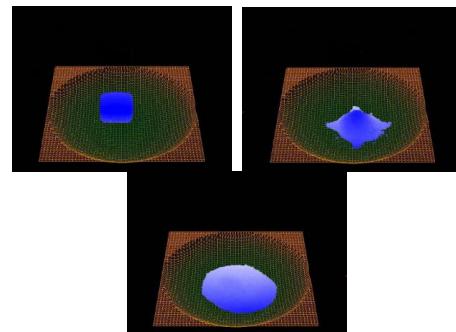
Заметим, что коэффициент при множителе  $(\vec{C}, \vec{\gamma}) \cdot \vec{\gamma}$  заключает в себе степень упругости удара и может изменяться в интервале от 1 до 2. Абсолютно упругий удар соответствует коэффициенту 2. Промежуточные значения указанного интервала симулируют различную степень потери энергии частицы при столкновении. Однако в случае моделирования жидкости более естественным выглядит неупругое столкновение с условием непротекания, когда нормальная компонента скорости частицы после контакта со стенкой становится нулевой. Такая модель соответствует коэффициенту 1.

## 6. ВИЗУАЛИЗАЦИЯ ЖИДКОСТИ

При выборе метода визуализации жидкости было поставлено требование получения полигональной модели поверхности жидкости, которая в дальнейшем может быть применена для наложения текстур или построения теней. Эффективным решением поставленной задачи является метод марширующих кубиков (Marching cubes) [6], основной параметр которого заключается в выборе размера ячейки при построении решётки. На практике было установлено, что удовлетворительный визуальный результат отображения поверхности при требовании интерактивности достигается при размере ячейки, по порядку величины равном размеру частиц.

## 7. РЕЗУЛЬТАТЫ

На рисунке 3 приведены результаты моделирования с использованием 65 тысяч частиц и размером ячейки марширующих кубиков равным 0,7 размера частицы. Симуляция проводилась в режиме реального времени с частотой кадров 47 кадр/с.



**Рисунок 3:** Кадры анимации.

При реализации предложенного алгоритма моделирования динамики жидкости и её взаимодействия с твёрдой поверхностью на графическом процессоре NVIDIA GeForce GTX 550Ti достигнута следующая производительность (данные приведены в таблице 1).

Количество частиц	Частота кадров, кадр/с	Время кадра, мс
8192	315	3,2
16384	270	3,7
32768	200	5
65536	115	8,7
131072	60	16,7
262144	25	40

**Таблица 1:** Результаты измерений.

## 8. ЗАКЛЮЧЕНИЕ

В статье был предложен и реализован метод симуляции динамики жидкости, представленной набором частиц, и её взаимодействия с твёрдой поверхностью, задаваемой картой высот, что делает алгоритм не зависимым от модели представления объектов окружающего виртуального мира, с которыми взаимодействует жидкость. Реализация на графическом ускорителе позволила получить скорость расчёта кадра анимации, достаточную для применения в интерактивных графических приложениях. Техника реализации демонстрирует возможности Shader Model 5.0 DirectX 11, а также использует возможность генерации геометрических примитивов на видеокарте по произвольной модели данных, описывающей объект виртуального мира.

На данном этапе было поставлено требование интерактивности анимации, которое и было достигнуто. Оптимизация полученного решения будет проведена в дальнейшей работе над проблемой.

## 9. ССЫЛКИ

- [1] Nuttapong Chentanez, Matthias Müller. *Real-Time Eulerian Water Simulation Using a Restricted Tall Cell Grid* / ACM Transactions on Graphics (SIGGRAPH 2011).
- [2] Nuttapong Chentanez, Matthias Müller. *Real-Time Simulation of Large Bodies of Water with Small Scale Details* / SIGGRAPH/EUROGRAPHICS Proceedings, 2010.
- [3] Mathieu Desbrun, Marie-Paule Cani. *Smoothed Particles: A New Paradigm for Animating Highly Deformable Bodies* / Proceedings of EG Workshop on Animation and Simulation, 2009.
- [4] Nolan Goodnight. *CUDA/OpenGL Fluid Simulation* / Technical article, NVIDIA Corporation, 2007.
- [5] Simon Green, Richard Tonge, Miguel Sainz, Dane Johnston, David Schoemehl. *Fluid Simulation in Alice: Madness Returns* / 2011.
- [6] William E. Lorensen, Harvey E. Cline. *Marching cubes: A high resolution 3d surface construction algorithm* / SIGGRAPH Proceedings, 1987.
- [7] Matthias Müller. *Fast and Robust Tracking of Fluid Surfaces*. ACM SIGGRAPH / EUROGRAPHICS Proceedings, 2009.
- [8] Matthias Müller, David Charypar, Markus Gross. *Particle-Based Fluid Simulation for Interactive Applications* / Eurographics/SIGGRAPH Proceedings, 2003.
- [9] Fabio Pollicarpo, Manuel M. Oliveira, Joao L. D. Comba. *Real-Time Relief Mapping on Arbitrary Polygonal Surfaces* / SIGGRAPH Proceedings, 2005.
- [10] Natalya Tatarchuk, Jeremy Shopf, Chris DeCoro. *Scalar to Polygonal: Extracting Isosurfaces Using Geometry Shaders* / ShaderX7: Advanced Rendering Techniques, Course Technology, 2009.
- [11] NVIDIA Corporation. *GPU Programming Guide GeForce 8 and 9 Series*, 2008.

# Analysis of Unsteady Space-Time Structures Using the Optimization Problem Solution and Visualization Methods

A.E. Bondarev

Keldysh Institute for Applied Mathematics RAS, Russia, Moscow

[bond@keldysh.ru](mailto:bond@keldysh.ru)

## Abstract

This paper presents an approximate approach to analysis of space-time structures for unsteady problems in CFD (computational fluid dynamics). The approach is based on the solution of optimization problem combined with methods of data visual presentation. This approach is intended for fast approximate estimation of unsteady flow structures dependence on character parameters (or determining parameters) in a certain class of problems. For some cases such approach allows to obtain the sought-for dependence in a quasi-analytical form.

**Ключевые слова:** *unsteady problems, space-time structures, optimization, inverse problems, visualization methods*

## 1. INTRODUCTION

Time-dependent processes in CFD problems are often accompanied by the presence of changeable space-time structures (STS) in the flow, such as separation zones, circulating flows, vortex bursts, etc. These structures cause many undesirable effects in practice: reduced lift, airframe and control vibrations. STS can appear and disappear defining the flow pattern and quantitative characteristics of the flow field. Simulating these changeable structures is therefore an important aspect of CFD.

Nowadays, modern computer hardware and mathematical methods allow one to simulate practically any time-dependent physical process in CFD and to obtain corresponding field of physical values. Calculating thoroughly the flow field one can obtain a beautiful picture of STS transformations. Nevertheless it is evidently insufficient for practical aims. In practice the phenomenon (physical effect) by itself is not the main point of interest. For practical engineering it is more interesting to know the circumstances leading to the phenomenon appearance, i.e. how this appearance depends on the problem character parameters, such as Mach number, Reynolds number, Prandtl number etc. To define such dependence one should solve the problem in optimization statement, which is based on multiple calculations of the inverse problem.

This paper presents an approximate approach to analysis of space-time structures. This approach is intended for fast and rough estimation of unsteady flow structures dependence on character parameters in a certain class of problems. The approach is based on the solution of optimization problem combined with methods of data visual presentation. Visualization methods are applied to the solution of optimization problem. The solution is obtained in a form of multidimensional array. According to classification described in [1], such approach can be referred to *data analysis methods*.

According to [2], there are two high-priority tasks for parallel computations: multidisciplinary problems and inverse problems. From this point of view the described below approach is very promising because it can be applied to a very wide range of time-dependent processes for various practical applications. Using the theory of optimization problems solution the approach is a modification of method [3] (parametric space analysis).

As it is shown below for concrete examples, this approach allows obtaining the sought-for dependencies in a form of quasi-analytical expressions.

## 2. INVERSE PROBLEMS AND CFD APPLICATIONS

Numerical computation of the inverse problems in CFD is enough difficult. One should calculate 4D problems (3D+time) in a *variational statement*. It requires using high-performance computers. The separate difficulty is to visualize the inverse problems solutions for multidimensional case. There is a significant lack of concepts and tools in scientific visualization now. Nevertheless, the rapid development of computing technologies and hardware allow us to solve this class of numerical simulation problems.

We can solve a wide range of CFD problems using the concept and statement of inverse problems. Typically, the practical engineering problem is to choose the desired variants from the set of admissible variants. This can be the choice of a geometric shape (minimal drag), the choice of flow control (maximal mixing), etc. According to [4,5], the inverse problems are classified as boundary searches, coefficient searches, retrospective inverse problems, optimization problems. In general, for practical goals inverse problems are formulated as follows. One should find the determining parameters for which a phenomenon of interest occurs in a certain class of problems. It shouldn't depend on the details of the phenomenon appearance. It can be quantitative appearance of the phenomenon (some variable reaches definite value) or qualitative appearance (vortex formation, flow separation, any other STS changing).

Let's consider formalized statement of the inverse problem in a general form.

The numerical solution of the chosen CFD problem is elaborated during the computation process. The solution is defined by the finite set of determining parameters (or character parameters) of the problem. These determining parameters can be divided into three main groups:  $A = (a_1, \dots, a_n)$  - the parameters determining physical properties and mathematical model of problem;  $B = (b_1, \dots, b_m)$  - the parameters determining the numerical method;  $C = (c_1, \dots, c_l)$  - the parameters determining organization of the calculation process.

All these parameters form the numerical solution

$F = F(A, B, C) = F(a_1, \dots, a_n, b_1, \dots, b_m, c_1, \dots, c_l)$  as a result of computation. So, the solution is based on the chosen mathematical model, numerical method and the way of calculation organization.

We consider the *event functional*  $\Phi(F(A, B, C))$ . Just as logical variable so this functional has two values:  $\Phi(F(A, B, C)) = 1$  – if the event of interest occurs (independently on the event details), or  $\Phi(F(A, B, C)) = 0$  – if the event of interest doesn't occur.

Presenting  $\Phi(F(A, B, C))$  in a form

$\Phi(F(A, B, C)) = \Phi(a_1, \dots, a_n, b_1, \dots, b_m, c_1, \dots, c_l)$  one can formulate the inverse problem in a general form as follows. Find all the determining parameters  $(a_1, \dots, a_n, b_1, \dots, b_m, c_1, \dots, c_l)$  for which a phenomenon of interest occurs in a certain class of problems, i.e.  $\Phi(a_1, \dots, a_n, b_1, \dots, b_m, c_1, \dots, c_l) = 1$  (2.1)

Considering the determining parameters  $(a_1, \dots, a_n, b_1, \dots, b_m, c_1, \dots, c_l)$  as a set of basis vectors, one can present the space of the determining parameters  $L(a_1, \dots, a_n, b_1, \dots, b_m, c_1, \dots, c_l)$  having  $(n + m + l)$  dimensions.

Then for general case the inverse problem can be formulated as the problem of finding in this space  $L$  all the subdomains  $L^*$  where the event of interest is observed, i.e.  $\Phi(L^*) = 1$ .

At the same time the problem of data filtration is solved. Setting the ranges for character parameters one can not guarantee the fact of appearance of the sought-for event inside the range. So if the event does not occur for some point of space, this point is not considered.

### 3. OPTIMIZATION PROBLEM AND VISUALIZATION

Using numerical or experimental modeling of unsteady phenomenon for practical goals in mechanics we usually know the reason of phenomenon appearance and quantitative parameter regulating this reason (control parameter)  $f_{cont}^*$ . The simulation is intended to define the control parameter dependencies on the determining parameters  $(f_1, \dots, f_n)$  of the problem. To obtain such dependencies  $f_{cont}^*(f_1, \dots, f_n)$  in a quasi-analytical or in a table form is a real practical goal of research. As a matter of fact these dependencies have been the main point of practical CFD applications last 50 years.

This paper considers a methodological approach to obtain these dependencies by means of numerical simulation. The approach can be described in general as follows.

Let's suppose that one has mathematical model of the CFD problem and reliable numerical method for solving. Then one can compute the straight problem of unsteady process simulation. During this simulation some event occurs.

To study thoroughly the unsteady event one should solve the inverse problem with purpose to find the exact value of control parameter, when the event (STS transformation, for instance) occurs.

To solve inverse problem one should multiply solve the straight problem varying the control parameter  $f_{cont}^*(f_1, \dots, f_n)$  until the

onset of the unsteady phenomenon (physical effect) of interest. During this optimization process the set of determining parameters  $(f_1, \dots, f_n)$  is fixed.

Then the determining parameters  $(f_1, \dots, f_n)$  are varied with chosen variation step and the inverse problem is repeatedly solved for each set of determining parameters. As a result of such computations we obtain control parameter dependence on determining parameters in general form of n-dimensional array  $f_{cont}^*(f_1, \dots, f_n)$ .

This form is not suitable for practical goals. The most effective way of a search of the sought-for dependence in a quasi-analytical form is visual presentation of the array.

Analyzing the array one can decrease the dimensionality. For this purpose one should omit those determining parameters, which do not influence at the control parameter. If one has as a result  $n \leq 3$  after such decreasing, then the rest of data can be visualized. For some cases the visualization is fast and effective way to obtain the dependence  $f_{cont}^*(f_1, \dots, f_n)$  in a form of quasi-analytical expression. To obtain such expressions it is assumed to approximate the array data (where it is possible) by simple geometric elements, such as lines, planes, parts of spheres etc.

It is very simple to do if we are dealing with the case of two determining parameters. We can approximate the surface  $f_{cont}^*(f_1, f_2)$ . For the case of three determining parameters we are able to build the isosurfaces. Then we can try approximating the isosurfaces by means of simple geometric elements. But for the case  $n > 3$  there is an evident lack of concepts and tools for visual presentation. The creation of reliable and suitable for human acceptance visual presentations for this case is a subject for discussions now.

By tradition the problem of multidimensional data visualization can be referred to the field of *Information Visualization* due to the fact that the solutions of such problems are necessary for business applications. There were some attempts to elaborate original visualization methods for multidimensional data. For instance, one can mention such approach as "Chernov's faces" [6]. The main idea of this approach is to present the values of different variables by various details of human face. Another attempt is presented in [7] for the space of events as "event tunnel". The space of events is presented in a form of 3D cylinder ("event tunnel"). Lengthwise axis presents the time; the events are presented as spheres inside the cylinder. As the distance from the point of observation grows, the sizes of spheres are reduced. Despite some success for business applications the artificial character of such visual concepts is evident.

So for common case one should try to decrease the array dimensionality up to 3 and hope the class of problems under consideration would allow such decreasing. Fortunately, for many real applications it is true, as it is shown below.

There are some well-known ways to decrease the array dimensionality. One of these ways is analysis of variances for each character parameter. Character parameter is considered as coordinate direction. The direction with minimal variances is rejected (compactification). Another way is the construction of different 3D projections for various triplets of determining parameters, as it is shown below for example of computations. Also one can apply PCA method (Principal Component Analysis)

[8]. This method is based on localization of 3 principal components and data presentation using these components as new coordinate system. Combining these methods one can decrease the array dimensionality for many practical cases.

Our approach is based on optimization. Therefore, it amounts to solving a set of similar small tasks. Each small task is the solution of inverse problem having a fixed set of determining parameters. So the approach corresponds to task parallelism ideology. Using the principle “one task – one processor” and having a minimal quantity of internal exchanges allows one to make very effective applications of this approach to practical problems. The possibility of rough grids using is another one serious advantage of the approach to be described.

The approach can be applied to analysis of finite difference schemes also. This methodological approach was successfully used in [9] to optimize the computational properties of hybrid finite difference scheme applied to the solution of the supersonic far wake problem. Viscosity and turbulence were taken into account. The emergence of undesired oscillations was considered as the event to be controlled. The weight coefficient of finite difference scheme was used as the control parameter. The grid step, Mach and Reynolds numbers were chosen as determining parameters. As a result for chosen class of problems the weight parameter dependence on determining parameters was obtained in a quasi-analytical form.

#### 4. APPLICATION EXAMPLES

The optimization approach is applied to analysis of unsteady circulating zones transformation. The problem of unsteady interaction of the supersonic viscous flow with jet obstacle is considered. This obstacle appears due to co-current underexpanded jet exhausting from the nozzle. The nozzle is placed to external supersonic viscous flow. Expanding jet propagates on the external surface of the nozzle and creates obstacle in external flowfield. The obstacle disturbs external flow and circulating zone appears ahead the obstacle. Typical flow structure is shown in Fig.1 by streamlines.

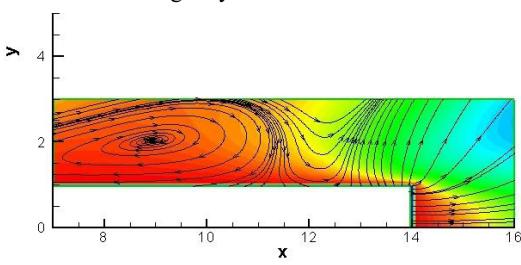


Figure 1: Flow structure for slow pressure ratio growth.

We consider a problem containing time-dependent boundary condition for underexpanded jet. Jet pressure ratio was set at the nozzle edge as time-dependent function  $n = n(t) = P_a / P_\infty$  (where  $P_a$  - jet pressure,  $P_\infty$  - external flow pressure). The full system of time-dependent Navier-Stokes equations for viscous compressible heat-conductive flow is used as mathematical model. Implicit hybrid finite difference WW-scheme is applied to solve the system of equations. This scheme has second order accuracy in time and space.

The dependence  $n = n(t)$  is chosen as linear function. It allows one to set different rates of pressure ratio growth up to  $n = 100$ .

As a result of calculations of straight problem a new STS formation is obtained. Increasing the rate of pressure ratio growth one obtains new space-time structure in the vicinity of circulating zone ahead the jet obstacle. This new structure is shown by streamlines in Fig.2.

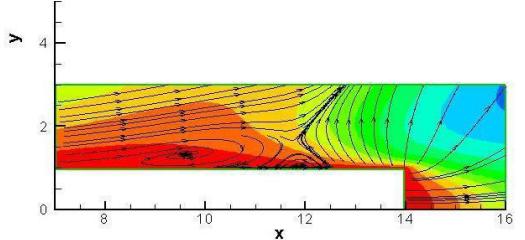


Figure 2: Flow structure for fast pressure ratio growth.

Let's consider the optimization approach application to analysis of unsteady event: the formation of the new space-time structure in the flow. The rate of pressure ratio growth is chosen as control parameter. The case of four determining parameters is considered. These four parameters are Mach number  $M_\infty$ , Reynolds number

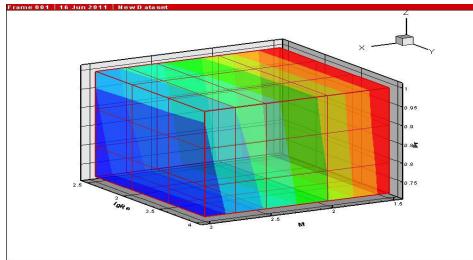
$Re_\infty$ , Prandtl number -  $Pr_\infty$  and  $Sh_\infty$  - Strouhal number for the problem under consideration. For each fixed set of these numbers  $(M_\infty, Re_\infty, Pr_\infty, Sh_\infty)$  the inverse problem is solved by varying pressure ratio growth rate until the onset of the new structure formation in the flow. These character numbers vary in ranges:  $1.5 \leq M_\infty \leq 3$ ;  $2.5 \leq \lg Re_\infty \leq 4$ ;  $0.72 \leq Pr_\infty \leq 1$ ;  $1 \leq Sh_\infty \leq 2$ . For each new fixed set of numbers  $(M_\infty, Re_\infty, Pr_\infty, Sh_\infty)$  the procedure described above is repeated.

So for each set of determining parameters  $(M_\infty, Re_\infty, Pr_\infty, Sh_\infty)$  one defines the exact value of pressure ratio growth rate when the new flow structure appears. Using this value one can just define the character time  $t^*$  and crucial velocity of pressure ratio growth  $V^*$  for the new STS appearance. As event character time  $t^* = t_{ev} / t_{n=100}$  is chosen, here  $t_{ev}$  - the time when the event occurs and  $t_{n=100}$  - the time when pressure ratio reaches 100.

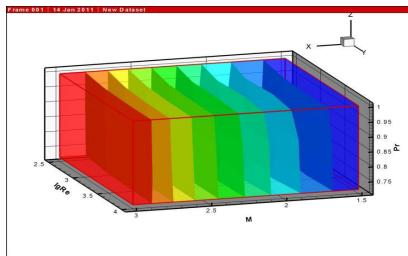
For computations two algorithms are elaborated – serial and parallel. Two types of grids are chosen: 5 and 10 points for each determining parameter. It requires computing 625 and 10000 inverse problems. The computations are performed by parallel complex K100. MPI technology is applied to control parallel computations.

As a result of approach application four-dimensional arrays are obtained. These arrays contain numerical presentations of the character time  $t^*$  and crucial velocity  $V^*$  dependencies on four determining parameters  $(M_\infty, Re_\infty, Pr_\infty, Sh_\infty)$ . The dependencies  $V^* = V^*(M_\infty, \lg Re_\infty, Pr_\infty)$  and  $t^* = t^*(M_\infty, \lg Re_\infty, Pr_\infty)$  are presented in Fig.3,4 by isosurfaces. The view of these isosurfaces shows that character time and crucial velocity does not depend on Reynolds

number for chosen class of problems in this laminar diapason of  $\text{Re}_\infty$ .

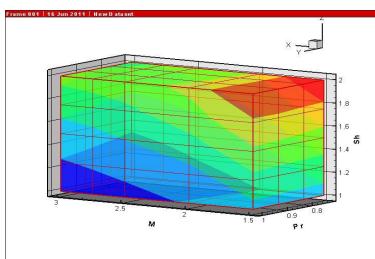


**Figure 3:** Crucial velocity dependence on Mach, Reynolds and Prandtl numbers

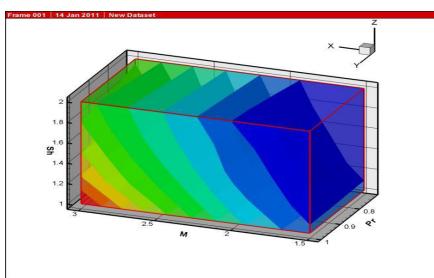


**Figure 4:** Character time dependence on Mach, Reynolds and Prandtl numbers

Further, the dimensionality of array can be decreased and one is able to consider 3D arrays  $V^* = V^*(M_\infty, \text{Pr}_\infty, \text{Sh}_\infty)$  and  $t^* = t^*(M_\infty, \text{Pr}_\infty, \text{Sh}_\infty)$ . These dependencies are shown in Fig.5,6 by isosurfaces.



**Figure 5:** Crucial velocity dependence on Mach, Prandtl and Strouhal numbers



**Figure 6:** Character time dependence on Mach, Prandtl and Strouhal numbers

Analyzing the view of these isosurfaces one can approximate the isosurfaces by planes. For the purpose of rough estimation the sought-for dependence can be written in a form of plane

$$AM_\infty + B\text{Pr}_\infty + C\text{Sh}_\infty = \text{const}.$$

It allows one to get average estimation of  $V^*$  and  $t^*$  dependencies on determining parameters as

$$V^* = V^*(M_\infty, \text{Pr}_\infty, \text{Sh}_\infty) = -0.1M_\infty + 0.115\text{Pr}_\infty + 0.24\text{Sh}_\infty$$

$$t^* = t^*(M_\infty, \text{Pr}_\infty, \text{Sh}_\infty) = 0.224M_\infty - 0.04\text{Pr}_\infty - 0.132\text{Sh}_\infty$$

These expressions describe the connection between control parameter (pressure ratio growth rate) and character parameters – Mach, Reynolds and Strouhal numbers.

The way to check the accuracy of such approach is simple enough. For this purpose one should specify character numbers and the rate of pressure ratio growth and then one should carry out the computation of the straight problem. The resulting space-time structure can be compared with the structure corresponding to approximate expression.

## 5. CONCLUSIONS

The optimization approach to unsteady processes analysis is considered. The approach allows carrying out fast and effective estimation of the way how the crucial points of flow structure transformation depend on determining parameters of the problem. Combining the inverse problems solutions with the visual presentations of such solutions for many real cases allows one to obtain the sought-for dependencies in a quasi-analytical form. The approach can be applied for rough grids. The approach amounts to solving a set of similar small tasks, so it can be applied also for parallel computations. The approach can be applied to a wide range of time-dependent processes for various practical applications.

## 6. REFERENCES

- [1] Bondarev A.E., Galaktionov V.A., Chechetkin V.M. *Analysis of the development concepts and methods of visual data representation in computational physics* // *Computational Mathematics and Mathematical Physics*, 2011, Vol. 51, № 4, p. 624-636.
- [2] Ilyin V.P. *Strategies and tactics of extremal parallelism / Siberian Science.* - 09.02.2012. - N 6. - <http://www.sbras.ru/HBC/article.php?id=622&id=11> (in Russian).
- [3] Sobol' I.M., Kartyshov S.V., Kulchickaya I.A., Levitan Yu.L. *Multicriterial optimization of mathematical models* // *Matematicheskoe modelirovaniye*.-1994.- N 6.- p. 85-93 (in Russian).
- [4] Alifanov O.M. *Inverse Problems for Heat Transfer*. Moscow, Mashinostroenie, 1988 (in Russian).
- [5] Beck J.V., Blackwell B., St.Clair C. *Inverse Heat Conduction. Ill-posed Problems*. John Wiley&Sons, USA, N.Y., 1985.

- [6] Savasere A., Omiecinski E., Navathe S. *An Efficient Algorithm for Mining Association Rules in Large Databases* // Proc. 21st Int'l Conf. Very Large Data Bases, Morgan Kaufmann, San Francisco, 1995, pp. 432-444.
- [7] Suntinger M., Obweger H., Schiefer J., Gröller M. E. *Event Tunnel: Exploring Event-Driven Business Processes* // IEEE Computer Graphics and Applications, Volume 28, Number 5, 2008, pp.46-55.
- [8] Gorban A., Kegl B., Wunsch D., Zinovyev A. (Eds.) *Principal Manifolds for Data Visualization and Dimension Reduction / LNCSE 58*, Springer, Berlin – Heidelberg – New York, 2007.
- [9] Bondarev A.E. *Optimizing Hybrid Finite Difference Scheme with regard to Viscosity and Turbulence on the Base of Inverse Problems* // Proc. of Conf. "High-Performance Computations in Mechanics and Physics", Moscow, 2009. C.39-44 (in Russian).

### About the authors

Dr. Alexander Bondarev is a senior researcher at Keldysh Institute for Applied Mathematics RAS. His contact email is [bond@keldysh.ru](mailto:bond@keldysh.ru)

# Scientific Visualization and FRep Geometric Modeling: a Survey

Alexander Pasko , Valery Adzhiev , Evgeniya Malikova\*, Victor Pilyugin\*

National Centre for Computer Animation, Bournemouth University, Bournemouth ,United Kingdom

\* Scientific visualization laboratory, National Research Nuclear University "MEPhI", Moscow, Russia  
 { apasko, vadzhiev} [@bournemouth.ac.uk](mailto:@bournemouth.ac.uk) , { malikova, pilyugin} [@sv-journal.com](mailto:@sv-journal.com)

## Abstract

A description of the use of geometric modeling in scientific visualization is given as the basis of this brief survey. The Function Representation (FRep) providing models of both geometry and point-wise attributes is particularly considered as suitable for various applications of scientific visualization. We present applications of the described approach in visualization of physical scalar fields, geological and biological structures as well as abstract mathematical models in the form of functions of multiple variables.

## 1. GEOMETRIC MODELING IN SCIENTIFIC VISUALIZATION

Computer visualization of scientific data known as scientific visualization is considered a modern efficient method of data analysis. This method is widely used in different theoretical and experimental researches [1]. Its main idea is to set up a correspondence between the given data being analysed and its static or dynamic graphic interpretation, that is analysed visually, and results of such analysis are interpreted in respect to the given data. Data analysed using scientific visualization can be of different nature. The aims of analysis can be different as well. Accordingly, the corresponding graphic data can be different.

Therefore, the data analysis problem using scientific visualization consists of consequently solving the two following problems:

1) The problem of obtaining graphical images of the data being analysed (problem of initial data visualization). To obtain such an image, a spatial scene (an assembly of spatial objects) with its geometric and visual characteristics has to be put into correspondence with initial data. Then, a graphical image can be generated using some rendering procedure for further visual analysis.

2) The problem of visual analysis of the graphical image of data being analysed. This problem is solved by the user of the visualization system. Observing the obtained graphical image, the researcher can make conclusions regarding the spatial scene, spatial objects within it, their spatial relations, shapes and textures. Then, these conclusions are interpreted in the terms of the application area where the initial data came from. The process of the spatial scene analysis cannot be completely formalised. The efficiency of visual analysis is determined by the experience of the researcher and by their abilities in visual and spatial thinking. If the researcher is not satisfied by the results of visual analysis, they can repeat several steps of the visualization pipeline by tuning its parameters and in particular parameters of the spatial scene. As the result, the process of solving the data analysis problem becomes iterative and interactive.

Obviously, creation and manipulation with the spatial scene plays the key role in solving the above problems. The efficiency of such geometric modeling is determined by the functionality of available software tools.

Russia, Moscow, October 09, 2012

## 2. FREP GEOMETRIC MODELING

In this paper, we give a brief survey of scientific visualization based on the Function Representation of geometric objects [2,3]. The Function Representation (FRep) defines a multidimensional point set (geometric object) by a single real function of multiple variables  $F(X)$ . Three-dimensional, time-varying and other multidimensional objects can be defined this way. In an FRep modeling system, an object is represented internally by a constructive tree data structure reflecting the logic of the object construction, where leaves represent geometric primitives and internal nodes represent operations. The function  $F$  is evaluated at a given point by an FRep tree postorder traversal procedure. The noticeable advantages of this representation are its procedural nature and extensibility, or a possibility to introduce a new primitive or operation through a small analytical expression or a short function evaluation procedure. In [4], a more general constructive hypervolume model was introduced, which supports modelling heterogeneous volumetric objects as point sets with attributes, where an attribute is a mathematical model of an object property of an arbitrary nature such as material, photometric, physical, and others assigned at every point. A hypervolume model represents a heterogeneous object by a real vector-function. More details on heterogeneous objects modeling can be found in [5].

Modeling on the basis of FRep can be performed using a high-level language HyperFun [6] and supporting its software tools. HyperFun is an international free and open source software project on FRep modeling, visualization and animation. The members of the HyperFun team, which is a freely associated group of researchers and students from multiple countries, have contributed to the application case studies described in this paper.

The HyperFun language supports all the basic concepts of FRep for constructing a single defining function for an object of complex geometry and a vector-function for a heterogeneous object. In HyperFun, the procedure for calculating the value of the function at a given point in space can be described using known constructs of the structured programming such as the conditional operator or the cycle operator, as well as special symbols for set-theoretic operations. Modeling of heterogeneous objects in HyperFun is supported by a special vector of attributes. HyperFun has the following universal features appropriate for modeling and visualization:

1. the broad and easily extensible set of geometric primitives and operations;
2. the procedural generation of geometric structures of high complexity;
3. modeling of object properties using the point attributes;
4. the visualization of isosurfaces using their piecewise linear approximation (polygonization) and raytracing;
5. possibility of using multidimensional heterogeneous objects in the visualization pipeline;

6. iterative changes of model parameters in the process of visual analysis.

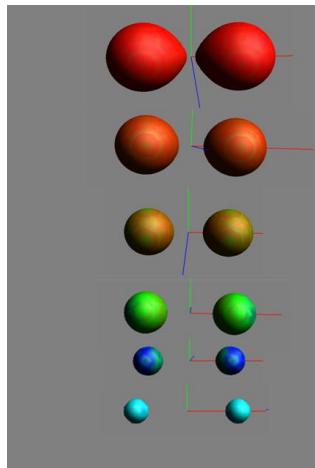
The main rendering method of HyperFun is polygonization of an isosurface followed by the use of modern rendering hardware. Back in 1986, an algorithm of isosurface polygonization was proposed and implemented [7]. It is free from topological ambiguities, which are essential to the well-known algorithm of marching cubes. The trilinear interpolation inside a cubic cell and the bilinear interpolation on a cell face are used for the hyperbolic arcs detection at the cell faces and for the construction of the edges connectivity graph to resolve topological ambiguities.

Because of its advanced level of geometric modeling HyperFun can be an effective tool for scientific visualization. The works on the development of the software product HyperFun were linked to earlier works at the National Research Nuclear University "MEPhI" on the software complex SAGRAF and its application software [8]. These products were widely used in "MEPhI" and other organizations for scientific visualization in experimental nuclear physics, physics of superconductivity and physics of protection of nuclear and physical objects, research of geological and biological structures and others.

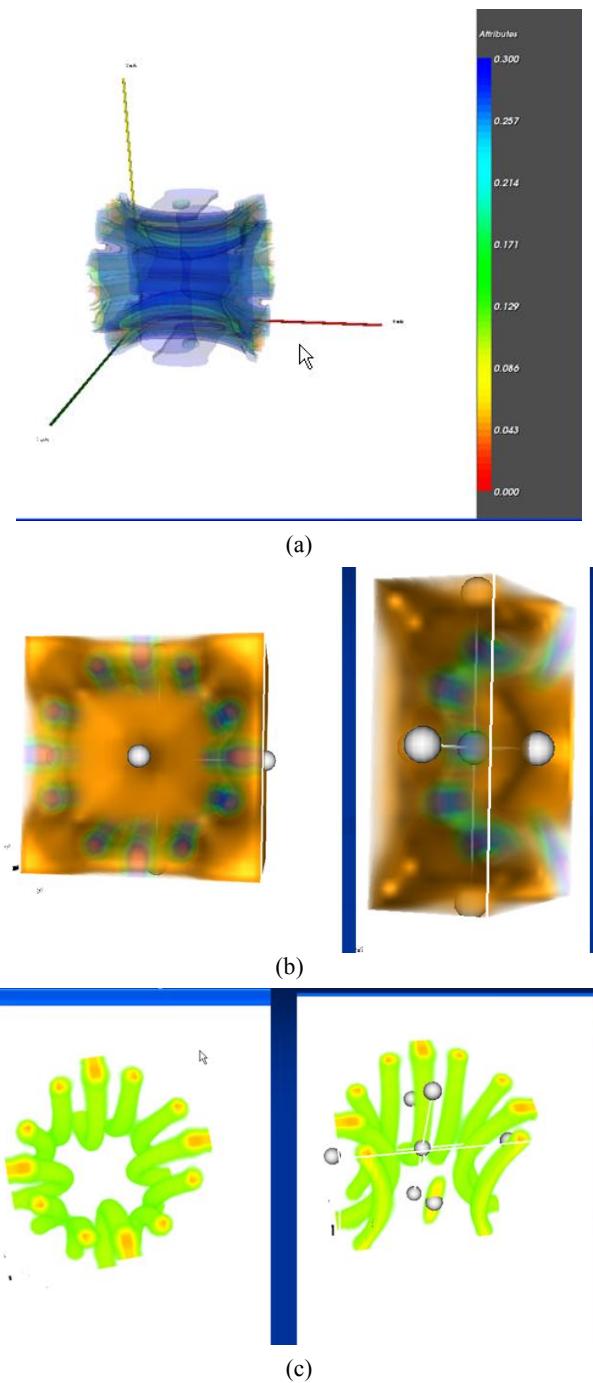
### 3. SCIENTIFIC VISUALIZATION APPLICATIONS BASED ON FREP

Let us present some examples of scientific visualization applications based on FRep and HyperFun software tools.

#### 3.1 Visualization of physical fields



**Figure 1:** The electron density scalar field and electrostatic potential scalar field of the He2 molecule visualization example: a set of isosurfaces of the electron density field colored according to the electrostatic potential field values.



**Figure 2:** The order-parameter field for the second type superconductor (Abrikosov vortices) visualization example. Different approaches to scalar field visualization: isosurfaces (a), volume visualization with the scalar field defined on a rectangular domain (b) and on a complex domain defined by one of the field's isosurfaces (c)

HyperFun was used as a modeling and visualization tool for analysis of physical scalar fields in the "Scientific Visualization" laboratory of the National Research Nuclear University "MEPhI". The studied scalar fields were given as functions of several variables defined on domains represented as geometric objects that also could be defined by functions of several variables. A functional description of a scalar field and its domain was obtained as the result of physical objects computer modeling within students' research work. Such

description of studied physical object was presented in a file in the form of numerical data that should be analyzed.

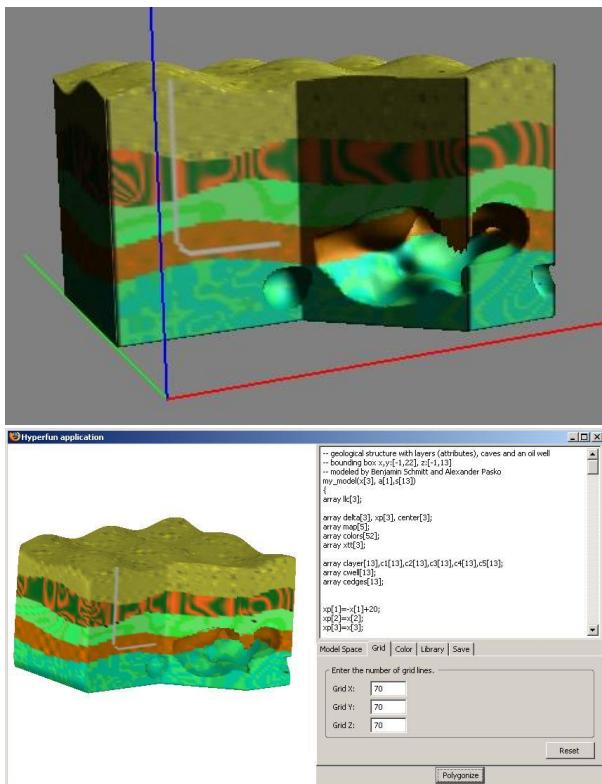
There are two examples of scalar field visualization shown in Fig.1 and Fig.2 :

- two scalar fields visualization: the electron density scalar field and the electrostatic potential scalar field of He2 molecule (Fig.1);
- the second type superconductor scalar field visualization (Fig.2).

To obtain these results several additional file reading based primitives and attribute functions were brought to the HyperFun API and used within the resulting HyperFun model. The visualization presented in Fig. 2 was made through the VTK based interface for HyperFun.

### 3.2 Visualization of geological structures

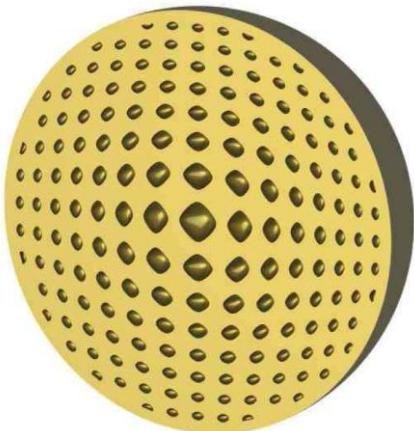
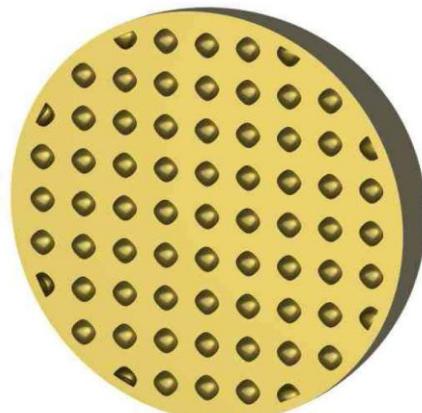
Heterogeneous volumetric objects in geosciences usually consist of multiple layers of different materials with cavities, wells, and other irregularities [9]. We present here a simplified example of an FRep model of such a geological object. There are two components in such a model: the model of the point set geometry and the model of material attributes. Fig. 3a shows a volumetric geological structure with layers of different materials, cavities and an oil well.



**Figure 3:** a) Volumetric geological structure with layers of different materials, cavities and a drilled well;  
b) Hyperfox extension's visualization window with the volumetric geological model.

A Web-based modeling and visualization tool Hyperfox has been implemented within a Firefox Web browser [10]. The user interface for this tool is shown in Figure 3b. The interface allows users to define a model in the HyperFun language, set various parameters for the visualization.

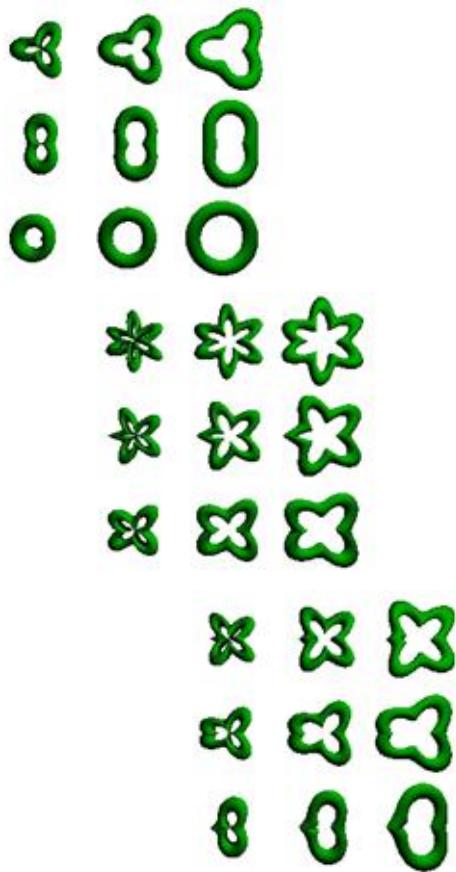
### 3.3 Visualization of biological structures



**Figure 4:** FRep modeling and visualization of a porous bone structure.

Irregular biological structures such as porous bones can be modelled procedurally using FRep as interconnected spherical pores with pseudo-random variations of their sizes and positions [11], see Fig. 4. The pore size is made variable depending on the distance to the surface reflecting the porous structure of bones. Such a model allows for analysis of the dynamics of diseases such as osteoporosis.

### 3.4 Visualization of functions of several variables



**Figure 5:** Visualization of a function of several variables in the form of an animated matrix of isosurfaces (three frames shown).

Visualization results of function of several variables are shown in Fig. 5. For visualization there were used matrices of isosurfaces of functions of three variables and animation was made with changes of one of variables synchronized with physical time [6].

## 4. CONCLUSIONS

This survey is based on the idea that the scientific visualization pipeline necessarily includes the step of introducing geometric models including volumetric and heterogeneous objects and a spatial scene before generating images. We characterize the Function Representation (FRep) of solid and heterogeneous geometric objects and the HyperFun language supporting FRep modeling. Examples of visualization in different research areas are given and illustrated, namely in physical fields simulation, studies of geological and biological structures and analysis of function of several variables. We can conclude that the experience of using FRep shows that it suits well scientific visualization in the research areas involving all kinds of scalar fields and objects with complex geometry and heterogeneous internal structures.

## 4. REFERENCES

1. Pilyugin V.V. , Sumarokov L.N., Frolov K.V., *Machine graphics and automatization of scientific research*.- Vestnik AN USSR.-1985.-Issue. 10.-C.50-58. (in Russian)
2. Pasko A., Adzhiev V., Sourin A., Savchenko V., *Function representation in geometric modeling: concepts, implementation and applications*, The Visual Computer, vol.11, No.8, 1995, pp.429-446.
3. Adzhiev V., Pasko A., Savchenko V., Surin A., *Shape modelling using real functions*.- Open systems. -1996. - #5 (19).- C. 14-18.(in Russian)
4. Pasko A., Adzhiev V., Schmitt B., Schlick C., *Constructive hypervolume modeling*, Graphical Models, vol. 63, No. 6, 2001, pp. 413-442.
5. *Heterogeneous Objects Modelling and Applications, Lecture Notes in Computer Science*, vol. 4889, Eds. Pasko A., Adzhiev V., Comninou P., Springer Berlin / Heidelberg, 2008, 285 p.
6. Adzhiev V., Cartwright R., Fausett E., Ossipov A., Pasko A., Savchenko V., *HyperFun project: a framework for collaborative multidimensional F-rep modeling*, Implicit Surfaces '99, Eurographics/ACM SIGGRAPH Workshop, J. Hughes and C. Schlick (Eds.), 1999, pp. 59-69, [http://www.hyperfun.org/HF\\_CGG2.htm](http://www.hyperfun.org/HF_CGG2.htm)
7. Pasko A. A., Pilyugin V. V., Pokrovskiy V. N., *Using Computer Geometry for Analysis of Functions of Three Variables*, Communications of Joint Institute of Nuclear Research, JINR P10-86-310, Dubna, Russia, 1986, 10 p. (in Russian)  
Publication in English: Geometric modeling in the analysis of trivariate functions, Computers and Graphics, vol.12, Nos. 3/4, 1988, pp. 457-465. <http://hyperfun.org/wiki/doku.php?id=frep:isopol>
8. Adzhiev V. et al., *SAGRAPH Computer Geometry and Graphics Complex and its Usage in Scientific Research*, Report JINR P10-85-116, Dubna, 1985.(in Russian)
9. V. Adzhiev, E. Kartasheva, T. Kunii, A. Pasko, B. Schmitt, *Hybrid cellular-functional modeling of heterogeneous objects*, *Journal of Computing and Information Science in Engineering, Transactions of the ASME*, vol. 2, No. 4, December 2002, pp. 312-322.
10. T. Vilbrandt, O. Fryazinov, C. Stamm, A. Pasko, *A web oriented function-based volume modeling framework*, Technical Report TR-NCCA-2008-02, ISBN 1-85899-123-4, The National Centre for Computer Animation, Bournemouth University, UK, 2008, 12 p.
11. A. Pasko, T. Vilbrandt, O. Fryazinov, P.-A. Fayolle, V. Adzhiev, *Procedural function-based modelling of volumetric microstructures*, *Graphical Models*, Elsevier, vol. 73, No. 5, 2011, pp. 165–181.

# Разработка адаптивного мультиплатформенного визуализатора результатов научных расчётов для высокопроизводительных вычислительных систем

Константин Рябинин

Механико-математический факультет

Пермский государственный национальный исследовательский университет, Пермь, Россия

kostya.ryabinin@gmail.com

## Аннотация

В данной статье рассматривается вопрос создания системы визуализации результатов научных расчётов. Для такой системы предлагаются использовать клиент-серверную архитектуру. Серверная часть может выполняться как на настольном компьютере, так и на высокопроизводительном вычислительном комплексе. Клиентская часть может выполнятся как на настольном компьютере, так и на мобильном устройстве (смартфоне или планшетном компьютере). Процесс визуализации адаптивно распределяется между клиентом и сервером так, чтобы обеспечить оптимальную нагрузку и наибольшую скорость.

**Ключевые слова:** Визуализация высокопроизводительных вычислений, мобильные платформы, OpenGL, VTK, VES.

## 1. ВВЕДЕНИЕ

В настоящее время расчётовые задачи, возникающие в физике, химии, биологии и других естественных науках требуют всё более сложных средств визуализации. Происходит это потому, что усложняются математические модели, используемые в этих задачах, и традиционные средства отображения результатов, такие как графики, диаграммы и таблицы, оказываются недостаточно наглядными. Наглядность же представления результатов вычислений очень важна для исследователя [13].

На сегодняшний день существует большое количество программных пакетов и библиотек, служащих для визуализации результатов научных вычислений. Однако среди них достаточно мало мультиплатформенных решений, которые могли бы работать как под управлением операционных систем для настольных компьютеров (Windows, GNU / Linux, Mac OS X и др.), так и под управлением операционных систем для мобильных устройств (iOS, Android и др.). Традиционно мобильные устройства не рассматриваются как средства, пригодные для проведения сложной визуализации, однако, с ростом их популярности и улучшением их технических характеристик, этот вопрос нуждается в пересмотре.

Кроме того, большинство систем научной визуализации не обладают достаточной интеграцией с самими вычислителями. В то же время для исследователя, который имеет дело с вычислительно-сложной задачей, решаемой на удалённом высокопроизводительном сервере, такая интеграция является очень актуальной. Интеграция позволяет сократить время получения данных и адаптации их к конкретной системе визуализации, а также предоставляет возможность управлять вычислениями и видеть их результат при помощи единого интерфейса.

В контексте использования удалённых вычислительных систем высокой производительности встаёт вопрос об эффективной пересылке данных, подлежащих визуализации, а также балансировке нагрузки клиента и сервера. К решению этого вопроса существует три подхода [12]:

1. Визуализация в полном объёме выполняется на клиенте (клиент получает от сервера данные, подлежащие визуализации).
2. Визуализация в полном объёме выполняется на сервере (клиент получает от сервера готовую картинку).
3. Визуализация распределена между клиентом и сервером.

На сегодняшний день наиболее распространёнными являются первые два подхода. Оба они обладают своими преимуществами, однако имеют и серьёзные недостатки.

Пользуясь первым подходом, можно достичь высокой степени интерактивности визуализации. Однако при этом к клиенту и к каналу связи клиента и сервера предъявляются повышенные требования: необходимо, в общем случае, передавать большие объёмы данных по сети, а затем выполнять построение изображения на локальной машине. В том случае, если в роли локальной машины выступает, например, мобильное устройство невысокой вычислительной мощности, подключённое к сети через низкоскоростное беспроводное соединение, этот подход неприемлем.

При втором подходе вычислительная нагрузка полностью перекладывается на сервер. С одной стороны, это позволяет значительно снизить системные требования для клиента, однако с другой – при обращении к серверу нескольких клиентов сразу нагрузка на сервер может оказаться слишком высокой. Кроме того, хотя для передачи картинки требуется, в общем случае, меньше трафика, чем для передачи данных, подлежащих визуализации, при интерактивном изменении сцены нагрузка на сеть резко возрастёт. В условиях низкоскоростного соединения организация интерактивности и воспроизведение плавной анимации могут оказаться невозможными.

Третий подход способен объединить в себе достоинства первых двух, минимизировав при этом влияние их недостатков. Основная идея заключается в том, чтобы часть визуализации выполнять на сервере, а часть – на клиенте. Средствами высокопроизводительного сервера может быть выполнена отрисовка наиболее сложных частей данных, а так же некоторое упрощение данных, передаваемых клиенту. Разделение нагрузки между клиентом и сервером может быть спланировано на основании быстродействия клиента, занятости сервера (количество других подключенных к нему клиентов) и скорости соединения.

Такой подход, однако, является наиболее сложным и допускает много различных вариантов реализации.

Целью данной работы является разработка системы визуализации результатов научных расчётов, основанной на адаптивном разделении предобработки данных и рендеринга между клиентом и сервером. Сервером для данной системы может выступать как обычный компьютер, так и высокопроизводительный массивно-параллельный вычислительный комплекс. Клиентами могут выступать

как настольные компьютеры под управлением различных операционных систем, так и мобильные устройства (смартфоны и планшетные компьютеры).

Задачами, которые необходимо решить для достижения поставленной цели, являются:

- анализ существующих разработок в области систем научной визуализации;
- анализ методов написания мультиплатформенных приложений;
- проектирование и реализация мультиплатформенного ядра системы визуализации (сервера и клиента);
- разработка средств автоматизации портирования графического интерфейса на различные платформы (платформы для настольных компьютеров и платформы для мобильных устройств);
- тестирование системы на реальных прикладных задачах.

## 2. ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ

Для наглядного представления научных данных исследователями традиционно используются математические программные пакеты, интегрирующие в себе функции построения изображений (такие, как MathCad, MatLab, Mathematica, Maxima и др.), либо пакеты, ориентированные исключительно на визуализацию (TechPlot, Origin, EasyPlot, IRIS Explorer, Surfer, Grapher, AMLab и др.) [11, 14]. Также в ряде случаев возможно использование систем автоматизированного проектирования (САПР).

Однако пакеты, интегрирующие в себе вычислительное ядро и визуализатор, являются недостаточно гибкими для эффективного решения с их помощью произвольных задач в высокопроизводительных вычислительных системах. Многие из них ориентированы лишь на настольные компьютеры и не могут эффективно работать на суперкомпьютерах. Кроме того, возможности визуализации в таких системах, как правило, ограничиваются построением двумерных и трёхмерных графиков.

Пакеты, ориентированные на одну только визуализацию, принимают на вход результаты вычислений в некотором формате и не предполагают автоматизированной коммуникации с программой-решателем. Поэтому пользователь должен самостоятельно осуществлять получение и адаптацию исходных данных.

Ещё более остро проблема адаптации данных стоит в случае использования САПР, так как этот класс программного обеспечения, вообще говоря, служит для решения иных задач.

Помимо этого, общей проблемой всех вышеописанных программных пакетов, является отсутствие для них мобильных версий. Существуют мобильные аналоги систем математических вычислений (PocetCAS, GraphCalc и др.), однако возможности встроенных в них визуализаторов также ограничены лишь построением графиков и поверхностей.

Детальный же анализ магазинов AppStore, Google Play Store и BlackBerry App World показал почти полное отсутствие систем научной визуализации для мобильных устройств под управлением iOS, Android и BlackBerry. Наиболее многофункциональной и производительной системой является KiviViewer [5], реализованная под iOS и Android. Эта система предоставляет возможность визуализировать относительно большие массивы геометрических данных, однако при этом не имеет прямой связи с ЭВМ, на которой производятся вычисления. Пересылку данных и адаптацию

их к формату, пригодному для работы системы визуализации, должен осуществлять пользователь.

На уровне библиотек визуализации существует целый ряд различных решений, таких как OpenDX, VTK, VizIt, ScientificVR и др. Подробный обзор этих программных средств приведён в диссертационной работе О. В. Джосан [12]. Данные библиотеки предоставляют инструментарий для написания систем научной визуализации. Наиболее перспективной разработкой является библиотека VTK [2], имеющая расширение pVTK, которое добавляет функции распараллеливания процесса визуализации. Таким образом, при помощи данной библиотеки может быть осуществлено параллельное построение изображения в высокопроизводительной вычислительной системе.

На основе библиотеки pVTK разработана свободно распространяемая кроссплатформенная система научной визуализации ParaView [3], обладающая графическим интерфейсом пользователя и ориентированная на использование в массивно-параллельных вычислительных системах. ParaView имеет клиент-серверную архитектуру, что позволяет выполнять удалённую визуализацию. Клиентская часть данной системы может работать под управлением большинства операционных систем для настольных компьютеров, однако версии для мобильных устройств на данный момент не существует.

Благодаря широким функциональным возможностям, библиотека pVTK может быть использована в качестве параллельного визуализатора на стороне сервера. Однако на стороне клиента данную библиотеку возможно применить лишь в качестве вспомогательного визуализатора: ориентированность этого средства на настольные компьютеры не позволяет использовать его на мобильных платформах в готовом виде.

Также старые версии pVTK не предоставляли средств для осуществления стереоскопического рендеринга. Учитывая высокую популярность стереоскопии и всё увеличивающееся количество устройств, позволяющих демонстрировать стереоизображения пользователю (стереомониторы, шлемы виртуальной реальности и т. д.), её поддержка необходима и была добавлена в новых версиях библиотеки.

Ещё одним минусом pVTK является отсутствие средств воспроизведения сложной анимации, то есть динамики поведения визуализируемых систем [12]. В то же время наглядное представление динамики с возможностью управлять масштабом времени (интерактивно ускорять и замедлять отображаемый процесс) – очень важная функция программы визуализации для исследователя.

Ранее упомянутая система KiviViewer основана на библиотеке VES [10], которая, в свою очередь, является адаптацией библиотеки VTK для мобильных устройств. Наследуя от VTK широкие возможности в области визуализации научной графики, библиотека VES может быть использована в качестве основы визуализатора на стороне мобильного клиента.

Для визуализации трёхмерных и двумерных сцен могут быть использованы различные модули графического расширения, среди которых существует большое количество кроссплатформенных решений. Примерами таких модулей служат Unity3D, SIO2, OGRE, Irrlicht, oolong, Cocos2D/Cocos3D, libGDX и др. Все перечисленные модули способны работать на мобильных устройствах, однако они ориентированы на создание игровых приложений и не подходят для визуализации научной графики. Игровые приложения отличаются от систем научной визуализации тем, что используют легковесные данные (оптимизированные низкополигональные модели персонажей и окружения), но при этом ориентированы на

высокую динамичность сцены и обилие визуальных спецэффектов. Приложения научной графики, напротив, принимают на вход большие объёмы данных и сложные структуры, визуализация которых осуществляется с использованием специфических алгоритмов (построение сечений, проекций, представление структур в разных масштабах и разных уровнях детализации и т. д.). Сложные визуальные эффекты в данном случае не так важны, а зачастую могут только помешать, отвлекая внимание исследователя. Таким образом, систему научной визуализации не следует основывать на модуле графического расширения, ориентированном на игровые приложения.

При создании интерактивной системы визуализации важную роль играет не только подсистема рендеринга, но и пользовательский интерфейс. Для организации кроссплатформенности, графический интерфейс следует разрабатывать с использованием какой-либо библиотеки, реализованной под все целевые платформы.

На сегодняшний день существует относительно немного кроссплатформенных библиотек для создания графических интерфейсов пользователя. Наиболее популярными являются Qt, GTK, Tk и Awt / Swing. GTK и Tk имеют реализацию только под операционные системы для настольных компьютеров. Awt и Swing используются для создания интерфейсов Java-приложений, однако тоже только для Java-машин, работающих на настольных компьютерах. Для Qt существуют реализации под операционные системы iOS (Qt-iPhone [9]) и Android (Necessitas [7]). Однако эти реализации находятся ещё в стадии разработки и отладки, а потому могут испытывать проблемы стабильности работы.

Кроме того, используя Qt в качестве библиотеки для создания приложения, которое работало бы под управлением операционных систем для настольных компьютеров и для мобильных устройств, программист сталкивается с проблемой различного дизайна интерфейса под различные платформы. Дело в том, что мобильные устройства и настольные компьютеры используют различные парадигмы управления и отображения данных. Так, например, основными средствами контроля над программой на настольном компьютере являются клавиатура и мышь, а на мобильном устройстве – сенсорный экран и управление при помощи жестов. Также различия заключаются и в дисплеях: экран мобильного устройства, как правило, значительно меньше, чем экран компьютера, а значит и информация на нём должна располагаться иным образом, чтобы обеспечить требуемую эргономику. Поэтому, даже используя какую-либо из существующих кроссплатформенных библиотек, программисту придётся создавать дизайн приложения дважды – под разные типы устройств.

Ещё один способ достижения кроссплатформенности – написание Web-приложений. Наиболее популярными на сегодняшний день технологиями в этой области являются HTML 5 (совместно с использованием JavaScript и CSS) и Flash. Обе эти технологии предоставляют программисту широкие возможности, однако ввиду того, что итоговое приложение выполняется в браузере (то есть в некоторой программной прослойке), возможен целый ряд проблем с производительностью и стабильностью работы. В настоящее время поддержка этих технологий со стороны браузеров активно развивается, вводятся различные оптимизации, и, соответственно, растёт сложность самих Web-приложений. Но в контексте научной графики, где имеют место большие объёмы данных, производительности Web-приложений пока ещё не достаточно. В особенности проблема производительности встаёт на мобильных

устройствах, так как их вычислительная мощность значительно ниже, чем у стационарных компьютеров. Из-за этого и поддержка Web-технологий для мобильных устройств оказывается в значительной степени сокращённой.

Несмотря на это, существует отдельный достаточно перспективный проект MoSync [6], предоставляющий программисту возможность создавать интерфейс мобильного приложения на HTML 5, а логику реализовывать на языке C++. Получаемое приложение может выполняться под управлением любой из наиболее популярных операционных систем для мобильных устройств. При этом, за счёт использования кода на C++, может быть достигнуто высокое быстродействие. Однако в контексте решения задач научной графики критичным является так же быстрая визуализация на уровне интерфейса. Таким образом, при использовании фреймворка MoSync (либо иных основанных на HTML 5 или Flash фреймворков), интерфейс, выполняющийся в браузере, станет узким местом. Кроме того, MoSync ориентирован только на мобильные устройства, тогда как для достижения мультиплатформенности необходима поддержка и настольных компьютеров.

Для эффективного рендеринга на мобильных устройствах под управлением таких операционных систем, как iOS и Android, используются библиотеки стандарта OpenGL ES [8] – адаптации стандарта OpenGL к мобильным платформам. По своей функциональности OpenGL ES предназначен для отображения трёхмерной графики, но с его помощью может быть создан и обычный двумерный интерфейс пользователя. Однако этот стандарт описывает лишь низкоуровневые функции работы с графикой, из-за чего создание пользовательского интерфейса, как и любых других сложных графических композиций, на его основе трудоёмко. Возникает необходимость разработки программных прослоек, называемых модулями графического расширения. Такие прослойки покрывают функциональность некоторой низкоуровневой графической библиотеки, предоставляя программисту высокоуровневое API. Чаще всего, модули графического расширения разрабатываются в объектно-ориентированной парадигме, так как она хорошо сочетается с терминами компьютерной графики – представление графической сцены в виде множества объектов, свойства и поведение которых объединены в некоторой иерархии классов.

Так, например, существует модуль графического расширения Clutter [1], служащий для создания интерфейсов на основе OpenGL ES. Однако этот модуль не является в достаточной степени распространённым (он используется в операционной системе MeeGo), и его работоспособность под управлением наиболее популярных платформ ограничена.

### 3. ПРЕДЛАГАЕМОЕ РЕШЕНИЕ

На основе проведённого теоретического исследования можно сделать два важных вывода:

1. Не было создано ещё ни одной мультиплатформенной (работающей на мобильных устройствах и на настольных компьютерах одновременно) системы научной визуализации, которая бы получила широкое распространение.
2. На сегодняшний день не существует эффективного решения задачи написания мультиплатформенного приложения, которое работало бы и на мобильных устройствах и на стационарных компьютерах.

Продолжая анализ, можно утверждать, что ситуация, описанная в выводе (1) частично обусловлена проблемой, сформулированной в выводе (2). Таким образом, создание мультиплатформенной системы научной визуализации следует начать с решения проблемы достижения мультиплатформенности вообще.

Анализ существующих технологий показал, что логика мультиплатформенного приложения может быть реализована на языке C++, так как код, написанный на этом языке, так или иначе может выполняться как на мобильных платформах iOS и Android, так и под управлением операционных систем для настольных компьютеров (Windows, GNU / Linux, Mac OS X и др.). В операционных системах для настольных компьютеров, а так же в iOS, есть возможность выполнения кода, написанного на C++, напрямую. В операционной системе Android для этого используется технология JNI [4], позволяющая вызывать методы, написанные на C++, из кода, написанного на Java, и наоборот.

Визуализация данных (трёхмерных структур, двумерных графиков и диаграмм разных типов) может быть осуществлена с использованием открытых библиотек VTK и VES. Эти библиотеки эквивалентны на уровне интерфейса, поэтому переключение между ними не потребует написания большого количества дополнительного кода. Для клиентов системы визуализации, работающих на мобильных устройствах, будет использована библиотека VES, а для настольных клиентов – библиотека VTK.

Однако при использовании этих библиотек должна быть решена проблема организации сложной анимации (для наглядного изображения динамики системы) и проблема поддержки стереоскопического рендеринга.

Сложная анимация может быть достигнута интерполяцией входных данных между некоторыми заранее выделенными состояниями (ключевыми кадрами). Такие состояния могут быть отмечены программой-решателем, которая решает исходную вычислительную задачу на стороне сервера.

Новые версии VTK поддерживают стереоскопическую визуализацию в разных форматах, однако следует решить вопрос совместимости с разными типами устройств вывода графической информации (стереомониторы, шлемы виртуальной реальности и т. д.).

В качестве решения проблемы унифицированного создания графического интерфейса пользователя был разработан модуль графического расширения, основанный на стандарте OpenGL ES. Так как OpenGL ES является подмножеством OpenGL, модуль работоспособен под управлением операционных систем как для мобильных устройств, так и для настольных компьютеров.

Разработанный модуль предоставляет возможность быстро создавать элементы двумерного и трёхмерного интерфейса (использовать пространственную анимацию и объёмные модели элементов интерфейса), предоставляя программисту удобную объектную модель, полностью скрывающую вызовы функций библиотеки стандарта OpenGL ES. Языком реализации модуля является C++. Модуль был протестирован на операционных системах iOS, Android, Windows, GNU / Linux и Mac OS X.

Однако сам по себе фреймворк создания произвольных интерфейсов, как это было показано ранее, ещё не обеспечивает лёгкого достижения мультиплатформенности. Проблема заключается в необходимости создания разного дизайна приложений для мобильных устройств и настольных компьютеров.

Для решения этой проблемы предлагается разработать язык декларативного описания интерфейсов, который позволил

бы описать лишь наиболее общие черты интерфейса. Затем, интерпретатор этого языка создал бы интерфейс, отвечающий декларированным чертам, но с учётом особенностей текущей платформы. Идеяно такой язык схож с языком XAML от компании Microsoft, однако он должен предоставлять более общие описания, чтобы по одной и той же декларации мог быть построен интерфейс как для мобильной, так и для настольной версии приложения.

Системой научной визуализации должны поддерживаться многомасштабность по размеру отображаемой структуры и по времени протекания в этой структуре изучаемых процессов.

Для достижения многомасштабности по размеру, предлагается использовать метафору микроскопа. В программе выделяются два способа изменения масштаба сцены – количественный и качественный.

Количественное масштабирование непрерывно и позволяет увеличивать или уменьшать текущую структуру в определённых пределах. Такое масштабирование метафорически соответствует подстройке фокуса у микроскопа. При этом с изменением коэффициента увеличения меняется уровень детализации сцены: когда части структуры становятся меньше определённого порогового значения, их детализация уменьшается. Оказавшиеся за пределами усечённой пирамиды видимости фрагменты структуры временно исключаются со сцены и не поступают на графический конвейер. Таким образом достигается оптимальная нагрузка на центральный и графический процессоры и повышается производительность.

Качественное масштабирование изменяется дискретно и приводит, вообще говоря, к полной перестройке сцены. Переход от одного масштаба к другому метафорически соответствует смене линзы у микроскопа. Разные уровни масштаба должны иметь поддержку также на стороне программы-решателя, так как именно от неё система визуализации получает данные о структуре отображаемого объекта. Не все программы-решатели обязаны поддерживать многомасштабность обсчитываемой структуры, однако в том случае, если такая поддержка имеется, для исследователя очень актуально иметь возможность быстро переключаться между масштабами в системе визуализации.

Многомасштабность по времени позволяет динамически настраивать скорость воспроизведения анимации структуры. Разные процессы, которые могут быть интересны для исследователя, могут иметь самую разную реальную скорость протекания – от долей секунды (если, например, речь идёт о моделировании квантовых взаимодействий) до миллионов лет (если речь идёт о моделировании планетарной системы). Более того, в одной системе могут протекать сразу несколько процессов с разной, возможно, переменной, скоростью. Для исследователя актуально иметь возможность быстро, во время работы визуализатора, изменять скорость воспроизведения анимации того или иного процесса, ставить воспроизведение на паузу или возобновлять, а также быстро проматывать анимацию к некоторой временной позиции. В связи с этим на интерфейс системы должна быть вынесена линия времени, п редоставляющая возможность перемотки и настройки скорости воспроизведения анимации.

Следующей важной задачей является управление программой-решателем непосредственно с интерфейса системы визуализации. Сложность добавления такой функциональности заключается в том, что для каждой конкретной задачи, вообще говоря, существует свой

решатель, имеющий свой набор команд управления. Очень часто под конкретную научную вычислительную задачу пишется отдельная программа, не имеющая почти никакого собственного интерфейса, и лишь принимающая на вход некоторые исходные данные. В этом случае система визуализации должна позволять формировать необходимые исходные данные и давать команду на запуск либо остановку расчётов.

Для каждого конкретного решателя предлагается создавать описание интерфейса управления (формы ввода данных и, если это необходимо, дополнительных команд, которые могут быть переданы решателю прямо в процессе счёта). Описание осуществляется на том же декларативном языке, на котором создаётся интерфейс всей системы визуализации. Далее файлы с описаниями интерфейсов различных решателей, а так же с необходимой мета-информацией (IP-адрес сервера, на котором находится решатель, параметры подключения и т. д.) размещаются на некотором Web-хостинге, откуда по запросу скачиваются системой визуализации. После того, как файл с описанием интерфейса решателя скачан, система может построить форму управления и отдать решателю команду начать вычисления по данным, которые ввёл в эту форму пользователь.

В качестве посредника между клиентской частью системы визуализации и программой-решателем выступает серверная часть системы. Для того, чтобы сервер мог транслировать команды от клиента решателю, последний должен поддерживать некоторый стандартный интерфейс управления. В качестве такого интерфейса предлагается использовать подход передачи управляющих команд в виде ключей, с которыми запускается решатель. Такой подход позволяет сделать решатель максимально изолированным от системы, и не принуждает создателей решателя загромождать его код какими-либо механизмами взаимодействия. В этом случае, серверу также необходимо иметь описание команд решателя, ставящее в соответствие запросы, приходящие от клиента, ключам решателя. Такое описание легко может быть составлено вручную на языке XML.

Более сложным является управление решателем в процессе его работы. В том случае, если это необходимо, наиболее простым вариантом будет использование механизма межпроцессного взаимодействия через сигналы. Такой подход также не потребует от создателей решателя большого количества дополнительного кода.

По мере вычисления, решатель формирует некоторые выходные данные, которые должны быть визуализированы. Предлагается осуществить разделение процесса визуализации между клиентом и сервером. Для визуализации на стороне сервера предлагается использовать библиотеку pVTK (параллельный вариант библиотеки VTK). При помощи этой библиотеки можно осуществить эффективный рендеринг на высокопроизводительной вычислительной системе.

Разделение рендеринга должно осуществляться эвристически, на основе данных о вычислительной мощности клиента, скорости соединения и загруженности сервера. Кроме того, во внимание необходимо принимать характер визуализируемых данных. На стороне сервера имеет смысл осуществлять визуализацию в первую очередь тех данных, которые не будут изменяться в процессе интерактивного взаимодействия пользователя с графической сценой на клиенте. Например, если итогом визуализации должна являться некоторая трёхмерная модель с наложенной на неё текстурой, причём текстура также представляет собой результат визуализации, то эту текстуру имеет смысл подготовить на сервере.

Задача разделения рендеринга полностью ложится на сервер системы визуализации. Решив её, он выполняет свою часть визуализации и отправляет часть «сырых» (подлежащих визуализации) данных вместе с результатом своей визуализации клиенту.

«Сырые» данные в общем случае так же должны проходить некоторую предобработку на сервере. Как правило, объём их слишком велик, и передавать их по сети слишком затратно, равно как и обрабатывать на стороне клиента. Поэтому в обязанности сервера входит выполнение упрощения набора данных, полученных от программы-решателя. Коэффициент упрощения также должен определяться эвристически, исходя из вычислительной мощности клиента и скорости соединения.

Настройку качества визуализации необходимо вынести на графический интерфейс клиента, чтобы пользователь мог, при желании, полностью отключить упрощение данных, так как в ряде случаев из-за упрощения могут теряться существенные для исследователя детали.

## 4. ЗАКЛЮЧЕНИЕ

На данный момент полностью реализованным является фреймворк для создания интерфейсов, базирующийся на стандарте OpenGL (и совместимый с со стандартом OpenGL). Этот фреймворк позволяет создавать все базовые элементы графического интерфейса пользователя и был протестирован под управлением операционных систем iOS, Android, Windows, GNU / Linux и Mac OS X. Остальные части системы находятся на этапе проектирования или реализованы частично.

Также изучена возможность использования библиотек VTK, pVTK и VES для научной визуализации.

В качестве прикладной задачи для отладки системы планируется рассмотреть расчёт затухания звуковой волны в шумоподавляющем пористом материале. Для решения данной задачи уже имеется математическая модель и программа-решатель [15].

## 5. ССЫЛКИ

- [1] *Clutter project.* <http://www.clutter-project.org/>
- [2] James Ahrens, Charles Law, Will Schroeder, Ken Martin, Michael Papka. *Parallel processing with VTK. Los Alamos National Laboratory - Technical Report#LAUR-00-1620.* 2000.
- [3] James Ahrens, Berk Geveci, Charles Law. *ParaView: An End-User Tool for Large Data Visualization.* Edited by C.D. Hansen, C.R. Johnson. Elsevier. 2005.
- [4] *Java Native Interface.* <http://java.sun.com/docs/books/jni/>.
- [5] *KiwiViewer project.* <http://www.kiwiviewer.org/>.
- [6] *MoSync project.* <http://www.mosync.com/what-is-mosync>.
- [7] *Necessitas project.* <http://sourceforge.net/p/necessitas/home/necessitas/>.
- [8] *OpenGL ES.* <http://www.khronos.org/opengles/>.
- [9] *Qt-iPhone project.* <http://www.qt-iphone.com/Introduction.html>.
- [10] *VES library.* <http://www.vtk.org/Wiki/VES>.
- [11] Максим Виноградов. *Современные средства визуализации и обработки двумерных научных данных.* Московский государственный технический университет имени Н.Э. Баумана. 2002. [http://www.amlab.ru/paper\\_max.shtml](http://www.amlab.ru/paper_max.shtml).
- [12] Оксана Джосан. *Исследование и разработка методов и программных средств визуализации результатов научных вычислений для массивно-параллельных вычислительных систем.* Московский государственный университет имени М.В. Ломоносова. 2009.

[13] Научная визуализация. Virtual Environment Group.  
[http://www.ve-group.ru/vr13\\_127.html](http://www.ve-group.ru/vr13_127.html).

[14] Е. Л. Карташева, Г .А. Багдасаров, А.С. Болдарев, И.В.Гасилова, С. В. Дьяченко, О .Г. Ольховская, В. А . Шмыров, С.Н. Болдырев, В.А. Гасилов. Визуализация данных вычислительных экспериментов в области 3D моделирования излучающей плазмы, выполняемых на многопроцессорных вычислительных системах с помощью пакета Marple. Институт математического моделирования РАН.  
<http://sv-journal.com/2010-1/01/index.html>.

[15] Александр Синер. Методика выбора звукопоглощающих конструкций для турбомашин на основе математического моделирования. Пермский государственный университет. 2010.

### Об авторах

Константин Рябинин – аспирант кафедры математического обеспечения вычислительных систем ПГНИУ.

Его адрес: [kostya.ryabinin@gmail.com](mailto:kostya.ryabinin@gmail.com).

# Визуализация потоков и обработка изображений: сближение и взаимодействие экспериментальных и численных данных

И.А. Знаменская

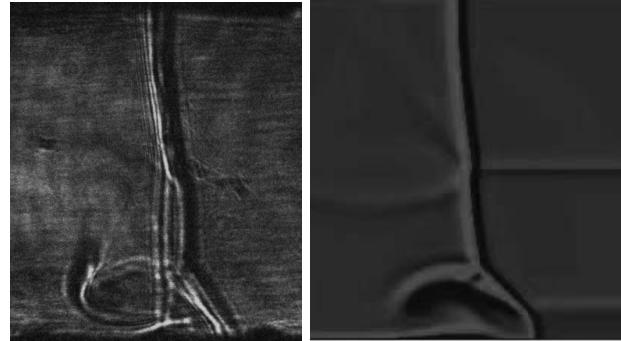
Московский государственный университет, физический факультет, Россия  
[znamen@phys.msu.ru](mailto:znamen@phys.msu.ru)

Методы визуализации потоков используются для исследований, анализа, обработки, представления качественных и количественных данных о потоках в механике: в газах (газо-аэродинамика, теплотехника), жидкостях (гидродинамика, теплофизика, мониторинг гидросферы), многофазных средах (гидравлика, нефтегазовая сфера, горение). Практически все методы экспериментальной визуализации потоков опираются на физические явления (излучение электромагнитных волн, их рассеяние, поглощение, отражение, рефракция, интерференция, люминесценция и.т.д.).

Цель визуализации потоков - качественное и количественное исследование, описание новых явлений и закономерностей в потоке, а также в их демонстрации. Для обработки и анализа полученных изображений потоков используется компьютерная техника, специализированное программное обеспечение. При этом ставятся сходные базовые задачи: уменьшение зашумленности исходного изображения, выделение элементов структуры исследуемых объектов, сохранение полученных результатов в удобном для дальнейшей работы и представления виде. В последнее десятилетие в связи с бурным развитием цифровых технологий визуализация процессов в динамических системах стала существенно доступнее.

Визуализация данных численных расчетов потоков - наглядное представление больших массивов числовой и другой информации, которое реализуется благодаря использованию компьютерной графики. Характерная особенность современного этапа визуализации потоков - сближение численных и экспериментальных изображений потоков. Верификация моделей и алгоритмов численных расчетов проводится на основе данных экспериментальной визуализации потоков. С другой стороны, результаты эксперимента расшифровываются и уточняются на основе данных численного анализа.

Дан обзор и анализ ряда современных методов экспериментальной визуализации: теневые методы, интерферометрия, PIV, LIF, ЛДА, термография, теневой фоновый метод, бароиндикаторные покрытия и др. На конкретных примерах иллюстрируются возможности, предоставляемые сравнением результатов эксперимента и расчета в сходных представлениях.



## Об авторе

Знаменская Ирина Александровна – д.ф.-м.н., профессор, руководитель лаборатории плазменной газодинамики кафедры молекулярной физики МГУ  
E-mail: [znamen@phys.msu.ru](mailto:znamen@phys.msu.ru)

# Модификация воксельного метода для построения единой триангуляционной оболочки объекта по множеству видов

В.А.Бобков, А.П.Кудряшов

Институт автоматики и процессов управления ДВО РАН

Владивосток, Россия

{bobkov, kudryashova}@dvo.ru

## Аннотация

Описан алгоритм построения единой триангуляционной трехмерной текстурированной модели сцены по множественному набору пространственных данных, основанный на воксельном подходе. Предложена оптимизированная структура исходных данных, обеспечивающая повышение производительности алгоритма. Реализована вычислительная схема с организацией параллельных вычислений на графических процессорах средствами языка CUDA.

**Ключевые слова:** воксельное пространство, трехмерная реконструкция, триангуляционная оболочка, текстурирование, маркирующие кубики, CUDA, GPU.

## 1. ВВЕДЕНИЕ

Один из подходов к созданию трехмерных компьютерных моделей реальных объектов и сцен основывается на использовании дальностных данных, получаемых с помощью лазерных дальномеров или методами компьютерного зрения по фотоизображениям. В литературе на такие данные ссылаются как на дальностные изображения (rangeimages) и карты глубин (depthmaps). В отличие от обычных изображений с каждым пикселям связывается расстояние до видимой точки объекта в пространстве сцены. Эти пространственные данные могут быть представлены в виде облака точек в трехмерном пространстве или в виде триангуляционной поверхности. Технологии получения дальностных данных и генерации цифровых пространственных моделей на практике применяются в робототехнике, геоинформатике, производстве, медицине и в др. приложениях. Нетривиальной задачей при построении трехмерных моделей по последовательности дальностных изображений, представленных в виде отдельных триангуляционных сеток (оболочек), является задача объединения этих триангуляционных сеток. Ее решение подразумевает построение единой связной оболочки, отвечающей исходным данным, без дублирования участков поверхности, видимых с нескольких видов. Важным требованием к алгоритмическому решению задачи является высокая скорость обработки данных, поскольку для насыщенных сцен со сложными объектами может потребоваться много видов с большим количеством описывающих треугольников (порядка  $10^4$ - $10^6$ ).

Очевидный путь к решению указанной задачи состоит в сопряжении исходных сеток с устранением избыточного триангуляционного описания посредством совместного анализа данных в трехмерном пространстве сцены и в двумерном пространствах изображений видов. Подобная схема решения была реализована в [4]. Однако она, как отмечают авторы, полностью не исключает дублирования. Более интересным представляется воксельный подход, основанный на использовании воксельной структуры пространства сцены. Эффективная реализация такого подхода была предложена в работе [1]. Основным

преимуществом предложенного решения является сведение исходной задачи к известной задаче построения изоповерхности в скалярном поле. Значительным недостатком данного метода можно его считать высокую ресурсоемкость, из-за чего реконструкция даже небольшой сцены по нескольким десяткам изображениям занимает много часов [2]. Однако, в последнее время, с появлением технологий использующих для вычислений графический процессор (GPU), таких как CUDA (*Compute Unified Device Architecture*) от компании NVIDIA, стало возможным увеличить скорость расчёта ресурсоемких алгоритмов в десятки и сотни раз. В работе [5] можно ознакомиться и с некоторым систематизированным обзором других методов реконструкции поверхности по плотному набору дальностных данных.

Представленный в настоящей статье метод построения единой сеточной модели основывается на подходе, предложенном в [1]. Работа выполнялась в контексте решения авторами более общей задачи по созданию трехмерных моделей на множественных последовательностях изображений с построением карт глубин. Вклад авторов состоит в предлагаемой оптимизации вычислений за счет эффективной структурной организации данных и реализации параллельных вычислений на графических процессорах, что в целом обеспечивает более высокую скорость обработки данных.

## 2. ОПИСАНИЕ МЕТОДА

Задача объединения поверхностей формулируется следующим образом. Имеются  $n$  видов, для каждого из которых на предварительном этапе была построена видимая поверхность в виде пространственной триангуляционной сетки. Триангуляционная сетка строится по плоской триангуляции на регулярной пиксельной решетке изображения с использованием карты глубин, которая вычисляется для точек, видимых на данном виде. Задача заключается в построении единой текстурированной триангуляционной оболочки, которая и будет являться трехмерной моделью сцены.

Предлагаемый метод работает в воксельном пространстве сцены. Используется непрерывная неявная функция  $D(V_i)$ , представленная значениями в узлах воксельной решетки. Функция конструируется как взвешенная сумма получаемых для  $n$  видов расстояний  $d_1(V_i), d_2(V_i), \dots, d_n(V_i)$  от точки  $V_i$  до ближайшей поверхности (рис. 1). Расстояние  $d_j$  берется на луче, направленном из центра проекций  $O_i$  в точку  $V_i$  и является величиной со знаком (положительное для точек, находящихся перед поверхностью, и отрицательное для точек за поверхностью).

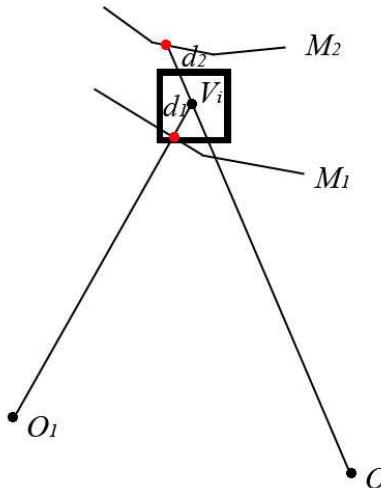


Рис. 1 Построение скалярной функции.

В качестве весовой функции  $W(V_i)$  берется скалярное произведение вектора нормали к поверхности и вектора направления на точку наблюдения. Такой выбор исходит из предположения, что при корреляционном подходе к определению расстояния до поверхности степень неопределенности для наклонных поверхностей выше, чем для поверхностей, наблюдаемых под прямым углом.

Весовая функция  $W(V_i)$  – константа для всех вокселов до видимой поверхности (для данного вида), а для вокселов за поверхностью линейно убывает до нуля в пределах  $\epsilon$ -окрестности. Такой выбор области определения весовой функции направлен на предотвращение возникновения ложных поверхностей.

В результате последовательной обработки всех видов каждому вокселю присваивается значение интегральной функции расстояния  $D(V_i)$  (величина со знаком). Тогда построение изоповерхности  $D(V_i) = 0$  и является решением поставленной задачи. Построение изоповерхности скалярного поля, заданного на воксельной решетке, выполняется с помощью известного алгоритма марширующих кубиков [3].

### 3. МОДИФИКАЦИЯ МЕТОДА

Предполагается, что для каждого из видов определены параметры камеры и построены триангуляционные сетки. Вначале создается воксельное пространство сцены, размерность которого задается пользователем. Эта размерность определяет детализацию результирующей трехмерной сцены.

Наиболее ресурсоемкой частью представленного выше метода является определение расстояний от вокселов до ближайшей поверхности (рис.1.) при формировании скалярного поля. Вычислительная трудоемкость возникает из-за необходимости перебора всех треугольников с вычислением соответствующих расстояний. Чтобы избежать избыточных вычислений предлагается механизм с предварительной индексацией всех треугольников сетки каждого вида (рис. 2). Для этого используется целочисленный массив с размерностью равной размеру исходного изображения.

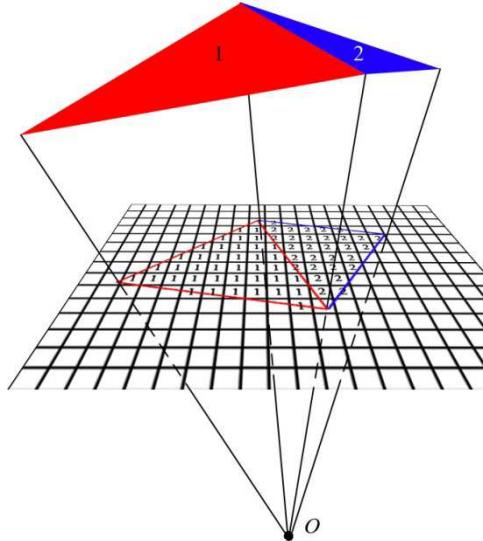


Рис. 2. Индексация треугольников.

Для каждого  $i$ -го вида строится проекционная матрица по формуле:

$$H_i = \begin{pmatrix} c_{11}^i f & c_{12}^i f & c_{13}^i \\ c_{21}^i f & c_{22}^i f & c_{23}^i \\ c_{31}^i f & c_{32}^i f & c_{33}^i \\ c_{41}^i f & c_{42}^i f & c_{43}^i \end{pmatrix}, \quad (1)$$

где  $c_{kl}^i$  – элементы матрицы камеры  $i$ -го вида, а  $f$  – фокус.

Тогда можно спроектировать каждый треугольник исходной сетки  $i$ -го вида на  $i$ -е изображение по формуле:

$$P_{tex}^i = H_i \times P, \quad (2)$$

где  $P$  – координаты точки в трехмерном пространстве.

В результате каждый элемент массива индекса приобретает индекс, указывающий на принадлежность к определенному треугольнику. Такая процедура возможна, поскольку исходная сетка для каждого вида является заведомо «видимой», т.е. не содержит невидимых на этом виде треугольников. Все непроиндексированные пиксели приравниваются к одному значению. Использование индексов при последующей работе алгоритма позволяет избежать полного перебора по всем треугольникам каждого вида и существенно уменьшает время работы алгоритма. При реализации метода был разработан дополнительный алгоритм с использованием альфа-канала, который необходим для отделения реконструируемого объекта от фона. Для каждого пикселя проиндексированного изображения проверяется его значение в альфа-канале исходного изображения. Если это значение равно нулю (т.е. пиксель является фоном), то этому пикселью присваивается то же значение, что и для непроиндексированных пикселей, что исключает построение поверхностей, относящихся к фону.

На последнем этапе, после формирования скалярного поля с помощью алгоритма марширующих кубиков выполняется построение изоповерхности  $D(V_i) = 0$ , которая и является решением построения общей триангуляционной сетки по множеству видов.

#### 4. РЕАЛИЗАЦИЯ НА CUDA

Поскольку метод работает с большими объемами данных, была реализована схема центральный процессор + графические процессоры, где наиболее ресурсоемкие вычисления выполняются на графической плате. Самая ресурсозатратная часть метода – это проход по всем вокселям сцены с вычислением значений скалярной функции  $D(V)$ . Поскольку вычисления этих значений для каждого вокселя являются независимыми, алгоритм хорошо распараллеливается с помощью технологии CUDA. Алгоритм марширующих кубиков также был оптимизирован для работы с помощью GPU. Основной сложностью в реализации этого метода является неизвестное количество получаемых треугольников. Некоторые реализации этого метода, например в OpenCV, предлагают использовать количество памяти, заведомо большее, чем необходимо для хранения полученных треугольников. В предлагаемом методе используется двукратный запуск алгоритма, первый из которых определяет количество треугольников и заводит под них память, а второй запуск уже заполняет ее. Данный подход позволяет избегать перезаполнения видеопамяти при обработки сцен больших размерностей.

#### 5. ТЕКСТУРИРОВАНИЕ

Текстурирование полученной модели производится с использованием всех текстур исходных видов. Каждый треугольник полученной трехмерной модели проецируется на каждое изображение исходных видов и принимается в качестве текстуры, если соответствует следующим критериям: видимость (треугольник виден на изображении), наблюдаемость (треугольник не заслоняется другими треугольниками модели), имеет максимальную площадь проекции.

#### 6. РЕЗУЛЬТАТЫ

Для получения сравнительных оценок эффективности предложенного метода были проведены вычислительные эксперименты на модельных сценах 1 и 2 (рис. 3, 5) и реальной сцене 3 «Temple» (рис. 7) (<http://vision.middlebury.edu/mview/>) [5]. Параметры используемого вычислительного оборудования: процессор IntelCoreDuo 2,6 ГГц, графическая плата NVidiaGeForce 9600GT. Во-первых, сравнивались производительности двух алгоритмических реализаций на центральном процессоре – предложенной в настоящей работе и реализации, аналогичной описанной в [1]. Во-вторых, оценивалось ускорение вычислений предложенного алгоритма за счет реализации параллельных вычислений на графической плате. Результаты вычислительных экспериментов приведены в табл.1 и на рис.4, 6, 8. Как видно из таблицы время выполнения практически не зависит от сложности сцены (количество треугольников), что подтверждает эффективность реализованного механизма индексации. Предложенный механизм индексации при реализации алгоритма на центральном процессоре ускоряет вычисления в 12-20 раз. Использование графических процессоров дает дополнительное ускорение в 7-15 раз. Таким образом, суммарное ускорение оценивается в 200-300 раз.

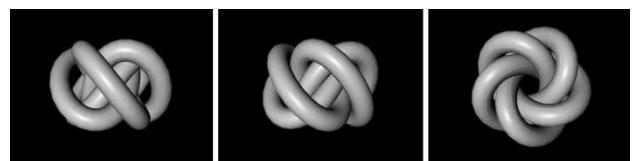


Рис. 3. Исходные виды (3 из 6) сцены 1.

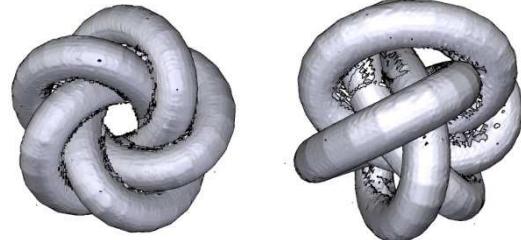


Рис. 4. Реконструкция сцены 1 по шести видам



Рис. 5. Исходные виды сцены 2 (2 из 11).

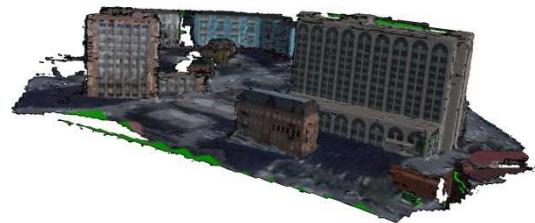


Рис. 6. Реконструкция сцены 2 по одиннадцати видам.



Рис. 7. Исходные виды сцены 3 «Temple».



Рис. 8. Реконструкция сцены 3 по трем (слева) и пяти (справа) видам.

	Размер- тьвокс. простран- ства	Количество видов		
		2	6	60
Модельная сцена 1 (8т.треуг. на вид)	128 <sup>3</sup>	66; 4.3; 0.9	145; 9.3; 1.5	641; 42; 3.4
	256 <sup>3</sup>	413; 21.2; 3.4	1034; 49; 6.2	4082; 186; 15.4
Модельная сцена 2 (22т.треуг. на вид)	128 <sup>3</sup>	132; 8.5; 1.1	194; 12.7; 2.4	894; 59; 4.2
	256 <sup>3</sup>	619; 31.2; 3.9	1438; 61.5; 6.7	5865; 225; 18.7
Реальная сцена 3 «Temple» (16т.треуг. на вид)	128 <sup>3</sup>	114; 7.5; 0.8	161; 10.7; 1.4	764; 52; 3.3
	256 <sup>3</sup>	589; 28.4; 3.4	1308; 55.5; 5.9	5143; 205; 13.8

Таблица 1. Время построения трехмерной модели без текстурирования (сек.). Первое значение – время работы аналога алгоритма [1], второе значение – время работы предложенного алгоритма с индексацией треугольников на CPU, третье значение – время работы алгоритма с использованием CUDA.

## 7. ЗАКЛЮЧЕНИЕ

В данной работе представлен алгоритм построения единой триагуляционной трехмерной текстурированной модели сцены по множественному набору пространственных данных, основанный на воксельном подходе. Предложенная структура данных с использованием индексации треугольников позволила существенно повысить производительность алгоритма в сравнении с аналогами. Дополнительное ускорение получено за счет организации параллельных вычислений на графических процессорах средствами языка CUDA.

## 8. СПИСОК ЛИТЕРАТУРЫ

- [1] Brian Curless and Marc Levoy. A Volumetric Method for Building Complex Models from Range Images // Computer Graphics (SIGGRAPH '96 Proceedings). 1996. – P. 303-312.
- [2] M. Goesele, B. Curless, S. M. Seitz. Multi-View Stereo Revisited // Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, Vol. 2. 2006. - P. 2402-2409.
- [3] W.E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm //

- [4] In Computer Graphics (SIGGRAPH '87 Proceedings), volume 21. 1987. - P. 163–169.  
P. Mordohai, J.-M. Frahm, A. Akbarzadeh, B. Clipp, C. Engels, D. Gallup, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewenius, H. Towles, G. Welch, R. Yang, M. Pollefeys, and D. Nister. Real-time video-based reconstruction of urban environments, - ISPRS Working Group V4 Workshop 3D-ARCH 2007: 3D Virtual Reconstruction and Visualization of Complex Architectures, (ETH Zurich, Switzerland), 2007. – P. 121-131.

- [5] Steve Seitz, Brian Curless, James Diebel, Daniel Scharstein, Richard Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms /Vision and Pattern Recognition. // IEEE Computer Society Conference on, vol. 1. 2006. – P. 519-526.

## Об авторах

Бобков Валерий Александрович - д.т.н, зав. лабораторией машинной графики Института автоматики и процессов управления ДВО РАН.

Адрес: г. Владивосток, 690090, ул. Радио, 5, ИАПУ.  
Телефон: (4232)313776  
E-mail: [bobkov@iacp.dvo.ru](mailto:bobkov@iacp.dvo.ru)

Кудряшов Алексей Павлович – к.т.н, м.н.с. лаборатории машинной графики Института автоматики и процессов управления ДВО РАН.

Адрес: Владивосток, 690090, ул. Радио, 5, ИАПУ  
E-mail: [kudryashova@iacp.dvo.ru](mailto:kudryashova@iacp.dvo.ru)

## Меры искажения на основе энтропийных характеристик для анализа цифровых изображений

В.Н. Горбачев, И. К. Метелев, Е.М. Кайнарова, Е.С. Яковлева

Северо-Западный институт печати

Санкт-Петербургского государственного университета технологии и дизайна, Санкт-Петербург, Россия

{Елена Кайнарова}helenkainarova@gmail.com

### Аннотация

Для характеристики различия двух цифровых изображений известен большой набор мер. Однако из-за отсутствия универсального критерия выбор подходящей меры, описывающей искажения в каждом конкретном случае, превращается в самостоятельную задачу. Для процесса бинаризации, известного в полиграфии как растиривание, и процесса встраивания цифровых данных, это область современной стеганографии, оказываются эффективными меры на основе относительной энтропии.

В стеганографии с помощью относительной энтропии сформулирован критерий  $\epsilon$ -секретности. Секретность является важным свойством, которое отличает любую профессиональную стеганографическую систему. На примере задачи встраивания бинарного изображения в полутоновое рассмотрена секретность двух алгоритмов, для которых вычислено значение относительной энтропии, усредненное по 2000 изображений. Аналогично критерию секретности с помощью относительной энтропии может быть введен критерий устойчивости к шумам для алгоритмов бинаризации, которые необходимы для печати полиграфической продукции. На практике уровень ошибок при печати определяется шумами. Поэтому использование более устойчивых к шумам алгоритмов бинаризации позволяет снизить число ошибок в тираже. Целесообразность подхода на основе относительной энтропии обусловлена ее свойствами, благодаря которым, введенный критерий устойчивости оказывается согласованным с оценкой эксперта, определяющего качество изображений. Для примера проведено сравнение ряда алгоритмов бинаризации с целью выявить наиболее устойчивый к шумам.

### Keywords:

меры искажения, относительная энтропия, стеганография, алгоритмы бинаризации .

### 1. ВВЕДЕНИЕ

На практике задача сравнить два изображения встречается часто и играет важную роль. Однако изображение является сложным объектом, предназначенным, как правило, для восприятия человеком, поэтому универсальных решений нет, что является источником постоянных поисков. Чтобы сравнивать цифровые изображения, разработан большой набор объективных мер искажений, которые применяются для конкретных задач [1].

Цель работы рассмотреть возможность практического применения относительной энтропии в качестве меры искажения при обработке цифровых изображений. Для этого были выбраны две задачи: встраивание цифровых данных, этим занимается современная стеганография, и процесс бинаризация (halftoning), который используется при печати.

Относительная энтропия характеризует различие двух

распределений случайной величины, она хорошо известна в теории информации, статистике и стеганографии, где на ее основе сформулирован критерий  $\epsilon$ -секретности [2].

Для задачи встраивания цифровых данных критерий  $\epsilon$ -секретности позволяет сравнивать различные стеганографические системы. На практике процесс сравнения оказывается не столь простым. Нерегулярное поведение характеристик цифрового изображения не позволяет построить удовлетворительные статистические модели изображений, поэтому для тестирования требуется большое число цифровых изображений. Обычно используют базы содержащие до 2000 изображений [3].

Аналогично критерию  $\epsilon$ -секретности с помощью относительной энтропии может быть сформулирован критерий устойчивости к шумам алгоритмов бинаризации в задаче растиривания. Это позволяет сравнивать различные алгоритмы, чтобы выбрать те, которые будут наиболее устойчивыми к шумам процесса печати. Шумы процесса печати являются источником ошибок в тираже, они имеют физическую природу, устраниТЬ их нельзя, однако путем выбора подходящего алгоритма бинаризации число ошибок можно уменьшить. Универсальных алгоритмов бинаризации, которые одинаково хорошо обрабатывали бы все изображения, нет. Поэтому для сравнения заданных алгоритмов также необходимо тестировать большой набор изображений. В качестве заданных были взяты алгоритмы бинаризации, представленные в [4] и [5], из них был найден наиболее устойчивый к шумам. Критерий устойчивости к шумам относится к объективным мерам искажения. Но благодаря свойствам относительной энтропии он оказывается согласованным с оценкой эксперта, который сравнивает качество изображений, полученных различными алгоритмами бинаризации.

### 2. ОТНОСИТЕЛЬНАЯ ЭНТРОПИЯ

Описывает различие распределений и ошибки при детектировании случайных сигналов.

Чтобы сравнить два распределения случайной величины  $p(x)$  и  $q(x)$ , где  $x$  дискретная переменная, используют относительную энтропию

$$Q(p||q) = \sum_x p(x)(\log p(x) - \log q(x)).$$

Можно показать, что  $Q(p||q)$  неотрицательна и равна нулю для одинаковых распределений  $p(x) = q(x)$ . Относительная энтропия не обладает симметрией по аргументам  $p$  и  $q$  и не удовлетворяет неравенству треугольника, поэтому не является метрикой в математическом смысле. Важным свойством является монотонность, которое означает, что при детерминированном преобразовании  $x \rightarrow y = f(x)$ ,  $p(x) \rightarrow p' = p(f(x))$ ,  $q \rightarrow q' = q(f(x))$

$$Q(p||q) \geq Q(p'||q'). \quad (1)$$

Это неравенство означает, что при преобразовании информации относительная энтропия не растет.

Относительная энтропия связана с проверкой гипотез. Пусть проверка заключается в том, чтобы различить два распределения  $p(x)$  и  $q(x)$ , когда принимается одно из двух решений - данное распределение есть либо  $p(x)$ , либо  $q(x)$ . В такой схеме возникают ошибки первого и второго рода, вероятность которых обозначим  $\alpha$  и  $\beta$ . Так, в задаче обнаружения, где цель описывается распределением  $p(x)$ , ошибка первого рода, или ложное срабатывание возникает, когда  $q(x)$  принято за  $p(x)$ . Ошибка второго рода, пропуск цели, возникает, когда  $p(x)$  принято за  $q(x)$ . Поскольку процесс принятия решения является детерминированным преобразованием, то вероятности ошибок и относительная энтропия удовлетворяют неравенству (1)

$$Q(\alpha||\beta) \leq Q(p||q). \quad (2)$$

Отсюда следует, что в силу свойства монотонности относительная энтропия может быть связана с ошибками при различении двух распределений.

Из этого неравенства можно получить следующую оценку. Если относительная энтропия ограничена сверху

$$Q(p||q) \leq \epsilon, \quad (3)$$

и  $\alpha = 0$ , то

$$\beta \geq 2^{-\epsilon}.$$

Это условие имеет простой смысл, чем ближе между собой распределения  $p(x)$  и  $q(x)$ , тем меньше их относительная энтропия  $Q(p||q)$ , тем больше будет вероятность ошибки второго рода  $\beta$ , когда  $p(x)$  принято за  $q(x)$ .

Приведенные свойства относительной энтропии позволяют использовать ее в качестве меры искажения.

### 3. ОТНОСИТЕЛЬНАЯ ЭНТРОПИЯ ДЛЯ СРАВНЕНИЯ СЕКРЕТНОСТИ АЛГОРИТМОВ ВСТРАИВАНИЯ

Одним из факторов, определяющих секретность алгоритмов встраивания цифровых данных в изображения, является мера неразличимости изображений. Это может быть описано с помощью относительной энтропии.

В стеганографической системе сообщение  $M$  встраивает в контейнер  $C$  в общем случае с помощью секретного ключа  $K$ . В результате получается заполненный контейнер  $S$ , его называют стегоконтейнером, из которого потом извлекают сообщение. Если  $E$  и  $D$  алгоритмы встраивания и детектирования, то в общем случае стеганографическая система описывается следующими соотношениями [3, 6]

$$\begin{aligned} E : C \otimes M \otimes K &\rightarrow S, \\ D : C \otimes S \otimes K &\rightarrow M. \end{aligned}$$

Два основных свойства стеганографической системы это неразличимость пустого и заполненного контейнера, когда  $C \approx S$ , и секретность.

Критерий секретности формулируется на основе относительной энтропии  $Q(C||S)$ , которая описывает различие между пустым и заполненным контейнером. Стеганографическая система называется  $\epsilon$ -секретной, если [2]

$$Q(C||S) \leq \epsilon. \quad (4)$$

Если  $\epsilon = 0$ , то система является совершенно секретной. В силу свойства относительной энтропии последнее означает, что пустой и заполненный контейнер оказываются неразличимы.

Далее рассматривается задача встраивания бинарного  $M$  изображения с секретным ключом в контейнер  $C$ , представленный полутоновым изображением. Использовалось два алгоритма. В первом случае бинарное изображение встраивалось в одну из битовых плоскостей путем побитового сложения

$$E_1 : C \rightarrow S = C - (B_V - B_V \oplus M \oplus K)2^{V-1},$$

где  $B_V$  одна из битовых плоскостей  $V = 1, 2, \dots$  полутонового изображения,  $K$  ключ, представленный случайной бинарной матрицей.

Во втором случае использовался алгоритм аддитивного встраивания, когда ключ прибавляется или вычитается в зависимости от значения бита  $m = 0, 1$ ,  $m \in M$

$$E_2 : C \rightarrow S = C + b(1 - 2M)K,$$

где  $b$  параметр, а ключ может быть представлен не обязательно бинарной матрицей.

Условие  $\epsilon$ -секретности в данном случае означает, что относительная энтропия

$$Q(p_C||p_S) = \sum_i p_C(i)(\log p_C(i) - p_S(i)) \leq \epsilon,$$

где  $p_C(i)$  и  $p_S(i)$  гистограммы яркости  $i = 0, 1, \dots, 255$  пустого и заполненного контейнера. Из этого условия следует, что чем меньше  $\epsilon$ , тем меньше различаются яркости полутонового изображения  $C$  и стегоизображения  $S$ . Из двух алгоритмов большую секретность имеет тот, у которого меньше  $\epsilon$ .

Результат расчета относительной энтропии для двух алгоритмов встраивания  $E_1$  и  $E_2$  представлен на рис.1. Первый алгоритм побитового сложения, рис. 1 а, для сокрытия бинарного изображения использует битовые плоскости  $V = 1, 2, \dots, 8$ . Здесь  $V = 1$  младшая битовая плоскость, ее вес 1,  $V = 8$  старшая битовая плоскость, ее вес  $2^7$ . Второй алгоритм аддитивного сложения, рис. 1 б, где параметр  $b \in [0, 1]$ . Оба графика получены усреднением по 2000 изображений. В рабочей области параметров, где  $V \in 1, 2, 3, b \in [0, 0.5]$ , относительная энтропия алгоритма побитового сложения, она лежит в интервале  $[0.0148, 0.02]$ , на два порядка больше. Это означает, что из этих двух алгоритмов более высокий уровень секретности имеет алгоритм аддитивного встраивания.

### 4. КРИТЕРИЙ УСТОЙЧИВОСТИ К ШУМАМ АЛГОРИТМОВ БИНАРИЗАЦИИ НА ОСНОВЕ ОТНОСИТЕЛЬНОЙ ЭНТРОПИИ

Процесс бинаризации необходим для печати цифрового изображения. Устойчивость алгоритмов бинаризации к шумам процесса печати может быть описана с помощью относительной энтропии.

Бинаризация заключается в преобразовании цифрового полутонового изображения, пиксель которого имеет яркость  $i = 0, 1, \dots, 255$ , в бинарное  $G$ , пиксель которого принимает только два значения черный и белый. Для моделирования процесса печати можно использовать бинарный канал с шумом, который будет описывать разнообразные шумы, возникающие при печати [7]. Если  $G'$  бинарное изображение на выходе канала с шумом, то относительная энтропия

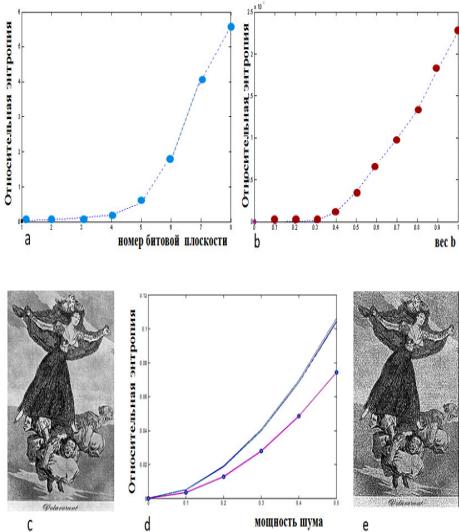


Рисунок 1: Относительная энтропия для задачи встраивания цифровых данных и процесса бинаризации полутонового изображения. а) алгоритм побитового сложения; б) алгоритм аддитивного встраивания. Каждая точка получена усреднением по 2000 картинок. с) исходное полутоновое изображение; е) бинарное изображение, полученное D-алгоритмом; д) сравнение устойчивости к шумам алгоритмов бинаризации. График получен путем усреднения по 80 изображениям из коллекции рисунков Caprichos.

$Q(G||G')$  характеризует соответствие между исходным и зашумленным изображением.

Критерий устойчивости алгоритма бинаризации к шумам можно сформулировать аналогично (4)

$$Q(G||G') \leq \epsilon.$$

Смысл этого условия простой. При заданном уровне шума, чем меньше  $\epsilon$ , тем устойчивость выше. Если  $\epsilon \approx 0$ , то алгоритм бинаризации, с помощью которого получено бинарное изображение  $G$ , нечувствителен к действию шума.

Разумеется критерий можно сформулировать на основе других мер искажения, взяв например, евклидово расстояние между  $G$  и  $G'$ . Использование однако относительной энтропии обусловлено следующим наблюдением. Относительная энтропия может описывать эксперту оценку качества изображений.

Справедливость этого наблюдения следует из свойства монотонности (2), если считать, что процесс принятия решения экспертом является детерминированным. Пусть эксперт, который оценивает качество, должен ответить на вопрос является ли представленное ему изображение исходным изображением  $G$  или зашумленным  $G'$ . В данном примере мы используем простейшую оценку. Если  $\epsilon$  мало, эксперт может совершить ошибку первого или второго рода. Так, он может решить, что ему предъявлено изображение  $G$  в то время как оно было  $G'$  и наоборот. Пусть вероятность ошибок  $\alpha$  и  $\beta$ . Если эксперту предъявляется только  $G$  или только  $G'$ , относительная энтропия между решениями будет  $Q(\alpha||\beta)$ . Тогда в силу (2)

$$Q(\alpha||\beta) \leq Q(G||G') \leq \epsilon.$$

Отсюда следует, что  $\epsilon$  – устойчивость алгоритма к шумам согласуется с оценкой эксперта.

Приведенный критерий может служить для сравнения работы разных алгоритмов бинаризации  $A_k$ ,  $k = 1, 2, \dots$ . Если  $G'_k$  зашумленное изображение, то можно вычислить значение относительной энтропии  $q_k = Q(G_k||G'_k)$ . Тогда алгоритм  $A_k$  будет более устойчивым, чем  $A_m$ , если  $q_k < q_m$ . Однако на практике такое неравенство может выполняться лишь в некоторой области параметров и для некоторого набора изображений, поскольку не существует идеального алгоритма бинаризации. В результате для оценки устойчивости возникает сложная задача с большим числом параметров.

Пример решения такой задачи показан на рис. 1. Здесь, на рис. 1 д, представлена относительная энтропия для семи алгоритмов бинаризации DOTDIF, ERRDIF, LAU, ULICHNEY, CDOD1, CDOD2, BAYER из [4] и D-алгоритма, введенного в [5]. Для зашумления бинарных изображений использовался бинарный шум с мощностью  $T \in [0, 0.5]$ , которая равна вероятности ошибки. Каждая точка была усреднена по 80 изображениям, выбранным из серии *Caprichos* Франциско Де-Гойя. Семь алгоритмов из [4] демонстрируют очень близкий результат, они показаны синей линией. Относительная энтропия для D-алгоритма, красная линия, отличается, она является наименьшей. Полученные монотонные зависимости означают, что устойчивость к шумам у D-алгоритма выше. Вместе с тем бинарные изображения D-алгоритма, одно из них показано на рис. 1 е, имеет высокое визуальное качество. Из найденного результата, например, следует, что для печати рисунков великого испанца D-алгоритм бинаризации является более предпочтительным.

## 5. ВЫВОДЫ

Используемые для сравнения изображений меры искажения обычно разделяются на объективные и субъективные. Меры искажения на основе относительной энтропии являются объективными. Однако монотонность относительной энтропии позволяет связать ее с ошибками в схемах детектирования весьма общего вида. Благодаря такой общности, эти схемы допускают описание ошибок, которые могут возникать, когда качество изображения оценивает эксперт.

Применение общей концепции относительной энтропии рассмотрено для двух задач. Первая из них – встраивание бинарного изображения в полутоновое, вторая – процесс бинаризации.

В задаче встраивания исследована секретность двух алгоритмов на основе известного критерия, сформулированного с помощью относительной энтропии. Чтобы получить надежный результат, было протестировано 2000 изображений. Мы получили монотонные зависимости от рабочих параметров, которые свидетельствуют в пользу одного из рассматриваемых алгоритмов. Отсюда следует, что критерий может быть использован на практике для сравнения секретности различных методов.

Для процесса бинаризации, которая используется для печати изображений, мы ввели критерий устойчивости к шумам алгоритмов бинаризации. Этот критерий, сформулированный на основе относительной энтропии, позволяет выбрать такие алгоритмы, которые при печати дают меньше ошибок в тираже. Мы сравнили набор базовых алгоритмов, для чего было вычислена относительная энтропия, усредненная по базе из 80 изображений. Поскольку универсального алгоритма бинаризации, который одинаково хорошо обрабатывает любые изображения нет, выбор базы оказался критичным. Полученные монотонные зависимости свидетельствуют в пользу выбранного критерия и позволяют выбрать один алгоритм. С практической точки зрения найден-

ные результаты означают следующее. Путем вычисления относительной энтропии для заданного набора изображений, можно подобрать алгоритм бинаризации, который позволит снизить уровень ошибок при печати.

## 6. СПИСОК ЛИТЕРАТУРЫ

- [1] B.W. Keelan, "Handbook of Image Quality", 2002.  
S. Wincler, "Digital Video Quality. Vision Models and Metrics", John Willey & Sons, Ltd, 2005.  
Д. Ватолин, А. Паршин, "Сравнение кодеков стандарта MPEG-4 AVC/H.264 с использованием объективных метрик", Conference Paper, GraphiCon, с. 447-454, 2006.
- [2] C. Cachin. "An information-theoretic model for steganography". In D. Aucsmith, editor, Information Hiding, 2nd International Workshop, volume 1525 of LNCS, pp 306 - 318. Springer-Verlag, New York, 1998.
- [3] I.J. Cox, M.L. Miller, J.F. Bloom, J. Fridrich, T. Kalter, "Digital Watermarking and Steganography", ELSEVIER, Morgan Kaufmann Publishers, 2008.
- [4] D.L. Lau, G.R. Arce, "Modern Digital Halftoning", Second Edition, CRC Press, 2008.
- [5] В.Н. Горбачев, В.Н. Дроздов, Е.С. Яковлева, "Об одном алгоритме бинаризации полутоновых изображений с выделением слабоконтрастных структур", Вестник МГУП, 3, с. 65, 2009.
- [6] J. Fridrich, "Steganography in Digital Media. Principles, Algorithms, and Application", Cambridge University Press, New York, 2010.
- [7] V.N. Gorbachev, E.S. Yakovleva, "A binary noisy channel to model errors in printing process", E-print, arXiv: submition/0329792 [cs.OH] 2 Okt 2011.

## АВТОРЫ

Валерий Николаевич Горбачев, преподаватель Северо-Западного института печати Санкт-Петербургского государственного университета технологии и дизайна (СЗИП СПб ГУТД).

Иван Кириллович Метелев, аспирант СЗИП СПб ГУТД, ivan.metelev@yandex.ru.

Елена Михайловна Кайнарова, преподаватель СЗИП СПб ГУТД, helenkainarova@gmail.com.

Елена Сергеевна Яковлева, преподаватель СЗИП СПб ГУТД, 2305lena@mail.ru.

# Устойчивая сегментация цифрового изображения

М.В. Харинов

Федеральное государственное бюджетное учреждение науки Санкт-Петербургский институт информатики и автоматизации Российской академии наук (СПИИРАН)  
khar@jias.spb.su

## Аннотация

В докладе рассматривается сегментация изображения посредством оптимальных приближений ступенчатой (кусочно-постоянной) функцией. Оптимальность устанавливается по среднеквадратичному отклонению вне зависимости от способа получения приближений и достигается по различным алгоритмам. Исследуется вычисление оптимальных приближений по признакам устойчивости относительно заданного множества модификаций. Для эффективного получения оптимальных приближений обобщается и развивается модель Мамфорда-Шаха (1985), которая в вычислительной части объединяется с мультипороговой версией гистограммного метода Оцу (1979). Предлагаемое решение обосновывается аналитически и экспериментально на примере стандартного изображения.

**Ключевые слова:** модель Мамфорда-Шаха, метод Оцу, метод  $k$ -средних, перекрывающиеся разбиения, среднеквадратичное отклонение, условие устойчивости.

## 1. ВВЕДЕНИЕ

В литературе проблему сегментации часто относят к слабо формализованным, что справедливо, если сегментация определяется не как результат, а как процесс разбиения пикселей изображения на множества по некоторому алгоритму. В этом случае, результирующее разбиение зависит от особенностей входных данных, что приводит к разделению изображений на ряд предметных областей (телевизионных, аэрокосмических, медицинских и пр.), для которых разрабатываются конкретные алгоритмы сегментации, а достоверность сегментации оценивается эмпирически по адекватности зрителю восприятию или результатам автоматического обнаружения заранее заданных объектов.

В более точной постановке проблема сегментации состоит в определении сегментированного изображения независимо от конкретного алгоритма или технологии его получения. В этом случае, вычисляемые сегменты отличаются от произвольных сегментов тем, что составляют оптимальные приближения изображения, минимально отличающиеся от изображения по значению некоторого функционала. При этом достоверность сегментации можно проверить формально без использования понятия зрительно воспринимаемых объектов, а выделение объектов трактовать как неформальную интерпретацию сегментации. В терминологии кластерного анализа [15], речь идет математической постановке задачи автоматической классификации, при которой искомые разбиения множеств на классы характеризуются оптимальным качеством. Одной из классических оценок качества разбиения является суммарная квадратичная ошибка или взвешенная сумма внутриклассовых дисперсий, которая является основой метода  $k$ -средних [6,7,15] и ряда других методов в абстрактном кластерном анализе и наглядной автоматической сегментации цифровых изображений.

## 2. ПОСТАНОВКА ЗАДАЧИ

Классическая оценка качества разбиения интенсивно применяется в методах оптимальной сегментации, однако в литературе приводятся недостаточно полные данные по оптимальным приближениям даже для стандартных изображений, в частности, для изображения «Лена». В гистограммном методе Оцу [11] строится оптимальное разбиение пикселей изображения на два множества (класса), отвечающее минимальному значению обсуждаемого функционала в одномерном варианте. В известной работе [5], для задач выделения на изображении обычных и текстурных объектов, реализуется мультипороговый метод Оцу и предпринимается попытка вычисления последовательности оптимальных приближений изображения. Результаты, однако, ограничиваются начальными пятью приближениями изображения из-за экспоненциального роста продолжительности вычислений с увеличением числа классов (градаций яркости), на которые разбивается множество пикселей изображения.

В случае последовательного разбиения пикселей по яркости на  $m = 1, 2, \dots$  градаций, минимизируемый функционал  $\sigma^2$  записывается как взвешенная сумма дисперсий  $\sigma_i^2$  по множествам  $i$ :

$$\sigma^2 = \sum_{i=1}^m \omega_i \sigma_i^2, \quad (1)$$

где  $\omega_i$  — вероятность, сопоставляемая множеству  $i$  (в [5] рассматривается эквивалентный функционал).

Для изображения из  $N$  пикселей величина  $N\sigma^2$  есть сумма квадратов отклонений яркостей пикселей от средних значений, вычисленных по каждому множеству пикселей. В модели Мамфорда-Шаха [1,4,8,9,12], в которой рассматриваются множества пикселей связных сегментов изображения, эта величина является основной компонентой «энергетического» функционала [4]:

$$N\sigma^2 + \lambda L, \quad (2)$$

где  $L$  — суммарная длина границ между сегментами, а  $\lambda$  — так называемый, «регуляризационный» параметр.

Величина  $\sigma$  в (2) описывает среднеквадратичное отклонение изображения от своего кусочно-постоянного приближения с усредненными по сегментам значениями пикселей.

В популярной работе [4] ряд последовательных значений параметра  $\lambda$  устанавливается эвристически в процессе обработки изображения. В современной практической версии модели FLSA (Full  $\lambda$ -Scheduled Algorithm, [1,12]), применяемой в программном комплексе "ENVI" (Environment for Visualizing Images, [http://www.itvis.com/portals/0/pdfs/envi/Feature\\_Extraction\\_Module.pdf](http://www.itvis.com/portals/0/pdfs/envi/Feature_Extraction_Module.pdf)), параметр  $\lambda$  устанавливается автоматически из условия неизменности функционала (2) при слиянии сегментов, что нарушает идею минимизации функционала, как справедливо отмечается в [1]. В [16] указанный параметр полагается равным 0,

что не противоречит (2), и сохраняет нетривиальный смысл минимизации  $\sigma$ , если она выполняется для каждого числа сегментов. В этом случае, оптимальность приближений определяется в классическом смысле — минимальным отличием приближения от изображения по среднеквадратичному отклонению  $\sigma$  или суммарной квадратичной ошибке  $E \equiv N\sigma^2$ :

$$\sigma = \min \Leftrightarrow E = \min, \quad (3)$$

что, в принципе, не исключает возможности учета границ между сегментами посредством аддитивной или мультиплексивной добавок в критерии слияния, как в версии [4] или версии FLSA [1,12] модели Мамфорда–Шаха.

Как мы установили в предыдущей работе [2], принципиальное ограничение вычислительной схемы в модели Мамфорда–Шаха заключается в недостаточности операции слияния сегментов для получения, вообще говоря, неиерархической последовательности оптимальных приближений. Для преодоления указанного недостатка мы обобщаем операцию слияния и вводим дополнительную операцию коррекции множеств пикселей изображения.

### 3. АНАЛИТИЧЕСКОЕ ОБОБЩЕНИЕ ОПЕРАЦИИ СЛИЯНИЯ

Рассмотрим реклассификацию  $k$  пикселей средней яркости  $I$ , исключаемых из множества 1 пикселей со средней яркостью  $I_1$  и включаемых в число пикселей множества 2 со средней яркостью  $I_2$ . Сопутствующее приращение  $\Delta E$  квадратичной ошибки  $E$  выражается в виде:

$$\Delta E = \begin{cases} \Delta E_{\text{merge}}, & k = n_1, \\ \Delta E_{\text{correct}}, & k < n_1, \end{cases} \quad (4)$$

где  $n_1, n_2$  — число пикселей в множествах 1 и 2, соответственно, а  $\Delta E_{\text{merge}}$  и  $\Delta E_{\text{correct}}$  имеют вид:

$$\Delta E_{\text{merge}} = \frac{(I_1 - I_2)^2}{\frac{1}{n_1} + \frac{1}{n_2}}, \quad \Delta E_{\text{correct}} = \frac{(I - I_2)^2}{\frac{1}{k} + \frac{1}{n_2}} - \frac{(I - I_1)^2}{\frac{1}{k} + \frac{1}{n_1}}. \quad (5)$$

$\Delta E_{\text{merge}}$  описывает приращение квадратичной ошибки при слиянии множеств пикселей посредством реклассификации всех пикселей из множества 1 в множество 2 и уменьшении числа множеств на 1.  $\Delta E_{\text{correct}}$  описывает приращение квадратичной ошибки при реклассификации части пикселей из множества 1 в множество 2 и сохранении числа множеств. Далее условимся разделять термины «слияние» и «реклассификация», употребляя последний только для обозначения модификации классификации части пикселей.

В вычислениях по Мамфорду–Шаху и [16] сегментация выполняется посредством итеративного слияния рассматриваемых множеств пикселей. Следуя [16], слияние множеств 1 и 2 выполняется при минимальной величине приращения квадратичной ошибки  $\Delta E_{\text{merge}}$ :

$$\Delta E_{\text{merge}} = \min. \quad (6)$$

Коррекция множеств выполняется посредством реклассификации пикселей из условия уменьшения квадратичной ошибки. Критерием коррекции является отрицательная величина приращения квадратичной ошибки  $\Delta E_{\text{correct}}$ :

$$\Delta E_{\text{correct}} < 0. \quad (7)$$

Элементарным преобразованием из (5) и (7) получаем условие коррекции, учитывающее только знак  $\Delta E_{\text{correct}}$ :

$$|I - I_1| > \alpha \cdot |I - I_2|, \quad (8)$$

где коэффициент  $\alpha < 1$  описывает соотношение количеств пикселей в множествах и определяется в виде:

$$\alpha \equiv \sqrt{\frac{n_2(n_1 - k)}{n_1(n_2 + k)}}. \quad (9)$$

Из (8), (9) для реклассификации  $k$  одинаковых пикселей следует, что если среднеквадратичное отклонение снижается при реклассификации одного пикселя, то оно снижается и при реклассификации остальных пикселей той же яркости, поскольку с ростом  $k$  коэффициент  $\alpha$  уменьшается.

Применяя логическое отрицание к выражениям (7), (8), получаем условие устойчивости:

$$\Delta E_{\text{correct}} \geq 0 \Leftrightarrow |I - I_1| \leq \alpha \cdot |I - I_2|. \quad (10)$$

Разбиение пикселей изображения на множества считается устойчивым, если для всех рассматриваемых пар множеств реклассификация заданных подмножеств пикселей из одного множества в другое не приводит к уменьшению  $\sigma$ .

Очевидно, что оптимальное разбиение изображения является устойчивым, но устойчивое разбиение не обязательно оптимально. Ограничения на допустимые пары множеств 1 и 2, например, обменивающихся пикселями при сохранении связности, или рассмотрение одновременной реклассификации двух и более подмножеств пикселей приводят к различным вариантам программной реализации условия устойчивости (10). Здесь же следует заметить, что метод  $k$  — средних [6,7,15], вообще говоря, приводит к неустойчивому разбиению пикселей с завышенным значением  $\sigma$ , поскольку в этом методе не учитывается коэффициент  $\alpha$  (9).

### 4. МЕТОДЫ МИНИМИЗАЦИИ $\sigma$

Для вычисления последовательности оптимальных разбиений пикселей изображения на множества, обеспечивающих минимизацию среднеквадратичного отклонения приближений от изображения, мы воспроизвели вычисления в версии FLSA модели Мамфорда–Шаха и, для сравнения, в версии [16]. Затем, в рамках версии [16], реализовали вычисление неиерархических приближений с обычными связными сегментами и со связными множествами из несвязных пикселей, а также разработали новую версию мультипорогового неиерархического метода Оцу.

В мультипороговом методе Оцу искомые множества пикселей отвечают диапазонам яркости, на которые по гистограмме разбивается рабочий диапазон яркости. При этом учитывается свойство компактности оптимальных разбиений по яркостной оси, состоящее в том, что число классов, на которые разбивается множество пикселей, совпадает с числом диапазонов (градаций), на которые разделяется рабочий диапазон яркости. Требуется разбить рабочий диапазон яркости на последовательное число диапазонов так, чтобы при замещении значений пикселей усредненными по диапазонам значениями яркости, среднеквадратичное отклонение  $\sigma$  полученного приближения от изображения оказалось минимальным при каждом числе диапазонов от 1 до максимального значения  $M$ , равного числу встречающихся на изображении яркостей. В отличие от [5], в нашей реализации метода Оцу строится полная последовательность

тельность оптимальных разбиений.

Пара тривиальных разбиений на 1 и на  $M$  диапазонов задана. Остальные разбиения получаются и преобразуются слиянием или дроблением текущих диапазонов с изменением числа диапазонов и числа соответствующих градаций средней яркости на единицу, как в [6]. Для каждого разбиения итеративно осуществляется минимизация квадратичной ошибки  $E$  «по частям», которая выполняется для всех перекрывающихся участков гистограммы, состоящих не более чем из  $l$  последовательных диапазонов яркости ( $l = 2, 3, \dots$ ). При этом оптимальные пороговые значения яркости находятся перебором вариантов.

Эксперимент показал, что с ростом  $l$  возрастает продолжительность обработки, как в [5]. Но, с другой стороны, уже при начальных значениях  $l$  последовательность искомых приближений стабилизируется, и снижается максимальная величина модификации  $\sigma$ , нулевое или близкое к нулю значение которой может служить критерием окончания оптимизации. Расчет устойчивых приближений и их оптимизация по  $\sigma$  выполняется по формулам (4)-(10) без суммирования квадратов значений яркости.

В разработанном обобщенном варианте вычислений по модели Мамфорда–Шаха и [16] в качестве начального берется любое разбиение пикселей изображения на  $N$  множеств, в частности, на отдельные пиксели, которое затем модифицируется за счет операций слияния и коррекции. Строится последовательность разбиений от 1 до  $N$ . Для множеств пикселей задается отношение смежности по аналогии со связными сегментами — два множества считаются смежными, если на изображении встречаются смежные пиксели из этих множеств. Множества пикселей делятся на подмножества, образуемые пикселями одинаковой яркости. В качестве указанных подмножеств рассматривались либо отдельные пиксели, либо все пиксели каждой яркости из числа пикселей, отнесенных к данному множеству.

Минимизация среднеквадратичного отклонения  $\sigma$  приближений от изображения выполняется в алгоритмах поочередной коррекции разбиений пикселей изображения на множества и слияния смежных множеств пикселей.

Слияние смежных множеств пикселей выполняется из условия (6) и сопровождается уменьшением числа множеств пикселей, а также увеличением  $\sigma$ . В процессе коррекции смежные множества пикселей обмениваются подмножествами пикселей одинаковой яркости из условий (7), (8) уменьшения  $\sigma$ . Из нескольких возможностей реклассификации данного подмножества пикселей выбирается та, что обеспечивает максимальное падение  $\sigma$  согласно выражению  $\Delta E_{correct}$  в (5). Коррекция циклически повторяется до получения устойчивого разбиения изображения на множества пикселей, при котором реклассификация любого предусмотренного подмножества пикселей из данного множества в смежное увеличивает среднеквадратичное отклонение, или, по крайней мере, оставляет его неизменным. На каждом шаге коррекции среднеквадратичное отклонение падает, а число множеств пикселей не меняется.

Результат сегментации зависит от условий выполнения коррекции множеств пикселей. При блокировании коррекции, нарушающей связность пикселей, вычисляется последовательность разбиений изображения на связные сегменты. Если же при коррекции допускается нарушение связности сегмента–донора, то результатом сегментации служат множества пикселей в виде наборов несмежных сегментов с одинаковой средней яркостью.

## 5. ЭКСПЕРИМЕНТАЛЬНЫЕ РЕЗУЛЬТАТЫ

Рис. 1 иллюстрирует оптимальные приближения, вычисленные для стандартного изображения «Лена».

$$\sigma = 31.60341$$



Рисунок 1: Оптимальные приближения изображения.

Исходное изображение показано в левом верхнем углу. Рядом, в правом верхнем углу, показано «близкое к оптимальному» приближение исходного изображения двумя связными сегментами [2], — значение среднеквадратичного отклонения  $\sigma$  указано над приближением. Остальные 8 изображений иллюстрируют оптимальные приближения исходного изображения в последовательном числе градаций яркости и размещены в порядке возрастания числа градаций от 1 до 8 слева направо и сверху вниз.

Таблица.

Оценки  $\sigma$  для оптимальных приближений изображения «Лена» в  $m$  градациях яркости

$m$	$\sigma$	$m$	$\sigma$	$m$	$\sigma$
1	55.88322	17	3.70326	33	1.72854
2	30.64564	18	3.50441	34	1.68547
3	21.21739	19	3.32383	35	1.64729
4	14.96450	20	3.15658	36	1.61313
5	11.69762	21	3.01260	37	1.57894
6	10.03975	22	2.87305	38	1.54589
7	8.46072	23	2.73844	39	1.51218
8	7.51121	24	2.60572	40	1.47850
9	6.81359	25	2.49076	41	1.44586
10	6.14397	26	2.37391	42	1.41400
11	5.57864	27	2.25584	43	1.38327
12	5.11403	28	2.15389	44	1.35340
13	4.75689	29	2.05655	45	1.32287
14	4.42306	30	1.95565	46	1.29542
15	4.17825	31	1.87573	47	1.26794
16	3.92460	32	1.79485	48	1.23991

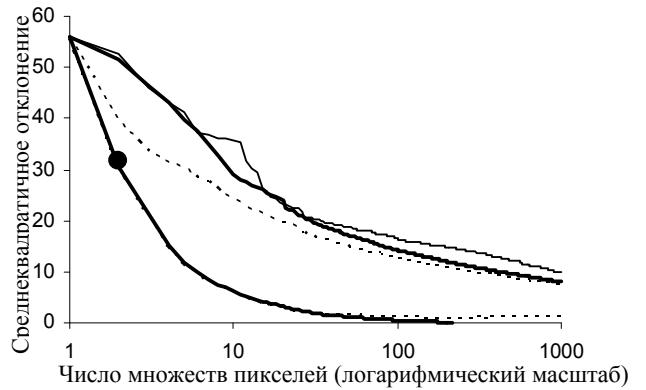


Рисунок 2: Графики зависимости оценок  $\sigma$  от числа множеств пикселей для приближений изображения «Лена».

Нетрудно заметить, что приближение изображения двумя связными сегментами имеет выраженную структуру и состоит из «площадок» и «соединительных» элементов шириной в один пиксель, слабо влияющих на среднеквадратичное отклонение. При этом оптимальное приближение в двух градациях можно преобразовать в приближение, близкое к оптимальному, если отдельные «площадки» на нем стереть, а остальные соединить между собой посредством «соединительных» элементов.

Всего оптимальных приближений изображения в различном числе градаций 216, так как в изображении «Лена» встречаются пиксели 216 градаций исходной яркости. Для приближений в  $m = 1, 2, \dots, 48$  усредненных яркостных градациях значения  $\sigma$  перечислены в четных столбцах таблицы — по 16 штук в каждом столбце.

Оптимальные приближения вычислены «по частям» в нашей реализации мультипорогового метода Оцу при  $l \leq 3$ . Проверено, что на полученные разбиения не влияет преобразование изображения в негатив, хотя процесс вычисления приближений при этом меняется. В ходе экспериментов ни одно перечисленных в таблице значений  $\sigma$  не удалось уменьшить за счет применения алгоритмов анализа связных множеств и др. Однако пока мы не исключаем, что при следующих значениях  $l$  некоторые значения  $\sigma$  окажется возможным уточнить<sup>1</sup>.

Характерно, что последовательность оптимальных приближений изображения не является иерархической. При этом яркостные диапазоны из различных разбиений гистограммы и сегменты из различных разбиений изображения перекрываются между собой (рис. 1).

Если в мультипороговом методе Оцу перебор пар множеств пикселей ограничивается благодаря условию компактности, то в вычислениях по модели Мамфорда–Шаха и [16] сравниваются только смежные множества пикселей. Интересно выяснить, как согласуются при этом результаты расчетов в той и другой моделях.

Для изучения эффекта учета пространственного распределения пикселей мы выполнили программную реализацию модели Мамфорда–Шаха, поддерживающую операции слияния и коррекции для любых множеств пикселей. При начальном разбиении изображения на связные сегменты и единственной операции слияния смежных сегментов выполняются вычисления со связными сегментами, как в классических вариантах модели Мамфорда–Шаха [1,4,8,9,12] и версии [16]. Операция коррекции, в общем случае, выводит из множества связных сегментов. Поэтому при выполнении коррекции связность пикселей, отнесенных к одному множеству, учитывается как самостоятельное условие [2], или игнорируется. В последнем случае, учет геометрических свойств множеств пикселей ограничивается отношением смежности между множествами.

Зависимость  $\sigma$  от числа множеств пикселей для результирующих разбиений иллюстрируется графиками на рис. 2.

На рис. 2 зависимость оценок  $\sigma$  от числа градаций яркости в диапазоне  $m$  от 1 до 216 показана сплошной нисходящей линией, которая описывает результаты вычислений в нашей версии мультипорогового метода Оцу. Сливающаяся с

<sup>1</sup> Примеры оптимальных, точнее говоря, предельно близких к оптимальным приближений реальных изображений будут выложены на сайте СПИИРАН по электронному адресу <http://www.spiiras.nw.ru> с целью применения в задачах тестирования и сравнения алгоритмов оптимизации по  $\sigma$ .

ванием оптимального приближения в двух градациях яркости с использованием интерактивного управления процессом установления «соединительных элементов».

Как видно на рис. 2, введение операции коррекции заметно улучшает приближения изображения по  $\sigma$ , выражющееся в том, что пунктирная кривая располагается ниже переплетающихся кривых. Тем не менее, поскольку жирная точка на рис. 2 оказывается почти в полтора раза ниже соответствующего значения  $\sigma$  на пунктирной кривой, можно заключить, что автоматическая аппроксимация изображения связанными сегментами [2] все же приводит к завышенным значениям  $\sigma$  из-за не решенной пока проблемы эффективного автоматического установления соединительных элементов между сегментами. Как показано в [2], из-за недостаточно точной аппроксимации по  $\sigma$ , в сегментированном изображении пропускаются те или иные визуально воспринимаемые объекты, причем результаты разбиения изображения на несколько сегментов зависят от начального разбиения изображения, особенностей вычислений и пр. Указанную трудность удается обойти за счет рассмотрения смежных множеств пикселей без учета связности пикселей из одного множества, т.е. за счет ослабления связности.

Эксперимент показывает что, если в качестве начального разбиения на множества берется разбиение пикселей изображения по градациям исходной яркости с учетом связности множеств, но без учета связности пикселей из одного множества, то получаемые разбиения воспроизводят результаты вычислений, полученные перебором пороговых значений по методу Оцу. График зависимости  $\sigma$  от числа множеств, вычисленных в алгоритме слияния/коррекции, сливается с графиком  $\sigma$  в зависимости от числа градаций.

Практически к тем же самим разбиениям на перекрывающиеся множества приводят вычисления при начальном разбиении изображения на отдельные пиксели и достаточной редукции числа множеств пикселей. В случае стандартного изображения (рис. 1), слияние смежных множеств пикселей по очереди с коррекцией получаемых разбиений приводит к тому, что при снижении числа формируемых множеств до 30 все пиксели каждой градации исходной яркости оказываются отнесенными к одному и тому же множеству, что обеспечивает воспроизведение оптимальных приближений в 1, 2, ..., 30 градациях яркости. По визуальному восприятию, сравниваемые приближения совпадают между собой, так как при более чем 30 множествах пикселей (градациях средней яркости) заметным образом не отличаются от исходного изображения.

## 6. ПРИМЕНЕНИЕ

Оптимальные приближения интересны не только тем, что их можно получать по различным алгоритмам. Оказывается, что они устойчивы относительно изменения масштаба изображения и получаются сходными для изображений в различной форме представления (рис. 3).

На рис.3 показан результат покомпонентной сегментации цветового изображения "Лена" из  $512 \times 512$  пикселей (слева) с последующим преобразованием в полутоновое представление (справа), посредством усреднения оптимальных приближений в трех градациях каждой из RGB-компонент.

Несмотря на то, что обсуждаемое изображение рис. 3 цветовое и по размерам в два раза превосходит изображение рис. 1, результат его сегментации (справа на рис. 3) оказывается близок к результату сегментации в четырех градациях изображения "Лена" из  $256 \times 256$  пикселей (в централь-



Рисунок 3: Сегментация цветового изображения.

ном ряду справа на рис. 1). Таким образом, формальное условие устойчивости (10) обеспечивает устойчивость сегментации изображения в прикладном смысле.

В контексте проблемы автоматического распознавания смысл получения последовательности оптимальных приближений рис. 1 заключается в вычислении для выделения объектов множества сегментов, составляющих ничтожную долю всевозможных сегментов изображения. В этом случае, в процессе фильтрации при детектировании объектов необходимо проанализировать признаки сравнительно небольшого числа сегментов.

Особенностью оптимальных приближений изображения является то, что они задаются последовательностью разбиений с перекрытиями сегментов. При этом иерархическая структура оказывается признаком, который может использоваться для детектирования объектов.

## 7. ОСОБЕННОСТИ ВЫЧИСЛЕНИЙ

Обращаясь к модели Мамфорда-Шаха [1,12] и [16], не следует недооценивать вычислительной проблемы, связанной с необходимостью скоростного выполнения порядка  $N - 1$  итераций, где  $N$  — число пикселей в изображении. Проблема усугубляется при встраивании в многочисленные итерации циклической же коррекции множеств пикселей, но решается без рутинной стандартной оптимизации программ за счет тысячекратного снижения числа итераций в быстром алгоритме одновременного парного слияния смежных множеств пикселей по всему полю изображения [2], который относится к алгоритмам поглощения [15].

С другой стороны, возникает проблема запоминания последовательности перекрывающихся разбиений на сегменты для оперативного преобразования и обработки в памяти компьютера. Этую проблему, вероятно, можно решить посредством аппроксимации последовательности перекрывающихся разбиений несколькими иерархическими последовательностями разбиений, для хранения, преобразования и фильтрации которых в фиксированном объеме памяти разработан аппарат дихотомической сегментации изображения в терминах структуры данных «динамических» деревьев Слейтора-Тарьяна [3,13,14,17]. Следует отметить, что упомянутый аппарат разрабатывается нами с начала 90-х годов прошлого века, тогда как в практику обработки изображений активно начинает внедряться только с начала текущего века [10], пока в менее освоенном варианте (<http://www.lix.polytechnique.fr/~nielsen/Srmjava.java>).

Во избежание зацикливания, при программировании коррекции разбиения изображения рекомендуется точно детектировать нулевое изменение  $\Delta E_{correct}$  и обновлять значения параметров множеств пикселей (число пикселей и среднюю яркость) при каждой реклассификации их подмножеств. Следует обратить внимание также на то, что в

режиме on-line модификации параметров множеств пикселей порядок анализа и преобразования множеств сказывается на процессе минимизации среднеквадратичного отклонения, но не влияет на результаты оптимизации, в случае достижения глобального минимума. При этом инвариантность результата сегментации может служить критерием окончания оптимизации, необходимого, например, при переборе пороговых значений яркости в мультипороговом методе Оцу. В противном случае, изменение порядка модификации множеств допустимо использовать как дополнительный прием для получения разбиений с минимальным среднеквадратичным отклонением  $\sigma$ .

Для исключения влияния изменения порядка слияния/коррекции множеств на процесс вычислений достаточно упорядочить пары множеств по величине изменения  $\sigma$  или другому параметру. Однако, судя по первым экспериментам с различными изображениями, упорядочение пар множеств, существенно не влияет на результаты оптимизации, замедляя при этом вычисления.

Решение вычислительных проблем систематизации, упорядочения и фильтрации результатов сегментации для удобства выделения конкретных объектов является интересным направлением дальнейших исследований.

## 8. ЗАКЛЮЧЕНИЕ

Таким образом, в докладе аналитически и экспериментально обоснован подход к сегментации цифровых изображений посредством оптимальных кусочно-постоянных приближений, характеризующихся минимальными значениями среднеквадратичного отклонения приближения от изображения. В качестве основы вычислений использованы метод Оцу и модель Мамфорда–Шаха, которые объединены в единой постановке задачи согласно версии [16].

В нашей реализации метода Оцу разработан гистограммный способ вычисления последовательных оптимальных приближений изображения посредством коррекции "по частям" и изменений числа градаций на единицу. Для оптимизации приближений по среднеквадратичному отклонению предложено дополнить слияние множеств пикселей в модели Мамфорда–Шаха и [16] операцией коррекции (7)–(9) и применять обобщение понятия связности множеств пикселей, к которым либо предъявляется, либо не предъявляется условие связности пикселей, относимых к одному множеству. В качестве формального признака оптимального разбиения установлено свойство (10) устойчивости сегментации относительно предусмотренных вариантов реклассификации пикселей.

Одним из перспективных направлений дальнейших исследований является обобщение формул (4)–(10) на многомерный случай в контексте разработки и реализации метода кластеризации, альтернативного методу  $k$ -средних, который, вероятно, будет полезен для улучшения сегментации цветовых изображений.

## 9. ССЫЛКИ

- [1] Crisp D.J., Tao T.C. Fast Region Merging Algorithms for Image Segmentation // *The 5th Asian Conf. on Computer Vision (ACCV2002)*, Melbourne, Australia, 23–25 January 2002. — pp. 1–6.
- [2] Kharinov M.V. Image segmentation by means of optimal approximations // Компьютерная графика и зрение (Графикон'2011) / Труды 21-й между. конф.. М: МАКС Пресс, 26–30 сентября 2011. — С. 145–148.
- [3] Kharinov M.V. Adaptive Dichotomous Image Segmentation Toolkit // *Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications*. Vol. 22, № 1, 2012. — pp. 228–235.
- [4] Koepfler G., Lopez C., Morel J. A Multiscale Algorithm for Image Segmentation by Variational Method // *SIAM Journal on Numerical Analysis*, Vol. 31, No 1, 1994. — pp. 282–299.
- [5] Ping-Sung Liao, Tse-Sheng Chen, Pau-Choo Chung. A Fast Algorithm for Multilevel Thresholding // *J. Inf. Sci. Eng.* Vol. 17 (5), 2001. — pp. 713–727.
- [6] Likas A, Vlassis N. and Verbeek J.J. The Global K-Means Clustering Algorithm // *Pattern Recognition*. Vol. 36, 2003. — pp. 451–461.
- [7] MacQueen J.B. Some Methods for classification and Analysis of Multivariate Observations // *Proc. Fifth Berkeley Symp. Math. Stat. and Probab.*, Berkeley: University of California Press, Vol. 1, 1967. — pp. 281–297
- [8] Mumford D., Shah J. Boundary detection by minimizing functionals, I // *Proc. IEEE Comput. Vision Patt. Recogn. Conf.*, San Francisco, 1985. — pp. 22–26.
- [9] Mumford D., Shah J. Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems // *Communications on Pure and Applied Mathematics*, Vol. XLII, No 4, 1989. — pp. 577–685.
- [10] Nock R., Nielsen F. Statistical Region Merging // *IEEE Trans. Pattern Anal. Mach. Intell.* Vol. 26(11), 2004. — pp. 1452–1458.
- [11] Otsu N. A Threshold Selection Method from Gray-Level Histograms // *IEEE Transactions on systems, MAN, and CYBERNETICS*, Vol. SMC-9, №. 1, 1979. — pp. 62–66.
- [12] Robinson B.J., Redding N.J., Crisp D.J. Implementation of a fast algorithm for segmenting SAR imagery // *Scientific and Technical Report*, Australia: Defense Science and Technology Organization, Australia, 01 January 2002. — 42 p.
- [13] Sleator D.D.; Tarjan R.E. Self-Adjusting Binary Search Trees // *Journal of the ACM* Vol. 32 (3), 1985. — pp. 652–686
- [14] Tarjan R.E. Efficiency of a Good But Not Linear Set Union Algorithm // *Journal of the ACM* Vol.22 (2), 1975. — pp. 215–225.
- [15] Айвазян С.А., Бухштабер В.М., Енуков И.С., Мешалкин Л.Д. Прикладная статистика: Классификация и снижение размерности. М.: Финансы и статистика, 1989. — 607 с.
- [16] Бугаев А.С., Хельвас А.В. Поисковые исследования и разработка методов и средств анализа и автоматического распознавания потоковой информации в глобальных информационных системах. Шифр «Лацкан» // *Отчет по НИР*. М.: МФТИ, 2001. Том 1, — 140 с.
- [17] Харинов М.В. Запоминание и адаптивная обработка информации цифровых изображений. СПб.: Изд-во С.Петерб. ун-та, 2006. — 138 с.

## Об авторе

М.В. Харинов — старший научный сотрудник СПИИРАН. Его адрес: [khar@iias.spb.su](mailto:khar@iias.spb.su).

# Интегральные Спиновые Изображения в Системах Поиска Трёхмерных Моделей

Александр Крыловецкий, Игорь Черников  
Факультет компьютерных наук

Воронежский государственный университет, Воронеж, Россия  
aakryl@sc.vsu.ru, ichernikov@yandex.ru

## Аннотация

Актуальность задачи поиска трёхмерных моделей обусловлена быстрым ростом популярности и доступности цифровых пространственных изображений в Internet, а также появлением специализированных баз данных. Данная задача является одной из самых сложных в области компьютерного зрения, поскольку для ее решения требуется эффективный и робастный алгоритм сравнения трёхмерных полигональных моделей произвольной формы. Один из подходов к этому основан на глобальных дескрипторах поверхности, содержащих информацию о форме всей поверхности модели в сжатом и ёмком виде. В данной статье мы представляем новые глобальные дескрипторы поверхности *интегральные спиновые изображения*, которые, по сути, являются расширением известных локальных дескрипторов спиновых изображений. Мы также предлагаем специальный метод нормализации трёхмерной модели для вычисления интегрального спинового изображения. В статье показано, что новый метод сравнения моделей с использованием интегральных спиновых изображений обеспечивает адекватную меру похожести формы моделей и может успешно применяться в системах поиска трёхмерных моделей.

**Ключевые слова:** *Интегральное спиновое изображение, Система поиска, Сравнение 3D моделей, Нормализация 3D моделей.*

## 1. ВВЕДЕНИЕ

Центральной проблемой, возникающей при поиске в трёхмерных базах данных, является разработка метода сравнения трёхмерных моделей. В общем случае модели имеют разный масштаб, по-разному ориентированы и локализованы в пространстве, что делает невозможным непосредственное сравнение 3D моделей. Поэтому изначально для каждой модели формируется интегральное представление её формы (глобальный дескриптор поверхности). Затем эти дескрипторы сравниваются друг с другом посредством одного из статистических методов (например, коэффициент корреляции). Таким образом, работа системы поиска трёхмерных моделей может быть разбита на два независимых этапа: онлайн и оффлайн. На этапе оффлайн для всех моделей из базы вычисляются глобальные дескрипторы поверхности. Вычисление дескриптора модели-запроса, сравнение запроса с моделями из базы и формирование ответа системы происходит на этапе онлайн. Схематическое описание системы поиска и пример её работы приведены на Рис. 1.

В настоящей работе мы предлагаем новый метод поиска 3D моделей с использованием интегральных спиновых изображений – глобальных дескрипторов поверхности, основанных на идеи известных локальных дескрипторов поверхности спиновых изображений [2, 3]. Для построения

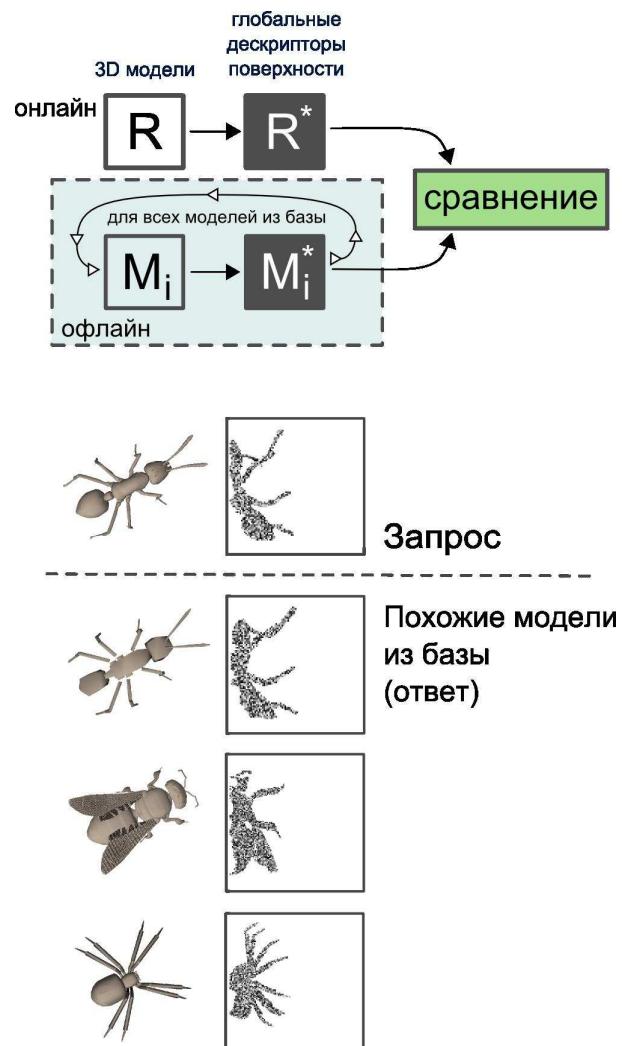


Рис 1: Схема и пример работы системы поиска трёхмерных моделей.

интегральных спиновых изображений необходимо вычислить опорную точку и вектор. Интегральные спиновые изображения также зависят от масштаба модели. Поэтому непосредственному вычислению этих дескрипторов должна предшествовать специальная процедура нормализации модели.

Статья организована следующим образом. В секции 2 мы напомним идею обычных спиновых изображений. Секция 3 содержит определение интегральных спиновых изображений и особенности их вычисления. В секции 4 изложена процедура нормализации 3D модели. Алгоритм сравнения интегральных спиновых изображений представлен в секции 5. Некоторые результаты предложенного метода также находятся в этой секции. Секция 6 заключительная.

## 2. ОБЫЧНОЕ СПИНОВОЕ ИЗОБРАЖЕНИЕ

Спиновое изображение – это локальный дескриптор поверхности, успешно используемый в таких задачах компьютерного зрения, как распознавание объектов и совмещение поверхностей [1, 2, 3]. Спиновое изображение строится относительно одной из точек модели (эту точку называют *опорной*) и описывает форму небольшого участка поверхности модели. Идея спинового изображения заключается в сопоставлении опорной точке цилиндрической системы координат без использования полярного угла. Для каждой точки поверхности модели из окрестности опорной вычисляются относительные цилиндрические координаты  $\alpha$  и  $\beta$  (Рис. 2).  $\alpha$  и  $\beta$  находятся по следующим формулам

$$\alpha = \sqrt{\|\mathbf{X} - \mathbf{P}\|^2 - (\mathbf{n} \cdot (\mathbf{X} - \mathbf{P}))^2}, \beta = \mathbf{n} \cdot (\mathbf{X} - \mathbf{P})$$

Окрестность опорной точки разбивается на так называемые корзины по значениям координат  $\alpha$  и  $\beta$  таким образом, что точки с малоотличимыми относительными координатами попадают в одну и ту же корзину. Полученные корзины имеют форму трёхмерных колец (Рис. 3).

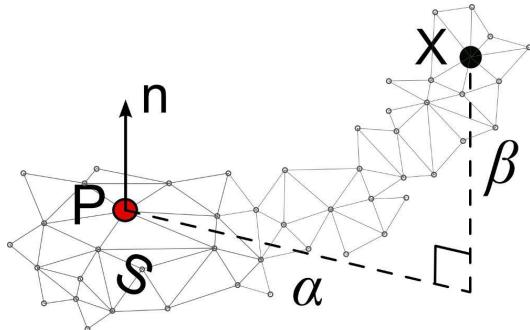


Рис 2: Относительные цилиндрические координаты  $\alpha$  и  $\beta$ .  $P$  – опорная точка,  $n$  – вектор нормали,  $X$  – точка модели,  $S$  – часть поверхности модели.

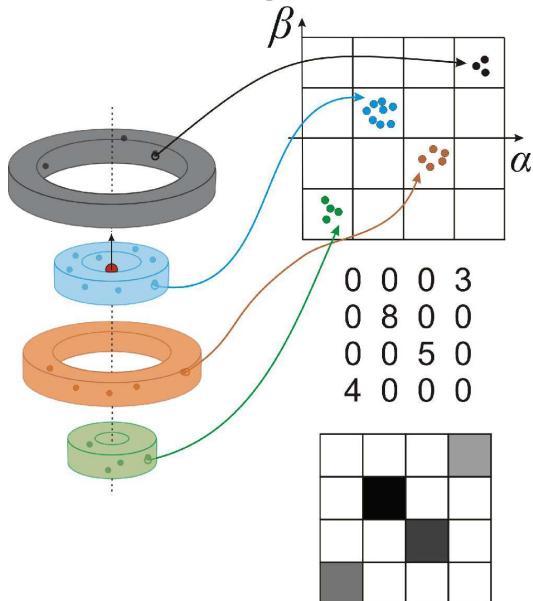


Рис 3: Корзины спинового изображения. Точки с похожими относительными координатами попадают в одну и ту же корзину (трёхмерное кольцо) – элемент двумерной целочисленной матрицы. Графическим представлением спинового изображения является изображение в градациях серого, цвет каждого пикселя которого определяется соответствующим элементом матрицы.

Положение каждой корзины в пространстве задаётся двумя индексами. Спиновое изображение представляет собой двумерную матрицу корзин, элементом которой является количество точек попавших в соответствующую корзину.

Спиновые изображения обладают отличными описательными характеристиками и являются инвариантными преобразованиям вращения и трансляции.

## 3. ИНТЕГРАЛЬНОЕ СПИНОВОЕ ИЗОБРАЖЕНИЕ

На основе идеи спиновых изображений могут быть построены глобальные дескрипторы поверхности. В данном случае мы не должны ограничиваться окрестностью опорной точки, а, наоборот, должны использовать все точки модели (Рис. 4).

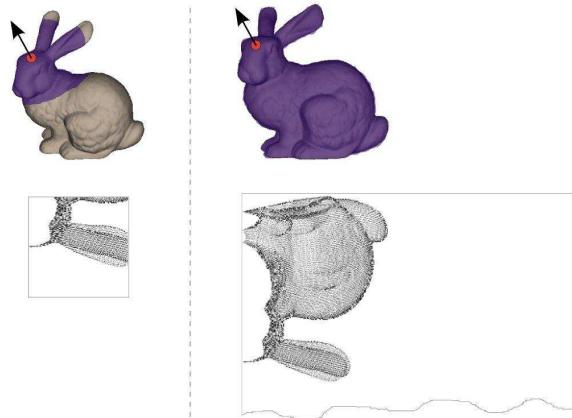


Рис 4: Обычное (слева) и интегральное (справа) спиновые изображения. Первое описывает только часть поверхности модели, второе – форму всей модели.

Для вычисления обычного спинового изображения необходимы координаты опорной точки и вектора нормали к поверхности модели в этой точке. При этом опорной точкой является одна из точек модели. Выбор опорной точки и вектора при построении глобального дескриптора интегрального спинового изображения представляет собой более сложную задачу. Очевидно, что глобальные дескрипторы поверхности, вычисленные для идентичных моделей, должны быть совершенно одинаковые. Это условие является основополагающим при разработке метода сравнения моделей. Вычисление опорной точки и вектора для интегральных спиновых изображений должно быть основано на глобальных характеристиках модели, например, центр масс, главные оси и др. Ещё одна проблема, приводящая к ошибкам сравнения, связана с разным масштабом моделей. Поэтому оптимальное масштабирование моделей должно также предшествовать вычислению интегральных спиновых изображений. Детальное описание процедуры нормализации трёхмерных моделей, включающей масштабирование и вычисление опорной точки и вектора, приведено в следующей секции.

Интегральное спиновое изображение вычисляется аналогично описанной выше схеме для обычного спинового изображения, за исключением того, что используются специфические опорная точка и вектор, и что окрестность опорной точки выбирается таким образом, чтобы абсолютно все точки модели попали в неё. За счёт вычисления относительных координат  $\alpha$  и  $\beta$  для всех точек модели, интегральное спиновое изображение описывает форму всей поверхности модели. Преимуществами использования идеи спиновых изображений при построении глобальных дескрипторов поверхности

являются хорошие описательные характеристики, небольшие вычислительные затраты и доступность эффективного и простого алгоритма их сравнения. Метод сравнения трёхмерных моделей описан в секции 5.

## 4. НОРМАЛИЗАЦИЯ

Нормализация 3D модели для вычисления интегрального спинового изображения включает оптимальное масштабирование и нахождение опорной точки и вектора. Интегральные спиновые изображения нормализованных моделей могут непосредственно сравниваться друг с другом для выявления сходства соответствующих трёхмерных моделей.

Все этапы нормализации могут выполняться независимо друг от друга. Более того, нормализация одной модели происходит без всякого знания обо всех других моделях. Эти обстоятельства позволяют предварительно в режиме оффлайн провести нормализацию и вычисление дескрипторов для всех моделей из некоторой базы. Во время запроса останется только обработать одну модель и сравнить её дескриптор поверхности последовательно с дескрипторами всех моделей из базы.

Крайне необходимо, чтобы на этапе нормализации точки всех моделей были одинаковым образом и равномерно распределены вдоль поверхности. Для этого мы используем метод контроля разрешения поверхности, предложенный в [3]. Главным преимуществом этого метода является минимальное изменение формы поверхности в результате её обработки.

### 4.1 Оптимальное масштабирование

Две модели (два облака точек в трёхмерном пространстве) имеют одинаковый масштаб, если среднее отклонение точек каждой из них равно единице. Это утверждение доказано в [4]. На основании этого можно легко вычислить оптимальный коэффициент масштабирования для произвольного набора точек  $P = \{p_1, p_2, \dots, p_n\}$ . Для центрированного (центр масс в начале системы координат) набора точек  $P$  среднее отклонение находится по формуле

$$D = \frac{1}{n} \sum_{i=1}^n \|p_i\|^2.$$

Оптимальный коэффициент масштабирования вычисляется как

$$s = \frac{1}{\sqrt{D}}.$$

Оптимальное масштабирование модели достигается путём умножения координат всех её точек на коэффициент масштабирования.

### 4.2 Опорная точка

В случае обычного спинового изображения опорной точкой является одна из точек поверхности модели. Спиновое изображение выстраивается относительно этой точки и описывает форму небольшого участка поверхности (в окрестности опорной точки). Как правило, спиновые изображения вычисляются для всех или для подмножества точек модели. К примеру, для решения задачи совмещения поверхностей спиновые изображения одной поверхности последовательно сравниваются со всеми спиновыми изображениями другой поверхности. Затем определяются лучшие соответствия между точками этих поверхностей. На основе пар соответствующих точек находится наилучшее

преобразование одной поверхности к другой. В системе поиска трёхмерных моделей интегральное спиновое изображение вычисляется один раз для каждой модели в единственной опорной точке. Таким образом, опорная точка для каждой модели должна определяться одинаковым способом. Целесообразно в качестве опорной точки для интегрального спинового изображения выбрать центр масс 3D модели (среднее арифметическое координат всех точек модели). Очевидно, что у похожих моделей центры масс расположены близко друг к другу. И, следовательно, интегральные спиновые изображения, построенные в этих точках с верно выбранными векторами, также похожи. В следующей подсекции представлен метод для определения опорного вектора.

### 4.3 Опорный вектор

Обычное спиновое изображение строится для точек поверхности в окрестности опорной точки, а вектор нормали к поверхности в опорной точке используется для определения относительных координат. В случае интегрального спинового изображения опорная точка не имеет вектора нормали, так как не лежит на поверхности. Одним из способов определения вектора для интегрального спинового изображения (назовём этот вектор *опорным*) заключается в использовании информации о главных осях модели. Методом для получения такой информации является известный метод главных компонент (Principle Component Analysis – PCA). В данном методе определение главных компонент многомерных данных сводится к вычислению собственных значений и собственных векторов ковариационной матрицы этих данных. В нашем случае данными является модель, представленная облаком точек в трёхмерном пространстве. Элементом ковариационной матрицы  $C$  для центрированного облака точек в трёхмерном пространстве  $P = \{p_1, \dots, p_n\} \subset R^3$  будет

$$C_{ij} = \frac{1}{n-1} \sum_{k=1}^n p_k^{(i)} * p_k^{(j)},$$

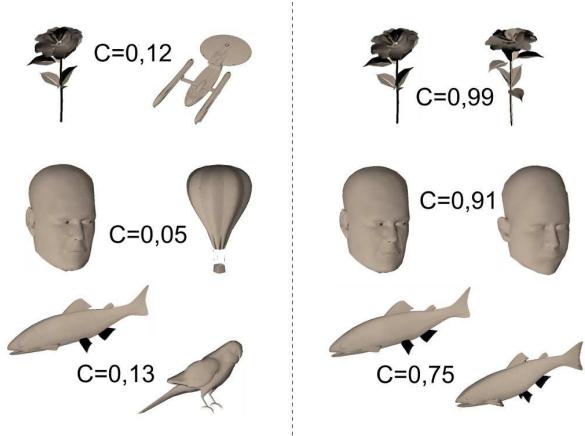
где  $p_k^{(i)}$  и  $p_k^{(j)}$  –  $i$ -ая и  $j$ -ая координаты точки  $p_k$ .

Матрица  $C$  является симметричной матрицей размерности три. Её собственные вектора (главные оси) образуют ортогональную тройку векторов и соответствуют распределению точек модели вдоль каждой координаты пространства. Таким образом, похожие модели имеют похожие главные оси. В качестве опорного вектора для построения интегрального спинового изображения удобно выбирать собственный вектор матрицы  $C$ , соответствующий наибольшему собственному значению. Такой подход является весьма надёжным и всегда приводит к правильному выбору опорного вектора, за исключением абсолютного равенства собственных значений ковариационной матрицы.

## 5. СРАВНЕНИЕ

Интегральные спиновые изображения представляют собой двумерные целочисленные матрицы. Так как глобальные дескрипторы содержат информацию о форме всей поверхности модели, сходство дескрипторов означает сходство соответствующих моделей. Наиболее простым и одновременно эффективным методом сравнения таких данных является корреляция. Для сравнения двумерных матриц вполне подходит линейный коэффициент корреляции. Несколько пар моделей из базы Princeton Shape Benchmark [5] с посчитанными коэффициентами

корреляции между их интегральными спиновыми изображениями представлены на Рис. 5.



**Рис 5:** Пары моделей различной формы (слева) и пары моделей похожей формы (справа). С – линейный коэффициент корреляции, вычисленный для интегральных спиновых изображений соответствующих моделей.

Предложенный метод сравнения достаточно хорошо различает трёхмерные модели и позволяет адекватно осуществлять поиск моделей в крупных базах.

## 6. ЗАКЛЮЧЕНИЕ

Предложенные в статье глобальные дескрипторы поверхности интегральные спиновые изображения сохранили все полезные качества своих локальных предшественников и оказались вполне пригодными для интегрального представления формы трёхмерной модели. В силу своей специфики невозможно непосредственно вычислить интегральное спиновое изображение, поэтому необходима процедура нормализации модели. Разработанная процедура обеспечивает эффективную и оптимальную нормализацию 3D моделей. Метод сравнения интегральных спиновых изображений основан на линейном коэффициенте корреляции. Он успешно определяет сходство моделей и выполняется за приемлемое время. В дальнейших исследованиях мы планируем улучшить интегральные спиновые изображения, используя идеи предложенных ранее модификаций спиновых изображений: трёхмерные и ориентированные спиновые изображения [6]. Планируется уделить внимание проблеме индексации баз трёхмерных моделей в целях ускорения работы системы поиска моделей, основанной на интегральных спиновых изображениях, а также вопросу обработки анизотропных моделей. Сравнение предложенных методов с лучшими из известных и апробация на других базах трёхмерных моделей также являются важными направлениями наших исследований.

## 7. ССЫЛКИ

- [1] Correa, S., Shapiro, L. *A new signature-based method for efficient 3-D object recognition*. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 1, pp. 769-776, 2001.
- [2] Johnson A.E., Hebert M. *Using spin images for efficient object recognition in cluttered 3D scenes*. IEEE Trans. Pattern Analysis and Machine Intelligence, 21(5), pp. 433-449, 1999.

- [3] Johnson A.E. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Carnegie Mellon University, 1997.
- [4] Kazhdan M. *Shape Representations And Algorithms For 3D Model Retrieval*. PhD thesis, Princeton University, 2004.
- [5] Princeton Shape Benchmark. <http://shape.cs.princeton.edu/benchmark/>
- [6] Крыловецкий А.А., Черников И.С. *Модифицированные спиновые изображения в системах трехмерной реконструкции*. Телематика'2010 : тр. XVII Всерос. науч.-метод. конф. Санкт-Петербург. 21-24 июня 2010 г.

## Об авторах

Александр Крыловецкий – доцент кафедры цифровых технологий ВГУ. Его адрес: [aakryl@sc.vsu.ru](mailto:aakryl@sc.vsu.ru).  
Игорь Черников – аспирант кафедры цифровых технологий ВГУ. Его адрес: [ichernikov@yandex.ru](mailto:ichernikov@yandex.ru).

# Регуляризованный по Тихонову биспектральный вариационный метод обработки оптических изображений

А.А. Потягалова<sup>1</sup>, А.В. Ларичев<sup>2</sup>, А.В. Разгулин<sup>3</sup>, Н.Г.Ирошников<sup>4</sup>

<sup>1</sup> ABBYY, [Anastasiya\\_Poty@abbyy.com](mailto:Anastasiya_Poty@abbyy.com)

<sup>2</sup> МГУ им. М.В. Ломоносова, физический факультет, [larichev@optics.ru](mailto:larichev@optics.ru)

<sup>3</sup> МГУ им. М.В. Ломоносова, факультет ВМиК, [razgulin@cs.msu.su](mailto:razgulin@cs.msu.su)

<sup>4</sup> МГУ им. М.В. Ломоносова, физический факультет, [nikita@optics.ru](mailto:nikita@optics.ru)

## АННОТАЦИЯ

Для решения задач восстановления изображений в условиях сильных фазовых искажений предлагается использовать регуляризованный по Тихонову биспектральный вариационный метод. Он сочетает такие преимущества биспектрального подхода, как возможность сохранения и восстановления фазовой информации, инвариантность к случайным сдвигам исходного сигнала, нетребовательность к количественной априорной информации о

статистических свойствах наблюдаемых сигналов. В комбинации с регуляризованным по Тихонову вариационным методом, который позволяет устойчиво обрабатывать изображения больших размеров, получается достаточно эффективный метод восстановления изображений. Приведенные в работе результаты тестирования метода в случае атмосферных и водных фазовых искажений выявили его преимущества перед традиционно используемым рекурсивным биспектральным методом.

## 1. ВВЕДЕНИЕ

Во многих прикладных задачах оценивания параметров изображений процесс обработки наблюдаемых изображений сопровождается значительными трудностями при извлечении исходной информации, которая искажена помехами различного происхождения и уровня. При этом статистические характеристики шума часто априорно полностью не определены или определены частично и не точно [1]. Данная ситуация типична для прикладных задач обнаружения и распознавания сигналов в радиолокации, гидролокации, оптической и радио астрономии, а также в системах дистанционного зондирования и в диагностических системах обработки медицинских сигналов. Поэтому, практическая реализация разнообразных методов фильтрации [2] и восстановления изображений в шумах сопряжена с постоянно растущим интересом для широкого круга специалистов в различных областях.

Среди традиционных подходов, обеспечивающих подавление помех при решении задач обработки изображений, используются методы винеровской и инверсной фильтрации [3,4], методы линейной и нелинейной фильтрации с помощью двумерных и одномерных скользящих окон [5,6], аддитивные алгоритмы восстановления изображений [7]. Эффективность первой группы методов в значительной степени зависит от полноты априорных сведений о статистике сигналов и помех, которые на практике имеются далеко не всегда. Подходы, использующие теорию линейной фильтрации, в ряде случаев позволяют реализовать оптимальные по критерию минимума среднеквадратической ошибки практические алгоритмы обработки изображений при условии, например,

нормального распределения аддитивного шума. Однако, в условия воздействия помех с распределением, отличным от нормального, например, в условиях воздействия смешанного шума, который может включать в себя аддитивную Гауссову компоненту и импульсный шум, лучшие по отношению к методу линейной фильтрации результаты обеспечивают методы и алгоритмы нелинейной фильтрации. Данная группа методов, несмотря на характерные для них специфические динамические искажения, обеспечивает успешное решение задачи подавления смешанного (негауссова) шума в условиях полной или частичной априорной неопределенности в отношении свойств шумов за счет использования какого-либо фильтра из немалого числа разнообразных сглаживающих фильтров, разработанных к настоящему времени. Однако эффективность этой группы методов падает при обработке "импульсных" сигналов, то есть сигналов, представляющих собой набор импульсов, длительность которых сравнима с длительностью помеховых импульсных выбросов, а также в случаях, когда для нескольких наблюдаемых реализаций обрабатываемых сигналов имеют место случайные сдвиги информационной составляющей, которые могут быть, например, обусловлены влиянием турбулентности канала распространения сигналов. Стоит также отметить, что границы применимости традиционных методов фильтрации и восстановления сигналов сильно зависят от отношения сигнала к шуму на входе измерительной системы. В большинстве перечисленных методов требуемую на практике точность оценивания параметров сигнала удается реализовать только для

отношений сигнал/шум, значительно превосходящих единицу.

Для целого ряда важных прикладных задач обработки сигналов и изображений, например, для задач обнаружения слабых целей в радиолокации и в гидролокации, а также при получении изображений объектов малой интенсивности в астрономии, отношение сигнал/шум не превышает единицу или имеет значение, ненамного превышающее единицу. К одному из перспективных подходов, который обеспечивает хорошие результаты при решении задач восстановления изображений в таких условиях, относится подход, основанный на использовании корреляционных функций третьего порядка и биспектров [8, 9].

Напомним, что корреляцией двух функций  $f(x, y)$  и  $g(x, y)$  называется функция

$$f(x, y) \otimes g(x, y) = \int f(\xi, \eta) g(x + \xi, y + \eta) d\xi d\eta.$$

Тройной корреляцией функции  $o(x, y)$  называется функция

$$o^{(3)}(x_1, y_1, x_2, y_2) = \int o^*(x, y) o(x + x_1, y + y_1) o(x + x_2, y + y_2) dx dy. \quad (1)$$

Биспектром функции называется преобразование Фурье от её тройной корреляции:

$$\begin{aligned} O^{(3)}(u_1, v_1, u_2, v_2) \\ = \mathcal{F}\{o^{(3)}(x_1, y_1, x_2, y_2)\}. \end{aligned} \quad (2)$$

Несмотря на сложность вычисления тройных корреляционных функций и биспектров по сравнению с обычными корреляционными функциями и энергетическими спектрами, моменты третьего порядка отличаются несколькими принципиальными преимуществами:

1. Тройная корреляционная функция позволяет сохранить комплексный Фурье-спектр сигнала, и, следовательно, восстановить информацию о фазовых характеристиках исходного изображения.
2. Тройная корреляционная функция Гауссова шума стремится к нулю, что обеспечивает в ряде практических приложений рабочность алгоритмов восстановления негауссовых сигналов на фоне Гауссовых шумов.
3. Биспектр инвариантен к смещению исходного сигнала.

Благодаря перечисленным выше преимуществам тройные корреляционные функции и биспектры успешно используются для обработки сигналов и изображений в гидролокации, астрономии ([9-12]), оптике ([13]), в системах обработки биомедицинских сигналов ([14]), в системах дистанционного зондирования и др. Несмотря на это, ряд важных

вопросов, таких как улучшение оценок сигналов, восстановленных по оценкам биспектров, а также исследование устойчивости оценок биспектров к воздействию негауссовых шумов, например, смешанных шумов, остаются до сих пор недостаточно изученными, что вполне закономерно, так как до последнего времени не хватало технических возможностей для сбора данных и последующей их обработки. Например, в системах обработки астрофизических изображений необходимо обрабатывать большое (порядка  $10^3$ ) количество кадров с различными искажениями, а это стало возможным совсем недавно.

В свою очередь, биспектральные методы обработки изображений можно разделить на два класса, каждый из которых ориентирован на решение своего круга задач. К первому классу относится рекурсивный метод, основанный на непосредственном решении получающейся линейной алгебраической системы относительно фазовой функции биспектра. Он неплохо зарекомендовавший себя в работе с относительно небольшими изображениями, особенно при добавлении какого-либо сглаживающего фильтра. Второй класс, использующий вариационный подход и основанный на минимизации функционала невязки, дает более точно восстановленное изображение и, требуя меньших вычислительных затрат, может быть использован для обработки большого количества данных. Для борьбы с нежелательными искажениями этот метод применяется в сочетании с итеративной регуляризацией.

## 2. ПОСТАНОВКА ЗАДАЧИ

Для рассматриваемых в данной работе оптических систем формирования изображений характерна приведенная на рис. 1 схема преобразования изображений. Добавление помех  $p(x, y)$  в фазу исходной функции происходит уже в фокальной плоскости, так же, как и изменения, вносимые функцией зрачка  $M(x, y)$ . В плоскости изображения осуществляется преобразование Фурье над уже видоизмененным пространственно-частотным спектром и формируется распределение амплитуд света, описываемое уравнением:

$$\begin{aligned} i(x, y) &= \mathcal{F}\{M(x, y) \cdot \exp\{ip(x, y)\} \cdot \mathcal{F}(o(x, y))\} \\ &= o(x, y) * h(x, y). \end{aligned} \quad (3)$$

Таким образом, распределение комплексных амплитуд в выходной плоскости представляет собой свертку входного сигнала  $o(x, y)$  с импульсной характеристикой

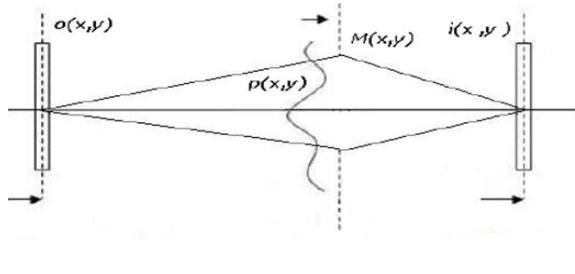


Рис. 1. Схема устройства пространственного преобразования изображений.

системы пространственных преобразований  $h(x, y) = \mathcal{F}\{M(x, y) \cdot \exp\{ip(x, y)\}\}$ . Искажения, вносимые функцией  $p(x, y)$  в систему, могут оказаться достаточно серьезными, поэтому одного изображения для восстановления исходного, как правило, оказывается недостаточно. В этой ситуации рассматривается серия искаженных изображений, алгоритм обрабатывает их по очереди, а потом усредняет результат.

### 3. ОБЩЕЕ ОПИСАНИЕ БИСПЕКТРАЛЬНОГО ПОДХОДА

Следуя [9] напомним основные этапы биспектрального подхода. После применения преобразования Фурье к соотношению (3) имеем:

$$I(u, v) = O(u, v) \cdot H(u, v), \quad (4)$$

где  $I(u, v) = \mathcal{F}\{i(x, y)\}$ ,  $H(u, v) = \mathcal{F}\{h(x, y)\}$ ,  $O(x, y) = \mathcal{F}\{o(x, y)\}$ . Для того чтобы восстановить функцию  $O(u, v)$  воспользуемся ее представлением через модуль  $|O(u, v)|$  и фазу  $\varphi(u, v)$

$$O(u, v) = |O(u, v)| \exp\{i\varphi(u, v)\}, \quad (5)$$

которое дает возможность восстановить модуль и фазу искомой функции параллельно. Модуль  $|O(u, v)|$  находится из соотношения (4):

$$|O(u, v)|^2 = \frac{|I(u, v)|^2}{|H(u, v)|^2 + \varepsilon},$$

где  $\varepsilon$  - малый параметр, подобранный для того, чтобы избежать деления на ноль. Отметим, что здесь и ниже при практической реализации алгоритма для серии изображений вместо  $|I(u, v)|^2$  и  $|H(u, v)|^2$  используются соответствующие усреднения по всей серии изображений.

Для восстановления фазы используется связь биспектральных функций исходного изображения, наблюдаемого изображения и аппаратной функции. С одной стороны, биспектральная функция (биспектр)  $O^{(3)}(u, u_j, v, v_j)$  вычисляется как образ Фурье тройной корреляции  $o^{(3)}(x, x_j, y, y_j)$  и, в силу (1) и (2), удовлетворяет тождеству

$$O^{(3)}(u, u_j, v, v_j) = O(u, u_j)O(v, v_j)O^*(u + u_j, v + v_j). \quad (6)$$

С другой стороны, из (4) вытекает соотношение  $I^{(3)}(u, u_1, v, v_1) = O^{(3)}(u, u_1, v, v_1) \cdot H^{(3)}(u, u_1, v, v_1)$ .

Учитывая (6) и представления

$$\begin{aligned} I^{(3)}(u, u_1, v, v_1) &= \\ |I^{(3)}(u, u_1, v, v_1)| \cdot \exp\{i\beta(u, u_1, v, v_1)\}, \\ O^{(3)}(u, u_1, v, v_1) &= |O^{(3)}(u, u_1, v, v_1)| \\ &\cdot \exp\{i\varphi(u, u_1, v, v_1)\}, \end{aligned}$$

в силу вещественности биспектра  $H^{(3)}(u, u_1, v, v_1)$  передаточной функции [9] получаем, что фазовые множители биспектра наблюдаемой функции и исходной функции совпадают:

$$\exp\{i\beta(u, u_1, v, v_1)\} = \exp\{\varphi(u, u_1) + \varphi(v, v_1) - \varphi(u + v, u_1 + v_1)\}. \quad (7)$$

Далее мы рассмотрим и сравним два метода нахождения искомой фазы  $\varphi(u, v)$  из уравнения (7) - рекурсивный и вариационный.

### 4. РЕКУРСИВНЫЙ БИСПЕКТРАЛЬНЫЙ МЕТОД

Рекурсивный метод восстановления фазы был известен еще в 70-х годах и основан на записи частного случая уравнения (7), отвечающего равенству фаз комплексных экспонент:

$$\beta(u, u_1, v, v_1) = \varphi(u, u_1) + \varphi(v, v_1) - \varphi(u + v, u_1 + v_1). \quad (8)$$

После дискретизации на сетке уравнение (8) приводится к системе линейных алгебраических уравнений, решение которой находится по явным формулам при некоторых дополнительных ограничениях. Кратко остановимся на реализации этого метода.

Для дискретизации вводится сетка с шагами

$\Delta u$ ,  $\Delta u_1$ ,  $\Delta v$ ,  $\Delta v_1$  по соответствующим переменным

$$\begin{aligned} u_p &= p \cdot \Delta u, & u_{p_1} &= p_1 \cdot \Delta u_1, & v_q &= q \cdot \Delta v, \\ v_{q_1} &= q_1 \cdot \Delta v_1, & p, p_1, q, q_1 \\ &= -N \dots N, \end{aligned}$$

а также сеточные функции

$$\begin{aligned} O_{p,p_1} &= \\ O(u_p, u_{p_1}), \quad O_{p,p_1,q,q_1}^{(3)} &= O_n^{(3)}(u_p, u_{p_1}, v_q, v_{q_1}), \\ \beta_{p,p_1,q,q_1} &= \beta(u_p, u_{p_1}, v_q, v_{q_1}), \quad \varphi_{p,q} = \varphi(u_p, v_q). \end{aligned}$$

Для введенных сеточных функций с учетом обозначений для  $r = p + q$ ,  $r_1 = p_1 + q_1$  соотношение для фаз (8) при  $q = 1$  принимает вид:

$$\varphi_{r,r_1} = \varphi_{1,1} + \varphi_{r-1,r_1-1} - \beta_{r-1,1,r_1-1,1}.$$

Так как  $O_n(u, v)$  - эрмитова функция и  $\varphi_{r,r_1} = \varphi_{-r,-r_1}$ , то  $\varphi_{0,0} = 0$ . Предполагая далее, что расположение восстановленного сигнала несущественно, получаем  $\varphi_{1,1} = 0$ . Тогда соотношение между значениями функций  $\beta$  и  $\varphi$  на рассматриваемой сетке принимает следующий вид:

$$\varphi_{2,1} = \varphi_{1,1} - \beta_{1,1,1,1},$$

$$\begin{aligned} \varphi_{3,1} &= \varphi_{1,1} + \varphi_{2,1} - \beta_{2,1,1,1} \\ &= 3\varphi_{1,1} - \beta_{2,1,1,1} - \beta_{1,1,1,1}, \end{aligned}$$

...

$$\varphi_{r,1} = r\varphi_{1,1} - \beta_{1,1,1,1} - \beta_{2,1,1,1} - \dots - \beta_{r-1,1,1,1}.$$

Отсюда находится фаза  $\varphi(u, v)$ , затем  $O(u, v)$  из (5), и, наконец, искомая функция  $o(x, y)$ :

$$o(x, y) = \mathcal{F}^{-1}\{O(u, v)\}.$$

Рекурсивный метод достаточно прост в реализации и неплохо зарекомендовал себя в работе с одномерными сигналами. В двумерном случае его практическое применение ограничено изображениями небольшого размера, поскольку с ростом размера изображений эффективность рекурсивного метода резко падает из-за медленной работы. Точность вычислений, проведенных рекурсивным методом, как правило, ниже, чем точность вычислений, проведенных вариационным методом. Результаты сравнения работы рекурсивного метода и вариационного приведены в следующем разделе.

## 5. ВАРИАЦИОННЫЙ МЕТОД

Рекурсивный биспектральный метод основан на уравнении (8), которое записано относительно фазовых функций и требует вычисления фазы  $\beta(u, u_1, v, v_1)$  биспектра наблюдаемого сигнала. В условиях существенных искажений этап выделения этой фазы может сопровождаться дополнительными вычислительными неустойчивостями. Чтобы уменьшить негативное влияние этих неустойчивостей и одновременно получить возможность обрабатывать изображения большого размера или кадры с большим количеством искажений, будем работать не с фазами, но с фазовыми множителями в (8). Кроме того, вместо решения уравнения (8) для фазовых множителей соответствующих биспектров поставим вариационную задачу нахождения минимума функционала невязки, который описывает среднеквадратичное отклонение комплексных экспонент по узлам сетки:

$$E(\varphi) = \sum_{ijkl} \left[ (\operatorname{Re} \Delta_{i,j,k,l})^2 + (\operatorname{Im} \Delta_{i,j,k,l})^2 \right] \rightarrow \min,$$

где  $\Delta_{i,j,k,l} = \exp\{i\beta_{i,j,k,l}\} - \exp\{i(\varphi_{i,j} + \varphi_{k,l} - \varphi_{i+k,j+l})\}.$

При выборе подходящего метода минимизации с учетом размерности задачи оказывается, что градиентный метод является наиболее приемлемым. Обладая меньшей скоростью сходимости по сравнению квазиньютоновскими методами, градиентный метод не требует вычисления и хранения матрицы вторых производных большого размера, так что он хорошо подходит для рассматриваемого класса задач.

Градиентный метод состоит в построении минимизирующей функционал  $E(\varphi)$  последовательности  $\varphi^{(k)} = \{\varphi_{i,j}^{(k)}\}$  по следующему правилу ([15]):

$$\varphi^{(k+1)} = \varphi^{(k)} - b_k \nabla E(\varphi^{(k)}), \quad b_k > 0, \quad k = 0, 1, \dots$$

В приложениях используется также метод сопряженных градиентов [13]; различные правила останова градиентного метода приведены в [15]. Численное исследование выявило неустойчивость алгоритма относительно искажений входных данных.

Методы решения неустойчивых задач минимизации строятся на основе обычных методов

минимизации, подвергнув их некоторой процедуре регуляризации [15]. Для построения регуляризованных методов часто поступают следующим образом: в общей схеме конкретного метода минимизации вместо целевого функционала используют функционал Тихонова

$$T(\varphi) = E(\varphi) + \alpha_k \|\varphi\|^2 \rightarrow \min,$$

то придёт к регуляризованному градиентному методу

$$\varphi^{(k+1)} = \varphi^{(k)} - b_k (\nabla E(\varphi^{(k)}) + 2\alpha_k \varphi^{(k)}), \quad k = 0, 1, \dots, \quad (9)$$

в котором нужно согласованно выбрать параметры  $\alpha_k, b_k$ . В качестве способа согласования параметров метода, входящих в функцию Тихонова, будет использован принцип итеративной регуляризации. В качестве примера последовательностей  $\alpha_k$  и  $b_k$ , удовлетворяющих требуемым условиям, можно взять последовательности

$$b_k = (1 + k)^{-1/2}, \quad \alpha_k = (1 + k)^{-p}, \quad 0 < p < 0,5.$$

После нахождения искомой фазы  $\varphi$ , доставляющей приближенный минимум функционалу Тихонова  $T(\varphi)$ , вычисляется функция  $O(u, v)$  и после применения обратного преобразования Фурье находится искомая функция объекта  $o(x, y)$ .

Как было отмечено выше, рекурсивный метод восстановления достаточно хорошо при восстановлении небольших изображений и сигналов, но градиентный метод восстанавливает искаженные изображения и сигналы точнее. Рекурсивный метод восстановления изображений достаточно хорошо устраняет импульсные шумы на изображениях небольшого размера, хоть и работает немного дольше, чем биспектральный вариационный метод. Типичные сравнительные результаты работы методов приведены на рис. 2.

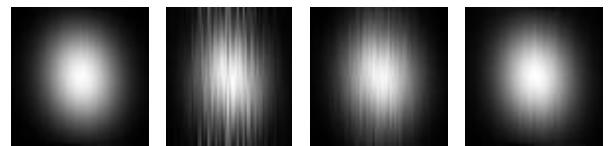


Рис. 2. Сравнение рекурсивного и вариационного

методов:

- i. исходное изображение без искажений;
- ii. искаженное изображение;
- iii. результат работы рекурсивного метода;
- iv. результат работы вариационного метода.

## 6. ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ ВОССТАНОВЛЕНИЕ ИЗОБРАЖЕНИЙ БИСПЕКТРАЛЬНЫМ ВАРИАЦИОННЫМ МЕТОДОМ

В данном параграфе приводятся результаты работы предлагаемого вариационного метода для различных типов искажений.

**6.1. Восстановление изображения в условиях атмосферных искажений** Одним из наиболее распространенных применений описываемого метода является астрофизика, поэтому первое из тестирований проводилось на изображениях с атмосферными искажениями, представленными на рис. 3. В качестве исходного изображения использовалось характерное для астрофизики изображение, состоящее из двух ярких пятен (рис. 4-iv). Серия искажений накладывалась на исходное изображение, создавая серию зашумленных искажений (рис. 4-i), которые подавались на вход вариационному методу. Искаженные изображения в рассматриваемом примере имеют существенные качественные отличия от исходных изображений – на них наблюдаются несколько ложных максимумов световой амплитуды. Таким образом, одним из главных субъективных критериев качества работы метода восстановления

является его способность правильно восстановить искомые максимумы и одновременно подавить ложные максимумы.

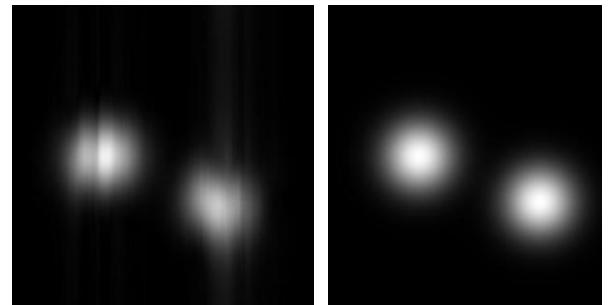
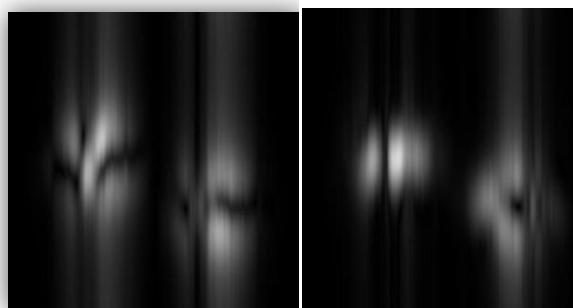
Как видно из рис. 4-ii простое усреднение не дает желаемого результата, позволяя лишь отчасти решить поставленную задачу. Представленный же на рис. 4-iii результат работы вариационного метода (9) с выбором параметров итеративной регуляризации

$$\beta_k = (1 + k)^{-1/2}, \quad \alpha_k = (1 + k)^{-1/4}$$

существенно отличается в лучшую сторону как хорошим качеством восстановления искомых максимумов амплитуды, так и исходного изображения в целом.



Рис. 3. Серия атмосферных искажений



iii. iv.

Рис. 4. Восстановление изображения с атмосферными искажениями:

- i. одно изображение из серии искаженных изображений;
- ii. простое усреднение серии изображений;
- iii. результат работы вариационного метода;
- iv. исходное изображение без искажений.

## 6.2. Восстановление изображения в условиях водных искажений

В приложениях часто встречается задача восстановления изображений с различными волновыми искажениями, например, движущимися волнами воды. Такого рода помехи можно наблюдать в задачах гидролокации. Таким образом, в искаженном изображении присутствует эффект размытия и частично эффект потери цвета. При этом в качестве входных данных используется серия искаженных изображений, а сами искажения имеют периодическую структуру. Для численного исследования эффективности нашего вариационного метода и сравнения с известными результатами воспользуемся моделью волновых искажений из статьи [16]. При моделировании таких искажений сдвиг каждой точки, которая удаляется от своей исходной позиции, происходит в соответствии со следующими правилами:

$$d_x = hp \left( 1 - \frac{1}{n} \right) + N \quad d_y = hq \left( 1 - \frac{1}{n} \right) + N$$

где  $h$  - расстояние между водной поверхностью и дном,  $n$  - коэффициент отражения воды,  $N$  - белый шум, а  $p$  и  $q$  - предполагаемые сдвиги волн воды с различными скоростями и длинами, которые определяются следующим образом:

$$p = - \sum a_i u_i \cos\{u_i x + v_i y - \omega_i t\},$$

$$q = - \sum a_i v_i \cos\{u_i x + v_i y - \omega_i t\}.$$

В результате получаются различные виды волн, отличающиеся по глубине и размытости. В такой модели есть некоторые допущения. Первое состоит в том, что часть водной поверхности является неподвижной и плоской. Второе - среднее искажение в любой точке водной поверхности равно нулю, при условии достаточно долгого наблюдения. Оба допущения вполне соответствуют поставленной задаче. Заметим, что при работе с реальными данными появляются некоторые другие помехи - например, искажения, добавляемые ветром или искажения волн воды от упавших объектов.

Рассмотрим примеры работы регуляризованного вариационного метода на примере двух изображений с волновыми искажениями – монохромного текста (рис. 5) и цветного изображения (рис. 6). На рис. 5-i представлен характерный вид текстового изображения, зашумленного волновыми искажениями. Наряду с геометрическими искажениями текста присутствуют заметные нерегулярные шумы. Такая ситуация возникает, например, в задаче восстановления текста на поверхности, находящейся под слоем водной поверхности в ветреную погоду. Другой причиной таких искажений может являться конвективное движение нагретых слоев воздуха (например, над шоссе), приводящее к существенным оптическим искажениям. Серия изображений, приведенная на рис. 5, является иллюстрацией устранения волновых искажений с помощью биспектрального вариационного метода. Результат усреднения серии искаженных изображений (рис. 5-ii) хоть и менее зашумлен, чем исходное изображение (рис. 5-iv), но заметно уступает по точности изображению (рис. 5-iii), восстановленному биспектральным вариационным методом (9) с параметрами итеративной регуляризации

$$\beta_k = (1 + k)^{-1/2}, \alpha_k = (1 + k)^{-1/3}.$$

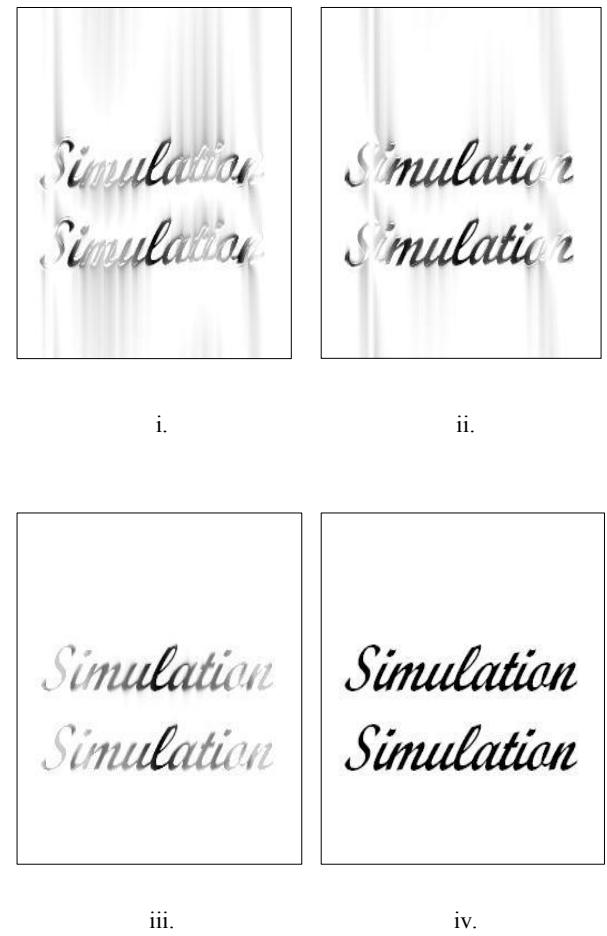


Рис. 5. Восстановление текстового изображения с волновыми искажениями:  
i. одно изображение из серии искаженных изображений;  
ii. результат работы рекурсивного метода;  
iii. результат работы вариационного метода;  
iv. исходное изображение без искажений.

Интересно сравнить качество восстановления изображений с одинаковым типом искажений регуляризованным вариационным методом с рекурсивным методом [16]. Отметим, что рекурсивным метод генерирует изображение, которое является более точным по сравнению с любым изображением, полученным путем усреднения, но уступает по точности результату, полученному при помощи вариационного регуляризованного метода, даже с учетом того, что при восстановлении рекурсивным методом использовалась серия из 100 изображений, а при использовании вариационного – только 10.

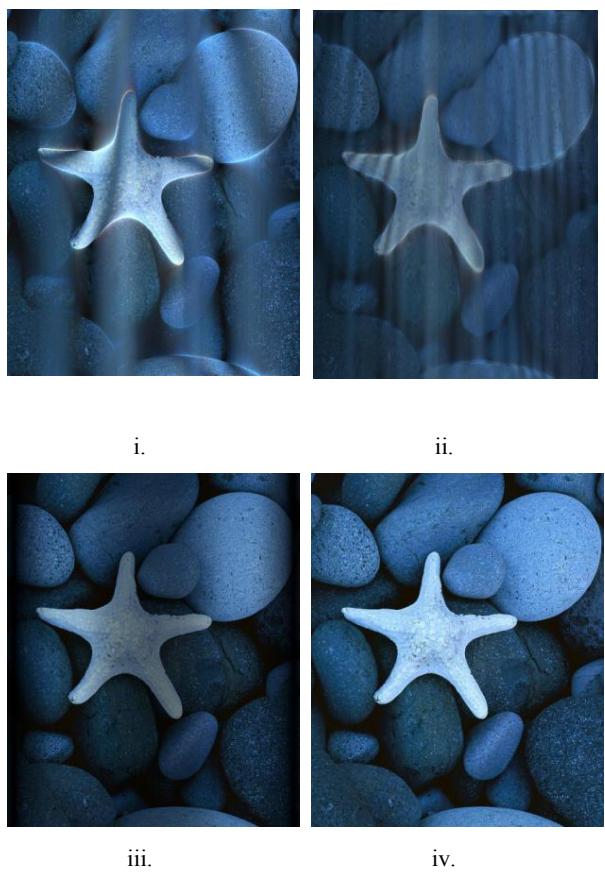


Рис. 6. Восстановление цветного изображения с водными искажениями:

- i. одно изображение из серии с водными искажениями;
- ii. результат работы рекурсивного метода;
- iii. результат работы вариационного метода;
- iv. исходное неискаженное изображение.

Ещё один пример работы с водными искажениями, приведенный на рис. 6, иллюстрирует работу вариационного регуляризованного метода в случае цветных изображений. Изображение, восстановленное регуляризованным методом (рис. 6-iii), более точно передает цвет исходного изображения, чем изображение, полученное в результате усреднения серии искаженных изображений (рис. 6-ii). Особенность данного примера состоит в том, что на нем заметно значительное улучшение контрастности по сравнению с искаженным изображением.

## 7. ЗАКЛЮЧЕНИЕ

Традиционный метод восстановления сигналов по оценкам биспектров является перспективным инструментом обработки изображений вследствие возможности сохранения и восстановления фазовой

информации, из-за низкой чувствительности к аддитивным шумам, а также в силу инвариантности к случайнм сдвигам исходного сигнала. Кроме того, биспектральный метод восстановления сигналов не требует количественной априорной информации о статистических свойствах наблюдаемых процессов. Естественно, эффективность биспектрального анализа во многих практических ситуациях снижается, например, когда отношения сигнал/шум (для случая Гауссовых помех) меньше единицы или когда статистика аддитивного шума отличается от нормального закона распределения. Однако и в этих случаях применение биспектральных и предложенных регуляризованных по Тихонову вариационных биспектральных методов обработки приводит к заметному улучшению качества восстановления изображений. При этом предложенные методы восстановления изображений сохраняют все отмеченные выше преимущества биспектрального анализа. Результаты численного моделирования и проведенный сравнительный анализ демонстрируют улучшение устойчивости (по сравнению с традиционной биспектральной обработкой) предложенных комбинированных методов восстановления сигналов. Предложенный подход позволяет реализовать метод восстановления изображений, в котором интегрированы преимущества биспектрального анализа и достоинства регуляризованных методов оптимизации. Предложенный комбинированный метод может быть использован для обработки изображений и сигналов в системах, подверженных воздействию интенсивных шумов в условиях априорной неопределенности о количественных статистических характеристиках сигналов и шумов, например, для радиолокационных или гидролокационных систем, для оптических систем восстановления астрономических изображений и систем обнаружения малоразмерных объектов.

**СПИСОК ЛИТЕРАТУРЫ**

1. О'Нейл Э. Введение в статистическую оптику. М., 1966.
2. Гудмен Дж. Введение в фурье-оптику. М.: Мир, 1970.
3. Andrews H.C., Hunt B.R. Digital image restoration // Prentice-Hall, Englewood Cliffs, NJ, 1977.
4. Frieden B.R. Image enhancement and restoration, in Picture Processing and Digital Processing // T.S. Huang, Ed. Berlin: Springer-Verlag, pp. 177-248 (1975).
5. Astola J., Kuosmanen P. Fundamentals of Nonlinear Digital Filtering // CRC Press LLC, N.Y. Boca Raton, 1997.
6. Pitas I., Venetsanopoulos A. N. Nonlinear Digital Filters: Principles and Applications. Kluwer Academic Publishers, Boston, 1990.
7. Zervakis M.E., Venetsapoulos A.N. A class of noniterative estimators for nonlinear image restoration // IEEE Trans. Circ. Syst. CAS-38, pp. 731-744 (1991).
8. Bartelt H., Lohmann A.W., and Wirnitzer B. Phase and amplitude recovery from bispectra // Applied Optics. Vol. 23, pp. 3121-3129 (1984).
9. Lohmann A.W., Weigelt G., and Wirnitzer B. Speckle masking in astronomy triple correlation theory and applications // Applied Optics. Vol.22, pp. 869-891 (1983).
10. Wirnitzer B. Bispectral analysis at low light levels and astronomical speckle masking // J. Opt. Soc. Vol.2, No.1, pp. 14-21 (1985).
11. Kuwamura S., Yoshinoya Y., Miura N., Tsumuraya F., Sakamoto M., and Baba N. Tomographic implementation of astronomical speckle imaging from bispectra // Optical Review. Vol. 18, No. 1, pp. 19–26 (2011).
12. Negrete-Regagnon P. Practical aspects of image recovery by means of the bispectrum // Optical Society of America. Vol. 13, No. 7, pp. 1557-1577 (1996).
13. Miller D.T., Chop D.J., Morris G.M., and Williams D.R. Bispectral imaging through unknown deterministic aberrations // J. of Modern Optics, Vol. 42, No. 7, pp. 1523-1546 (1995).
14. Sasaki K., Sato T., Nakamura Y. Holographic passive sonar // IEEE Trans. Sonics Ultrasonics, SU-24, pp. 193-200 (1977).
15. Васильев Ф.П. Методы оптимизации. М.: Факториал Пресс, 2002.
16. Wen Z., Li H., Fraser D., Lambert A. Reconstruction of underwater image by bispectrum // IEEE International Conference on Image Processing, ICIP, September 16-19, 2007.

# Система интерактивного расчета глобального освещения для гибридных сцен

Денис Боголепов<sup>1</sup>, Дмитрий Сопин<sup>1</sup>, Данила Ульянов<sup>2</sup>, Вадим Турлапов<sup>1</sup>

<sup>1</sup>Нижегородский государственный университет им. Н.И. Лобачевского

<sup>2</sup>Нижегородский государственный технический университет им. Р.Е. Алексеева

denisbogol@gmail.com, sopindm@gmail.com, danila-ulyanov@ya.ru

## Аннотация

Разработана система интерактивной визуализации для сцен гибридной природы с поддержкой глобального освещения и динамической геометрии. Под гибридностью понимается сочетание полигональных моделей с объектами других типов, таких как поверхности второго порядка, неявно заданные функции, фрактальные множества и объемные данные. В основе системы – конвейер трассировки лучей, который может быть сконфигурирован для задач с различным балансом скорости и качества визуализации. Для ускорения вычислений используется SAH BVH дерево, которое строится на ГПУ в реальном времени. Прототип системы разработан на базе NVIDIA CUDA и обеспечивает высокую производительность визуализации гибридных сцен.

**Ключевые слова:** глобальное освещение, двунаправленная трассировка путей, GPU, GPGPU, CUDA.

## 1. ВВЕДЕНИЕ

Трассировка лучей – метод построения изображений компьютерных сцен путем отслеживания взаимодействий лучей света с объектами сцены. Классическая трассировка лучей Уиттеда является простейшим алгоритмом глобального освещения, поскольку позволяет моделировать эффекты затенения, идеального отражения и преломления. Однако реалистичная визуализация невозможна без расчета вторичного освещения, типичными примерами которого являются каустики и перенос цвета между соседними поверхностями. Распределенная трассировка лучей Кука добавила в процесс расчета генерацию случайных направлений, что позволило учесть некоторые эффекты вторичного освещения, включая ограниченный перенос цвета. Данный метод был обобщен в стохастической трассировке путей Кайя, которая позволила учесть все пути света в сцене. Однако сходимость изображения достигается медленно и требует большого объема вычислений. Дальнейшим развитием метода является двунаправленная трассировка путей, основная идея которой состоит в единовременном испускании лучей из источников света и объектива виртуальной камеры. Данный подход позволяет значительно повысить скорость сходимости при расчете вторичного освещения.

Методы на основе трассировки лучей традиционно использовались в компьютерной графике для синтеза изображений высокого качества. Параллельная природа задачи и непрерывный рост вычислительных возможностей со временем вывели данные методы за рамки медленной предварительной визуализации. Первые системы интерактивной трассировки лучей требовали либо высокопроизводительных суперкомпьютеров [1]–[3], либо были ограничены небольшими сценами [4] и простыми эффектами освещения [5]. Сегодня существуют интерактивные системы визуализации, которые на обычном потребительском оборудовании

позволяют решать задачи, практически недоступные традиционному алгоритму растеризации [6]–[9].

Для повышения производительности трассировки лучей используются ускоряющие структуры. Долгое время основным показателем их эффективности служила скорость визуализации, при этом трудоемкость построения структур практически игнорировалась. В результате первые интерактивные системы трассировки позволяли обрабатывать сцены только со статической геометрией. Подобная ситуация привела к пересмотру ускоряющих структур, направленному на разработку алгоритмов быстрого построения или обновления при сохранении высокой скорости визуализации. Исследования показали, что поддержка динамических сцен возможна для всех основных структур, включая регулярные и иерархические сетки,  $k$ -деревья и иерархии ограничивающих объемов [10].

Наряду с динамическими сценами значительное внимание получило интерактивное глобальное освещение. Исследователи предложили различные техники, которые подделяют полное или частичное моделирование вторичного освещения на графической аппаратуре. В работе [11] представлен оптимизированный метод внешней преграды (*ambient occlusion*), который обеспечивает качественное моделирование затенения, но не учитывает многократных отражений света. В работе [12] описан метод отражающих теневых карт (*reflective shadow maps*), который позволяет обрабатывать только один отскок вторичного освещения. В работе [13] представлен метод мгновенной излучательности (*instant radiosity*), который сводит расчет вторичного освещения к прямому освещению от виртуальных точечных источников света. В работах [14]–[15] авторы реализовали варианты метода фотонных карт, который является универсальным решением для моделирования большинства эффектов глобального освещения.

Целью настоящей работы является разработка системы для интерактивного расчета глобального освещения на ГПУ без допущений (*unbiased*). В основе системы лежит ускоряющая структура на базе SAH BVH дерева, для которой в работе [16] были представлены эффективные алгоритмы построения и обхода. Программный конвейер визуализации может быть настроен на выполнение таких методов, как испускание лучей, трассировка лучей, стохастическая трассировка путей и двунаправленная трассировка путей. Указанные методы обладают различными возможностями и трудоемкостью, обеспечивая компромисс между скоростью и качеством визуализации в конкретной задаче. Вне зависимости от метода визуализация сохраняет корректность, основу которой составляют физически обоснованные модели материалов и источников света.

## 2. ОСНОВНЫЕ МОДЕЛИ И МЕТОДЫ

В данном разделе излагаются основные модели и методы, которые лежат в основе разработанной системы и показывают полноту моделирования глобального освещения.

## 2.1 Взаимодействие света с поверхностями

В общем случае световая энергия падает на поверхность в некоторую точку  $P$  с направлением  $\Psi$  и покидает в другой точке  $Q$  по направлению  $\Theta$ . Для описания этой зависимости используется функция объемного рассеяния (BSSRDF). Если световая энергия покидает поверхность из той же точки, то отражательные свойства могут быть описаны более простой двунаправленной функцией отражения (BRDF). Данная функция определяется в точке  $x$  как отношение яркости, отраженной по направлению  $\Theta$ , к облученности поверхности, падающей с направления  $\Psi$ :

$$f_r(x, \Psi \rightarrow \Theta) = \frac{dL(x \rightarrow \Theta)}{dE(x \leftarrow \Psi)} = \frac{dL(x \rightarrow \Theta)}{L(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi} \quad (1)$$

Наряду с двунаправленной функцией отражения для обратной стороны поверхности вводится двунаправленная функция пропускания (BTDF). Для совместного описания отражающих и пропускающих свойств часто используется двунаправленная функция рассеяния (BSDF).

## 2.2 Уравнение визуализации

Для каждой точки  $x$  поверхности и каждого направления  $\Theta$  уравнение визуализации определяет исходящую яркость  $L(x \rightarrow \Theta)$  как сумму излучаемой яркости  $L_e(x \rightarrow \Theta)$  и отражаемой яркости  $L_r(x \rightarrow \Theta)$ :

$$\begin{aligned} L(x \rightarrow \Theta) &= L_e(x \rightarrow \Theta) + L_r(x \rightarrow \Theta) \\ &= L_e(x \rightarrow \Theta) + \int_{H_x} f_r(x, \Psi \rightarrow \Theta) L(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi \end{aligned} \quad (2)$$

Такая форма уравнения называется полусферической, поскольку содержит интеграл по полусфере возможных направлений. Альтернативной формой является площадная, которая заменяет интегрирование по полусфере на интегрирование по поверхностям, видимым из точки  $x$ :

$$\begin{aligned} L(x \rightarrow \Theta) &= L_e(x \rightarrow \Theta) + L_r(x \rightarrow \Theta) \\ &= L_e(x \rightarrow \Theta) + \int_A f_r(x, \bar{x} \rightarrow \Theta) L(y \rightarrow \bar{x}) V(x, y) G(x, y) dA_y \end{aligned} \quad (3)$$

Через  $G(x, y)$  обозначен геометрический член, описывающий взаимное расположение поверхностей в точках  $x$  и  $y$ :

$$G(x, y) = \frac{\cos(N_x, \bar{x}) \cos(N_y, \bar{y})}{r_{xy}^2} \quad (4)$$

Через  $I(x, y)$  обозначена функция видимости, которая равна единице, если точки  $x$  и  $y$  взаимно видимы, и нулю в противном случае. Данная функция вычисляется с помощью операции испускания луча  $r(x, \Psi)$ , определяющей ближайшую точку соударения заданного луча с объектом сцены:

$$\begin{aligned} r(x, \Psi) &= \{y \mid y = x + \tau \Psi\} \\ \tau &= \min_{\tau > 0} \{ \tau \mid x + \tau \Psi \in A \} \end{aligned} \quad (5)$$

Очевидно, точки  $x$  и  $y$  взаимно видимы тогда и только тогда, когда существует такое направление  $\Psi$ , что  $r(x, \Psi) = y$ .

## 2.3 Стохастическая трассировка путей

Через каждый пиксель экранной плоскости алгоритм трассировки путей испускает первичный луч, для которого определяется ближайшая точка соударения  $x$  со сценой. Яркость сгенерированного луча вычисляется из уравнения визуализации методом Монте-Карло. Для повышения эффективности вычислений отраженная яркость разделяется на две составляющие:

$$\begin{aligned} L_r(x \rightarrow \Theta) &= L_{direct}(x \rightarrow \Theta) + L_{indirect}(x \rightarrow \Theta) \\ &= \int_{A_{sources}} L_e(y \rightarrow \bar{x}) f_r(x, \bar{x} \rightarrow \Theta) G(x, y) V(x, y) dA_y \\ &+ \int_{H_x} L_r(r(x, \Psi) \rightarrow -\Psi) f_r(x, \Psi \rightarrow \Theta) \cos(N_x, \Psi) d\omega_\Psi \end{aligned} \quad (6)$$

Прямое освещение  $L_{direct}(x \rightarrow \Theta)$  поступает в точку  $x$  непосредственно от источников света. Вторичное освещение  $L_{indirect}(x \rightarrow \Theta)$  падает в точку  $x$  после, по крайней мере, одного отражения от другой поверхности сцены.

### 2.3.1 Расчет прямого освещения

Для генерации теневых лучей используется следующий двухшаговый алгоритм:

- С помощью дискретного распределения  $p_L(k)$  выбирается источник света  $k$ .
- С помощью условной плотности вероятности  $p(y \mid k)$  на поверхности источника  $k$  выбирается точка  $y$ .

Составная плотность вероятности для выбора точки  $y$  на объединенной поверхности источников равна  $p_L(k)p(y \mid k)$ , что ведет к следующей оценке яркости теневого луча:

$$\langle L_{direct}(x \rightarrow \Theta) \rangle = \frac{L_e(y \rightarrow \bar{x}) f_r(x, \bar{x} \rightarrow \Theta) G(x, y) V(x, y)}{p_L(k)p(y \mid k)} \quad (7)$$

В текущей версии системы для выбора источника света используется дискретное равномерное распределение.

### 2.3.2 Расчет вторичного освещения

Для заданной плотности вероятности  $p(\Psi)$  генерируется случайное направление  $\Psi$ , при этом оценка компоненты вторичного освещения имеет вид:

$$\langle L_{indirect}(x \rightarrow \Theta) \rangle = \frac{L_r(r(x, \Psi) \rightarrow -\Psi) f_r(x, \Psi \rightarrow \Theta) \cos(N_x, \Psi)}{p(\Psi)} \quad (8)$$

В данном выражении отраженная яркость  $L_r$  остается неизвестной величиной, что приводит к рекурсивной процедуре вычисления вторичного освещения. Для остановки рекурсии устанавливается максимальное число итераций или применяется принцип русской рулетки. В настоящей работе для уменьшения дисперсии оценок используется выборка по значимости: плотность вероятности  $p(\Psi)$  строится пропорционально BSDF поверхности.

## 2.4 Двунаправленная трассировка путей

Для повышения скорости сходимости используется двунаправленная трассировка путей, которая испускает лучи одновременно из источников света и виртуальной камеры.

- Световой путь описывается последовательностью  $x_0, x_1, x_2, \dots, x_{N_p}$ , где через  $x_{i+1}$  обозначена точка, видимая из точки  $x_i$  по направлению  $\Theta_{x_i}$ .
- Видовой путь описывается последовательностью  $y_0, y_1, y_2, \dots, y_{N_e-1}$ , где через  $y_{j+1}$  обозначена точка, из которой видна точка  $y_j$  по направлению  $\Theta_{y_{j+1}}$ .

Все узлы сгенерированных путей соединяются теневыми лучами, и соответствующие вклады добавляются к яркости пикселя, что приводит к следующей оценке:

$$\langle L_r(x \rightarrow \Theta) \rangle = \sum_{i=0}^{N_p} \sum_{j=0}^{N_e-1} w_{ij} C_{ij} \quad (9)$$

Здесь через  $C_{ij}$  обозначен вклад теневого луча, который соединяет  $i$ -ый узел светового пути с  $j$ -ым узлом видового пути. Через  $w_{ij}$  обозначены нормировочные коэффициенты теневых лучей, которые необходимо учитывать для корректной сходимости метода.

## 3. АРХИТЕКТУРА СИСТЕМЫ

Для расчета изображения разработан программный конвейер трассировки лучей, все этапы которого исполняются на ГПУ и могут быть разделены на следующие две стадии.

### Light Stage

- На основе информации об источниках света для каждого пикселя изображения генерируется световой луч.
- Для каждого сгенерированного луча вычисляется ближайшая точка соударения с видимым объектом сцены.
- На основе свойств поверхностей в точках соударения генерируются вторичные лучи (или принимается решение о прекращении светового пути).
- Выполняется переход к шагу 2.

### View Stage

- На основе информации о камере для каждого пикселя изображения генерируется первичный луч.
- Для каждого сгенерированного луча вычисляется ближайшая точка соударения с видимым объектом сцены.
- Каждая точка соударения соединяется теневыми лучами со всеми префиксами светового пути.
- Каждый теневой луч тестируется на соударение с любым объектом между световым и видовым путем. Если пересечение найдено, то теневой луч удаляется.
- На основе свойств поверхностей в точках соударения вычисляется вклад каждого теневого луча в яркость соответствующего пикселя.
- На основе свойств поверхностей в точках соударения генерируются вторичные лучи (или принимается решение о прекращении видового пути).
- Выполняется переход к шагу 2.

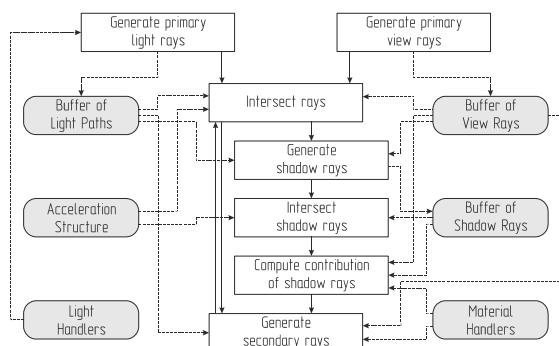


Рис. 1. Схема работы конвейера трассировки лучей.

Данный конвейер наряду с двунаправленной трассировкой путей позволяет реализовать другие методы визуализации. Полное отключение генерации световых лучей соответствует стохастической трассировке путей. Если при этом генерация вторичных лучей выполняется только для расчета идеального отражения или преломления, то конвейер реализует классический алгоритм трассировки лучей. Полное отключение вторичного освещения соответствует простому алгоритму испускания лучей. В результате возможна адаптация к широкому кругу задач – от полного моделирования глобального освещения до визуализации анимированных сцен в реальном времени с упрощенным освещением.

### 3.1 Адаптация конвейера к архитектуре ГПУ

Для обработки данных на графическом процессоре вся информация должна быть представлена в виде набора массивов (буферов). Однако во многих системах моделирования и форматах хранения используется более удобное описание компьютерной модели в виде графа сцены. В настоящей работе разработан универсальный механизм представления модели, который на пользовательском уровне функционирует как граф сцены, а на нижнем уровне обеспечивает эффективную сериализацию данных в набор массивов. Указанные массивы могут использоваться как для загрузки на

графический процессор, так и для передачи на другие вычислительные узлы при расчете на кластерной системе.

Другой особенностью настоящей работы является расширяемая подсистема материалов и источников света. Реализация указанных возможностей основана на концепции обработчиков, которые имеют уникальную реализацию для различных источников и материалов и представлены в виде указателей на функции с универсальной сигнатурой вызова (аппаратная поддержка доступна в последнем поколении графических ускорителей NVIDIA Fermi).

#### 3.1.1 Сериализация данных

На верхнем уровне сцена представлена в виде дерева, узлы которого служат только для организации объектов сцены в иерархии и не содержат конкретной информации. Для хранения специфических данных (включая сетки треугольников) используются ядра, которые могут быть включены в один или несколько узлов.

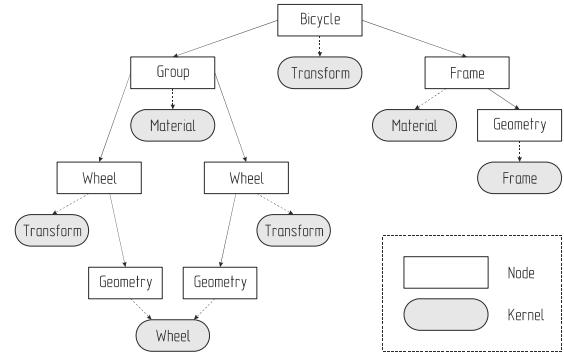


Рис. 2. Пример графа сцены для модели велосипеда.

Одной из ключевых особенностей конвейера является эффективная сериализация данных в массивы для передачи на ГПУ. Данный механизм основан на внутреннем представлении сцены в виде набора хранилищ данных различного типа, общая функциональность которых описывается следующим базовым интерфейсом:

- *Add node*. Добавляет информацию об узле графа сцены в хранилище. Память для данных ядра при этом не выделяется.
- *Delete node*. Удаляет информацию об узле из хранилища. Если для данных ядра в хранилище была выделена память, то она освобождается.
- *Allocate*. Выделяет в хранилище память для данных ядра. Запись данных ядра в хранилище не выполняется.
- *Write*. Записывает данные из ядра узла в хранилище. Память должна быть выделена заранее. При записи учитывается значения атрибутов узлов, примерами которых служат трансформации и материалы.
- *Free*. Освобождает память, которая была выделена в хранилище для данных ядра. Информация об узле из хранилища не удаляется.

Идентификация объектов в хранилище осуществляется по узлам, поэтому каждый узел не может включаться в граф сцены более одного раза. Однако включение одного ядра несколько раз возможно, поскольку оно может соответствовать нескольким узлам. При этом для каждого узла место в хранилище будет выделено отдельно, что необходимо ввиду специфики обработки данных на ГПУ (структуры с указателями обрабатываются неэффективно). Раздельное хранение информации об узлах необходимо также из-за возможных различий в свойствах узлов. Например, одна-

ковым ядрам геометрии с различными атрибутами трансформации соответствует различная хранимая информация.

### 3.1.2 Первичные лучи и сглаживание

В процессе расчета изображения каждому пикселью приписывается яркость  $L_{pixel}$ , которая выражается интегралом по экранной плоскости:

$$L_{pixel} = \int_{Screen} L(p \rightarrow eye) h(p) dp = \int_{Screen} L(x \rightarrow \bar{x}p) h(p) dp \quad (10)$$

Здесь через  $x$  обозначена точка, видимая из глаза  $eye$  через точку  $p$  экранной плоскости, а через  $h(p)$  – весовая функция (в текущей версии используется прямоугольный фильтр для усреднения падающей яркости по площади пикселя). Для оценки данного интеграла применяется метод Монте-Карло. На каждой итерации первичные лучи направляются на случайные точки в пределах соответствующих пикселей, при этом результирующее изображение получается путем усреднения нескольких проходов визуализации. Данный метод не влечет дополнительных расходов в трассировке путей и обеспечивает высокое качество сглаживания.

### 3.1.3 Поиск пересечения с объектами сцены

Наиболее трудоемким этапом визуализации является поиск ближайшего пересечения луча с объектами сцены. Для повышения производительности применяются различные ускоряющие структуры, которые хранят информацию о расположении объектов сцены и позволяют избежать полного перебора. В настоящей работе для этой цели используется иерархия ограничивающих объемов (*BVH*), для которой в работе [16] предложен эффективный алгоритм построения на основе эвристики площадей поверхностей (*SAH*), позволяющий в реальном времени визуализировать сложные динамические сцены.

Отличительной особенностью настоящей работы от большинства аналогичных проектов является возможность прямой визуализации ряда важных для виртуальной реальности объектов без предварительной триангуляции. Примерами таких объектов служат поверхности второго порядка, поверхности Безье, изоуровни скалярных полей, пространственные фрактальные множества, фрагменты сплошных сред. Важной областью применения сплошных сред является прямая визуализация объема, которая может быть легко интегрирована в полигональную модель (вода в прозрачном аквариуме или клубы дыма в помещении). Необходимым условием обработки таких объектов является наличие алгоритма пересечения с лучом. Для реализации указанных возможностей процедура поиска пересечения разбивается на несколько процедур, каждая из которых работает с определенным типом геометрических объектов.

### 3.1.4 Источники света и прямое освещение

В рамках настоящей работы разработана расширяемая подсистема источников света. На данный момент реализована поддержка изотропных точечных источников, направленных источников, непрерывных диффузных источников и бесконечно удаленных непрерывных источников, таких как верхний свет и карты окружения. Система поддерживает загрузку карт окружения из различных HDRI форматов и обеспечивает их эффективную обработку на основе выборки по значимости. Для задания характеристик всех источников используются физические величины – энергетическая яркость или сила света.

При расчете прямого освещения все источники света могут рассматриваться как один логический источник или обрабатываться независимо друг от друга. В первом режиме вне зависимости от числа источников в каждой точке соударе-

ния генерируется только один теневой луч. Во втором режиме теневые лучи генерируются для каждого источника света, а результаты расчетов суммируются.

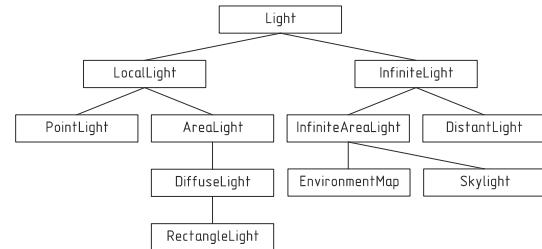


Рис. 3. Диаграмма классов для описания источников.

Первый режим обеспечивает высокую скорость работы, но при этом страдает скорость сходимости изображения. Для получения качественного результата во втором режиме требуется меньшее число итераций, но каждый проход визуализации более затратный. Тем не менее, при расчете финального изображения второй режим отработает быстрее и является предпочтительным.

### 3.1.5 Материалы и вторичное освещение

Для описания свойств поверхностей разработана расширяемая подсистема материалов, в основе которой лежит разделение механизмов рассеяния света на три типа: диффузное (*diffuse*), размытое зеркальное (*glossy*) и зеркальное (*specular*). Для описания конкретных типов рассеяния используются двунаправленные функции отражения (BRDF) и пропускания (BTDF), которые в большинстве случаев удобно обозначать общим термином *BxDF*. Текущая версия системы поддерживает модели Ламберта и Орена-Найара для диффузного отражения, модели Уорда и Блиннна для размытого зеркального отражения, а также модели идеального зеркального отражения и преломления.

Большинство реальных материалов описываются комбинацией нескольких типов отражения или пропускания. Таким образом, рассеивающие свойства поверхности можно представить взвешенной суммой *BxDF* ( $f_s$ ) различного типа, которая выражает *BSDF* ( $f_c$ ) поверхности:

$$f_s(x, \Psi \rightarrow \Theta) = \sum_{i=1}^N \alpha^i f_s^i(x, \Psi \rightarrow \Theta) \quad (11)$$

$$\sum_{i=1}^N \alpha_c^i \leq 1, \quad c \in \{R, G, B\} \quad (12)$$

Весовые коэффициенты  $\alpha^i$  в общем случае являются векторными и определяют свойства поверхности отдельно для каждого цветового канала. Функция *BSDF* является физически корректной, если справедливо неравенство (12). Данная функция используется в системе для описания материалов и передается для обработки на ГПУ.

Для каждой *BxDF* компоненты система позволяет задавать текстурную карту, которая корректирует весовой коэффициент в каждой точке  $x$  поверхности:

$$\tilde{\alpha}^i(x) = \alpha^i T^i(x), \quad i = 1..N \quad (13)$$

Здесь через  $T^i(x)$  обозначена функция выборки из текстуры. В текущей версии системы текстурные данные передаются на графический процессор в виде обычных массивов CUDA. Отказ от использования текстурной памяти ведет к некоторой потере производительности, но позволяет снять ограничения на число обрабатываемых текстур, их размер и формат пикселя. При таком подходе методы фильтрации должны быть реализованы программно. На данный момент система поддерживает интерполяцию по ближайшему элементу и билинейной интерполяции.

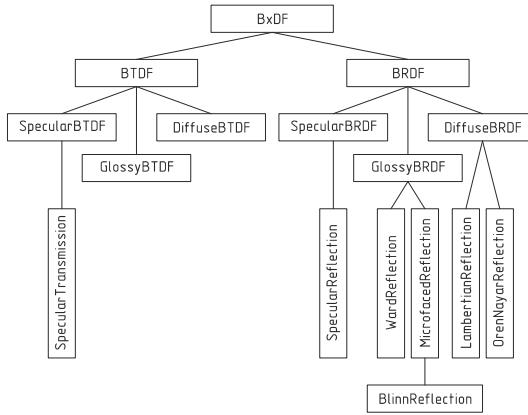


Рис. 4. Диаграмма классов для описания материалов.

Для эффективного расчета вторичного освещения используется выборка по значимости: для генерации направлений вторичных лучей плотность вероятности строится пропорционально BSDF поверхности. Для этого интеграл вторичного освещения разбивается на части, соответствующие отдельным BxDF компонентам:

$$L_{\text{indirect}}(x \rightarrow \Theta) = \sum_{i=1}^N \tilde{\alpha}^i(x) \int_{H_x} f_x^i(x, \Psi \rightarrow \Theta) L_r(r(x, \Psi) \rightarrow -\Psi) |\cos(N_x, \Psi)| d\omega_\Psi \quad (14)$$

Для оценки данного выражения используется следующий двухшаговый алгоритм:

- Конструируется дискретное распределение для  $N+1$  события, вероятности которых равны  $q_1, q_2, \dots, q_{N+1}$ . Для  $i \in [1, N]$   $i$ -ое событие состоит в выборе  $i$ -ой компоненты интеграла освещенности. Последнее событие состоит в поглощении пути.
- Если не принято решение о поглощении пути, то случайное направление  $\Psi$  генерируется с помощью плотности вероятности  $p_i(\Psi)$ , пропорциональной  $i$ -ой BxDF компоненте материала.

Оценка энергетической яркости для выбранного направления  $\Psi$  будет иметь вид:

$$\langle L_{\text{indirect}}(x \rightarrow \Theta) \rangle = \begin{cases} \frac{\int_x^i f_x^i(x, \Theta \leftrightarrow \Psi) L_r(r(x, \Psi) \rightarrow -\Psi) |\cos(N_x, \Psi)|}{q_i p_i(\Psi)} & i = 1 \dots N \\ 0 & i = N+1 \end{cases} \quad (15)$$

В настоящей работе вероятности  $q_i$  вычисляются на основе коэффициентов  $\tilde{\alpha}^i$  и текущего веса пути  $w(x)$ , который определяет вклад луча в яркость обрабатываемого пикселя:

$$q_i = \begin{cases} \frac{\tilde{\alpha}_R^i w_R(x) + \tilde{\alpha}_G^i w_G(x) + \tilde{\alpha}_B^i w_B(x)}{w_R(x) + w_G(x) + w_B(x)} & i = 1 \dots N \\ 1 - \sum_{i=1}^N q_i & i = N+1 \end{cases} \quad (16)$$

Данный подход обеспечивает корректный баланс BxDF составляющих, учитывает потенциальный вклад текущего луча и влияние текстурных карт.

## 4. АНАЛИЗ РЕЗУЛЬТАТОВ

### 4.1 Проверка корректности

Для проверки корректности использовались сцены для тестирования алгоритмов глобального освещения [17], которые выявляют различные аспекты переноса света. Все изображения были сгенерированы методом стохастической трассировки путей (рис. 5). Для математической оценки

результатов в каждом случае вычислялась среднеквадратичная ошибка (RMSE, рис. 6).

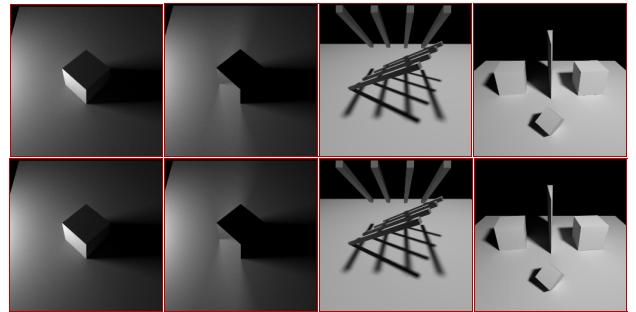


Рис. 5. Наши результаты (сверху) и эталонные (снизу): Secondary, Caustic, Shadow и Geometry.

Небольшие отклонения связаны с особенностями реализации датчика случайных чисел на графическом процессоре.

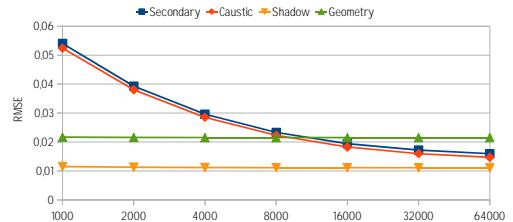


Рис. 6. Зависимость ошибки (RMSE) от числа итераций.

Тем не менее, полученные результаты позволяют говорить о корректности визуализации.

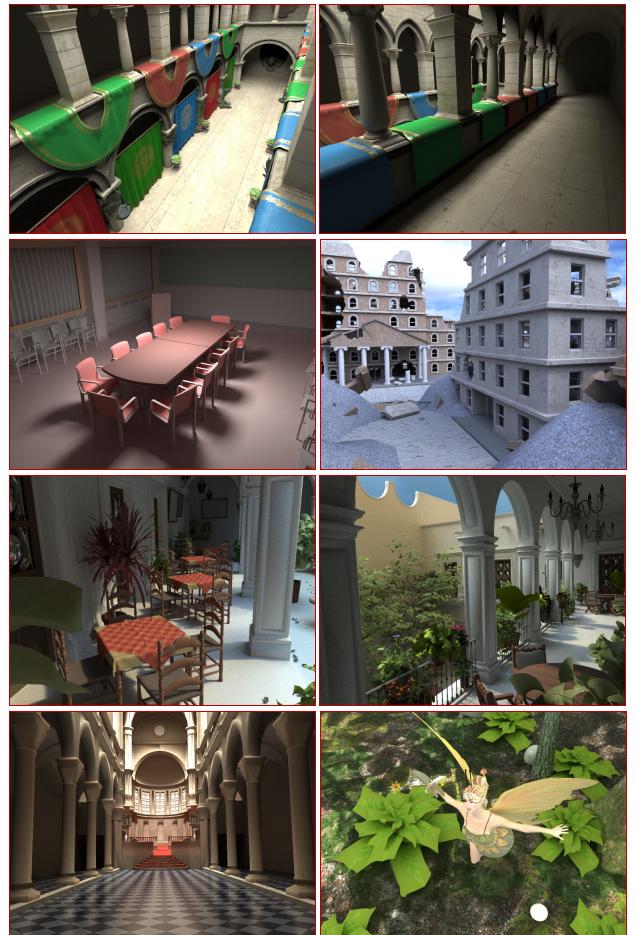


Рис. 7. Тестовые сцены и точки обзора: Crytek Sponza (2), Conference, Ruins, San Miguel (2), Sibenik, Fairy Forest.

## 4.2 Оценка производительности

Для оценки производительности использовались популярные сцены для сравнения алгоритмов глобального освещения (рис. 7). Модели содержат различные типы материалов, а также диффузные и зеркальные текстурные карты высокого разрешения. Изображения визуализировались в разрешении  $800 \times 600$  методом стохастической трассировки путей при максимальной глубине пути 10. Для расчетов использовался графический процессор начального уровня NVIDIA GeForce 560 1 Гб.

Test Scene	Triangles	MRays / sec	FPS
Crytek Sponza ( <i>outside</i> )	262 267	28.1	5.85
Crytek Sponza ( <i>inside</i> )	262 267	38.3	7.9
Conference Room	331 179	37.1	7.8
Ruins	336 967	41.4	8.5
San Miguel ( <i>ground floor</i> )	7 880 512	23.2	4.8
San Miguel ( <i>upper floor</i> )	7 880 512	20.2	4.2
Sibenik Cathedral	75 284	40.5	8.4
Fairy Forest	174 117	111.6	23.4

Таким образом, приемлемое для интерактивных приложений качество изображений (несколько десятков кадров) может быть получено за несколько секунд. Производительность может быть значительно повышена за счет сокращения максимальной длины путей до 2-3 отскоков с сохранением всех эффектов глобального освещения.

## 4.3 Дополнительная функциональность

Система поддерживает визуализацию методом двунаправленной трассировки путей, которая эффективна при расчете вторичного освещения. На рис. 8 показано сравнение обычной и двунаправленной трассировки в ситуациях, когда площадной источник направлен вниз и вверх. Во втором случае объекты получают лишь вторичное освещение.

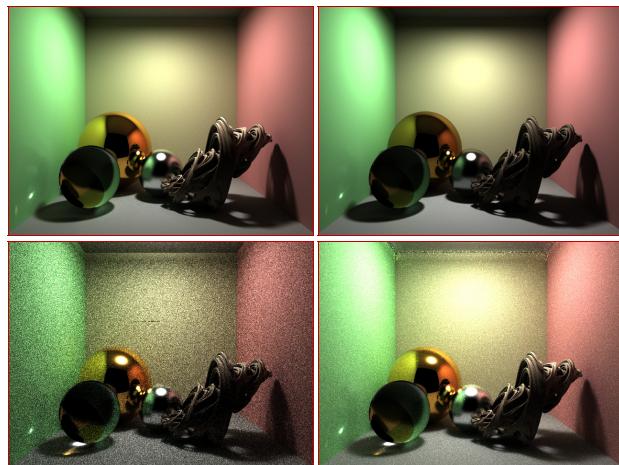


Рис. 8. Обычный (слева) и двунаправленный метод (справа)

Кроме того, данный рисунок демонстрирует возможность обработки таких неполигональных объектов, как сферы и фрактальные множества Джюлиа.

## 5. ЗАКЛЮЧЕНИЕ

В рамках настоящей работы разработана система интерактивной визуализации для сложных сцен гибридной природы с поддержкой глобального освещения и динамической геометрии. Под гибридностью понимается возможность сочетания традиционных полигональных моделей с объектами других типов, таких как полупрозрачные среды, поверхности Безье, изоуровни скалярных полей и поверхно-

сти второго порядка. В основе системы – гибкий графический конвейер, который допускает конфигурацию всех этапов формирования изображения. Настройка конвейера позволяет получать эффективные решения для различных задач – от расчета глобального освещения с расширенными моделями материалов и источников света до интерактивных систем с базовыми эффектами глобального освещения и динамической геометрией. Реализована межплатформенная версия системы на базе NVIDIA CUDA, которая обеспечивает высокую производительность построения ускоряющих структур и визуализации гибридных сцен.

## 6. ЛИТЕРАТУРА

- [1] M. J. Keates, R. J. Hubbold. *Interactive ray tracing on a virtual shared-memory parallel computer* // Computer Graphics Forum 14, 4 (October 1995), 189-202.
- [2] M. Muuss. *Towards real-time ray-tracing of combinatorial solid geometric models* // In Proc. of BRL-CAD Symposium '95, 5-9 June 1995.
- [3] S. Parker, W. Martin, P. J. Sloan, P. Shirley, B. Smits, C. Hansen. *Interactive ray tracing* // In Proc. of Interactive 3D Graphics (1999), 119-126.
- [4] I. Wald, P. Slusallek, C. Benthin, M. Wagner. *Interactive Rendering with Coherent Ray tracing* // Computer Graphics Forum 20, 3 (2001), 153-164.
- [5] A. Reshetov, A. Soupikov, J. Hurley. *Multi-Level Ray Tracing Algorithm* // ACM Trans. Graph. 24, 3, 1176-85.
- [6] C. P. Gribble, T. Ize, A. Kensler, I. Wald, S. G. Parker. *A coherent grid traversal approach to visualizing particle-based simulation data* // IEEE Transactions on Visualization and Computer Graphics 13, 4, 758-768.
- [7] S. Boulos, D. Edwards, J. D. Lacewell, J. Kniss, J. Kautz, P. Shirley, I. Wald. *Packet based Whitted and Distribution Ray Tracing* // In Proc. of Graphics Interface 2007 (GI '07). ACM, New York, NY, USA, 177-184.
- [8] J. M. Singh, P. J. Narayanan. *Real-Time Ray Tracing of Implicit Surfaces on the GPU* // IEEE Transactions on Visualization and Computer Graphics 16, 2, 261-272.
- [9] J. Mensmann, T. Ropinski, K. Hinrichs. *An Advanced Volume Raycasting Technique using GPU Stream Processing* // International Conference on Computer Graphics Theory and Applications 2010, 190-198.
- [10] I. Wald, W. R. Mark, J. Günther, S. Boulos, T. Ize, W. Hunt, S. G. Parker, P. Shirley. *State of the Art in Ray Tracing Animated Scenes* // Computer Graphics Forum 28, 6 (2009), 1691-1722.
- [11] P. Shanmugam, O. Arik. *Hardware accelerated ambient occlusion techniques on GPUs* // In Proc. of the 2007 symposium on Interactive 3D graphics and games, 73-80.
- [12] C. Dachsbaecher, M. Stamminger. *Reflective shadow maps* // In Proc. of the 2005 symposium on Interactive 3D graphics and games (I3D '05), 203-231.
- [13] I. Radax. *Instant Radiosity for Real-Time Global Illumination*. Institute of Computer Graphics and Algorithms. Technical Report (May 2008).
- [14] R. Wang, R. Wang, K. Zhou, M. Pan, H. Bao. *An efficient GPU-based approach for interactive global illumination* // ACM Trans. Graph. 28, 3, Article 91.
- [15] M. McGuire, D. Luebke. *Hardware-Accelerated Global Illumination by Image Space Photon Mapping* // In Proc. of High Performance Graphics 2009 (HPG '09), 77-89.
- [16] D. Sopin, D. Bogolepov, D. Ulyanov. *Real-time SAH BVH construction for ray tracing dynamic scenes* // In. Proc. of the GraphiCon'2011, 74-77.
- [17] B. Smits, H. W. Jensen. *Global Illumination Test Scenes* // Tech. Rep. UUCS-00-013, Computer Science Department, University of Utah, June 2000.

# Оптимизация определения видимых фрагментов текстуры для алгоритма виртуализации памяти графических ускорителей

Гордеев Антон, Денис Гладкий, Игорь Белаго, Станислав Кузиковский  
 Институт Автоматики и Электрометрии СО РАН  
 Новосибирск, Россия  
[togorde@gmail.com](mailto:togorde@gmail.com), [generalgda@gmail.com](mailto:generalgda@gmail.com), [{bel,stas}@sl.iae.nsk.su](mailto:{bel,stas}@sl.iae.nsk.su)

## Аннотация

Данная статья посвящена алгоритму виртуализации памяти графических ускорителей. Он используется при реализации фотореалистичного текстурирования. Для этого виртуальный мир текстурируется одной большой текстурой, разрешение которой может превосходить 128000 на 128000 пикселей.

Для определения видимых фрагментов текстуры, которые необходимо загружать в видеопамять для формирования кадра сцены, используется идея построения текстурной карты – таблицы индексов вышеупомянутых фрагментов.

В данной статье предложен метод оптимизации конструирования текстурной карты. Для этого используются возможности архитектуры Shader Model 4.0, а именно геометрический шейдер (geometry shader).

**Ключевые слова:** Виртуализация памяти, Графический ускоритель, Фотореалистичное текстурирование, Megatexture, Virtual texture mapping.

## 1. ВВЕДЕНИЕ

Современные компьютерные игры и другие интерактивные приложения виртуальной реальности, для достижения фотореалистичности изображения, используют большие объемы высоко детализированных текстурных данных. Поэтому на видеокарты накладываются постоянно увеличивающиеся серьёзные требования на объём видеопамяти. Также эти требования постоянно увеличиваются. Однако объём доступной графической памяти не растет в таком же темпе. Более того, память графических ускорителей требуется для хранения вершинных и индексных буферов, осуществления сложных эффектов пост-обработки, расчёта освещенности сцены и многих других задач.

## 2. ВИРТУАЛИЗАЦИЯ ПАМЯТИ ГРАФИЧЕСКИХ УСКОРИТЕЛЕЙ

Алгоритм виртуализации памяти графических ускорителей предназначен для уменьшения объёма памяти видеокарты, необходимой для размещения текстур, до предела, который в идеальном случае определяется разрешением экрана (для закраски  $N$  пикселей уникальными цветами необходимо, как минимум,  $O(N)$  байтов памяти – по блоку фиксированного размера на каждый пиксель).

Для текстурирования мира используется одна большая текстура, разрешение которой может достигать 128000 на 128000 пикселей [1]. Её несжатое представление может занимать объём, превышающий несколько гигабайт. Такая текстура не может целиком поместиться в видеопамять современных графических ускорителей.

Основная идея алгоритма заключается в том, что для каждого конкретного кадра не требуется загружать всю текстуру целиком, необходимы только те её фрагменты, которые будут видны в данном кадре на экране.

## 3. ПОСТРОЕНИЕ ТЕКСТУРНОЙ КАРТЫ

Для определения видимых фрагментов текстуры используется алгоритм построения текстурной карты в отдельном предварительном цикле работы видеокарты. При этом используется идея рендеринга геометрии в текстурных координатах [4]. Эта идея заключается в том, что в качестве геометрических координат вершины используются её текстурные координаты, а растеризация ведётся в специальный буфер. После завершения работы конвейера видеокарты, каждый фрагмент геометрии растеризуется в области, использующейся для его (фрагмента) текстурирования.

Поскольку, у каждой вершины вместо реальных геометрических координат используются её текстурные координаты, то в процессе работы графического конвейера невозможно использовать ряд его автоматических тестов отбраковки растеризуемых фрагментов, например:

- frustum culling
- back face culling
- z test

Поэтому необходимо эмулировать их с помощью программируемых блоков конвейера видеокарты.

В результате получается текстурная карта, закрашенные пиксели которой соответствуют участкам исходной большой текстуры, которые необходимо загрузить в видеопамять.

## 4. ОПТИМИЗАЦИЯ ПОСТРОЕНИЯ ТЕКСТУРНОЙ КАРТЫ

В ситуации, когда объекты перекрывают друг друга относительно камеры (occlusion culling) выделяются два возможных случая. В первом, фрагмент одного объекта перекрывает фрагментом другого. Во втором, фрагмент попадает на заднюю грань объекта, и перекрывается его передней гранью. Отбраковка фрагмента, при его попадании на заднюю грань, называется back face culling.

Для выполнения back face culling нами был разработан новый подход, базирующийся на геометрическом шейдере – этапе графического конвейера, доступного для видеокарт, реализующих архитектуру Shader Model 4.0 [3]. Он позволяет написать программу, обрабатывающую целый геометрический примитив (отрезок, треугольник и пр.), а не отдельные пиксели или вершины, как позволяли графические ускорители архитектуры Shader Model 3.0 [2]. Подход также может быть полезен при использовании различных алгоритмов, нестандартно использующих back face culling совместно с методом виртуального текстурирования.

Идея оптимизации заключается в том, что в геометрическом шейдере можно реализовать проверку, попадает ли треугольник на переднюю или заднюю грань объекта. По результату проверки, треугольник либо сбрасывается, либо передаётся на следующий этап графического конвейера. Сама проверка, как таковая,

может быть реализована в виде сравнения знака скалярного произведения нормали, обрабатываемого треугольника, с вектором от любой его точки к позиции камеры (вектор «на камеру»). Нормаль к треугольнику может строиться как статически, передаваясь на конвейер видеокарты в виде атрибута вершины, так и вычисляться динамически. В нашей реализации (листинг 1) мы использовали второй подход.

Стоит заметить, что, как вектор «на камеру», так и нормаль треугольника, не обязательно должны быть нормированным, поскольку знак скалярного произведения не зависит от длин векторов-операндов.

```
[maxvertexcount(3)]
void geometryShader(triangle GSPS_INPUT input[3],
    inout TriangleStream<GSPS_INPUT> OutputStream)
{
    float3 edge1 = input[1].ActualPos - input[0].ActualPos;
    float3 edge2 = input[2].ActualPos - input[0].ActualPos;
    float3 normal = cross(edge1, edge2);
    float3 toCamera = eye - input[1].ActualPos;
    if (dot(toCamera, normal) >= 0.0f)
    {
        for(int i = 0; i < 3; i++)
        {
            OutputStream.Append( input[i] );
        }
        OutputStream.RestartStrip();
    }
}
```

Листинг 1: код геометрического шейдера на языке HLSL.

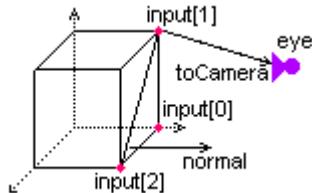


Рисунок 1: пример грани, прошедшей тест в геометрическом шейдере.

Степень ускорения построения текстурной карты, при использовании описанной оптимизации, зависит, как от конфигурации сцены, так и от положения камеры на ней. Можно выделить два краевых случая. В обоих из них сцена представляет собой график функции, заданной на подмножестве  $R^2$ . Примером может служить часто встречающиеся в приложениях такие односторонние полигональные сетки, как водная поверхность или ландшафт. В вырожденном случае первого рода (рисунок 2) камера находится над поверхностью и направлена «на неё» (скалярное произведение векторов направления «на камеру» и нормали любого треугольника поверхности больше нуля).

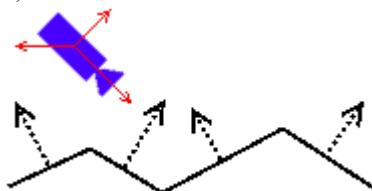


Рисунок 2: вырожденный случай 1-го рода.

В этом случае оптимизация не только не даст улучшения производительности, но и может снизить её, поскольку ни

один треугольник сцены не будет отбракован, несмотря на затраченные дополнительные вычисления.

В вырожденном случае второго рода (рисунок 3) камера находится «под поверхностью» (скалярное произведение вектора направления «на камеру» и нормали любого треугольника поверхности меньше нуля), в результате чего будут отбраковываться все треугольники сцены, что полностью разгружает этапы графического конвейера, следующие за геометрическим шейдером (растеризация, пиксельный шейдер и пр.).

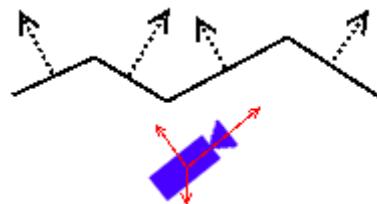


Рисунок 3: вырожденный случай 2-го рода.

Однако типичная сцена современных интерактивных приложений виртуальной реальности редко представляет собой описанные вырожденные случаи. Чаще всего она формируется из выпуклых многогранников или объектов, составленных из них. В подобной сцене, в среднем, стоит ожидать ускорение построения текстурной карты в два раза, благодаря отбраковке, примерно, половины треугольников.

## 5. РЕЗУЛЬТАТЫ ИЗМЕРЕНИЙ

Для тестирования производительности были произведены измерения количества кадров, рисуемых за секунду, при построении текстурной карты. Были использованы фрагменты виртуального мира из игры "Дальнобойщики 3", состоящие из различного числа треугольников. Измерения производились при фиксированном пролете видеокамеры. Использовалась видеокарта AMD Radeon 6770m.

Результаты измерений показали, что вышеописанная оптимизация, как и ожидалось, позволяет получить выигрыш в производительности близкий к 50%.

В таблице 1 приведены результаты измерения количества кадров, рисуемых за секунду при построении текстурной карты с вышеупомянутой оптимизацией и без неё.

Конфигурация сцены	С гш*	Без гш
Модель 1 (4500 треуг-ов)	935	934
Модель 2 (11000 треуг-ов)	775	483
Модель 3 (20000 треуг-ов)	336	189
Модель 4 (70000 треуг-ов)	225	139

Таблица 1, \* - геометрический шейдер.

Отсутствие прироста производительности при использовании оптимизации на сцене в первой записи таблицы (конфигурация «Модель 1») объясняется строением сцены. Она представляет собой ландшафт с расположенной на нём полигональной моделью жилого здания. Поскольку количество треугольников в последнем мало по сравнению с их общим числом на сцене, то данная конфигурация является близкой к вырожденному случаю первого рода.

В остальных строках таблицы представлены результаты тестирования на сценах, содержащих ту же ландшафтную сетку, но с уже более сложными (по количеству

треугольников) архитектурными объектами, что является более типичным случаем для современных приложений.

## 6. ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы был реализован алгоритм определения видимых фрагментов текстуры для виртуализации памяти графических ускорителей. Для него была предложена и реализована оптимизация, отбрасывающая задние грани в геометрическом шейдере. Были проведены измерения производительности, показавшие положительные результаты.

## 7. ССЫЛКИ

- [1] Matthaus G. Chajdas, Christian Eisenacher, Marc Stamminger, Sylvain Lefebvre. Virtual Texture Mapping 101. GPU Pro: advanced rendering techniques / edited by Wolfgang Engel. A K Peters, Ltd. 2010.
- [2] Randima Fernando. Shader Model 3.0 Unleashed. SIGGRAPH Proceedings, 2004.
- [3] Suryakant Patidar, Shiben Bhattacharjee, Jag Mohan Singh, P. J. Narayanan. Exploiting the Shader Model 4.0 Architecture. IIIT/TR/2007/145, March 2007.
- [4] Sylvain Lefebvre, Jerome Darbon, Fabrice Neyret. Unified Texture Management for Arbitrary Meshes. Technical Report RP5210, INRIA, 2004.

## Об авторах

Гордеев Антон – студент кафедры АФТИ ФФ НГУ. Его адрес: [togorde@gmail.com](mailto:togorde@gmail.com).  
Денис Гладкий – ассистент кафедр ИИС ФИТ и АФТИ ФФ НГУ, инженер-программист компании Playtox. Его адрес: [generalgda@gmail.com](mailto:generalgda@gmail.com).  
Игорь Белаго – научный сотрудник лаборатории №13 ИАиЭ СО РАН. Его адрес: [bel@sl.iae.nsk.su](mailto:bel@sl.iae.nsk.su).  
Станислав Кузиковский - научный сотрудник лаборатории №13 ИАиЭ СО РАН. Его адрес: [stas@sl.iae.nsk.su](mailto:stas@sl.iae.nsk.su).

# Оптимальное совмещение конечного множества точек с непрерывной поверхностью в трёхмерном пространстве

Александр Самойлов, Евгений Попов

Кафедра инженерной геометрии, компьютерной графики и автоматизированного проектирования

Нижегородский Государственный Архитектурно-Строительный Университет, Нижний Новгород, Россия  
{greatlimit@rambler.ru, popov@sandy.ru}

## Аннотация

Работа посвящена проблеме совмещения в трёхмерном пространстве двух множеств друг с другом, первое из которых представляет собой конечный набор точек (заданный в общем случае на нерегулярной сетке), а второе является фрагментом непрерывной поверхности.

Новый алгоритм решения данной проблемы основан на геометрических построениях и имеет преимущества по скорости и простоте программной реализации по сравнению с решением на основе методов оптимизации.

Алгоритм может найти эффективное применение в области бесконтактного определения геометрии крупногабаритных изделий, а так же для совмещения фрагментов цветных 2D изображений.

**Ключевые слова:** совмещение геометрических фигур, совмещение изображений, преобразование движения.

## 1. ВВЕДЕНИЕ

**Постановка задачи.** Дано два множества – непрерывная поверхность  $\Sigma$  в трёхмерном пространстве, заданная на компактной области  $D \subset \mathbb{R}^2$ , и набор точек  $P: P_i(x_i, y_i, z_i)$ ,  $i = \overline{1, N}$ . Требуется из всевозможных трёхмерных преобразований движения  $\Omega$  найти такое преобразование  $\Omega_{opt}$ , чтобы множество  $\Omega_{opt}(P)$  являлось ближайшим к поверхности  $\Sigma$  согласно заданной функции расстояния  $\rho$ , то есть чтобы было выполнено условие минимума:

$$\sum_{i=1}^N \rho(\Sigma, \Omega_{opt}(P_i)) = \min_{\Omega} \sum_{i=1}^N \rho(\Sigma, \Omega(P_i)) \quad (1),$$

где  $\rho(\Sigma, X)$  – функция расстояния от точки  $X$  до поверхности  $\Sigma$ . Преобразование движения в трехмерном пространстве задается 6 параметрами: координатами вектора сдвига  $\Delta_1, \Delta_2, \Delta_3$  и тремя углами поворота  $\varphi_x, \varphi_y, \varphi_z$ . Численное решение задачи методами оптимизации представляется затруднительным по следующим причинам:

- В общем случае между множествами трудно установить даже приблизительное соответствие, что не позволяет оценить интервалы изменения параметров  $\Delta_1, \Delta_2, \Delta_3, \varphi_x, \varphi_y, \varphi_z$ , вынуждая брать их максимальными, что в свою очередь существенно замедляет вычислительный процесс
- Поверхность может не быть односвязной, что увеличивает число ограничений в задаче оптимизации, тем самым усложняя её
- Поверхность не всегда задается аналитически (может вычисляться через интерполяционные формулы), поэтому её производная в общем случае не известна, что препятствует использованию эффективных численных алгоритмов, требующих знания производной

- Время вычисления функции расстояния зависит от  $N$  – числа точек множества  $P$ , что существенно замедляет вычислительный процесс в случае густого точечного множества

Сформулированная задача является актуальной, как минимум, по двум направлениям.

Во-первых, совмещение трёхмерных фигур – проблема, востребованная в контроле геометрии изделий на предприятиях, который предполагает сопоставление модели изготовленного изделия с моделью эталонного образца (первая, как правило, строится с помощью бесконтактных измерений, а вторая – на основе таблицы проектных величин эталонного образца). Для вычисления отклонений в конкретных точках изделия необходимо привязать обе модели к одной системе координат, для чего используется ряд опорных точек (базис), относительно которых проводятся бесконтактные измерения. Но в ряде случаев (деформируемые изделия; изделия, подвергающиеся транспортировке в разобранном виде и затем – сборке) задание базиса не имеет смысла. В подобных ситуациях нужен алгоритм совмещения двух моделей изделия в общей системе координат, не обладая информацией об изначальном базисе измерений.

Во-вторых, задача гипотетически может представлять более общую формулировку задачи о совмещении изображений, которая на данный момент является одной из фундаментальных задач компьютерного зрения. Если двумерному цветному изображению можно поставить в соответствие функцию двух переменных, значение которой выражает оттенок цвета в конкретной точке  $(x, y)$ , то по двум изображениям можно восстановить две поверхности и рассматривать меньшее по площади изображение как множество  $P$ . Задача о совмещении изображений является плохо обусловленной задачей [1]. Наиболее известные на сегодня методы её решения – корреляционные методы [6, 7] и методы, основанные на поиске инвариантов или особых точек [1, 2, 3].

## 2. РЕШЕНИЕ ДВУМЕРНОЙ ЗАДАЧИ

Основные положения разработанного подхода могут быть пояснены на примере двумерной задачи. В этом случае требуется найти двумерное преобразование движения, переводящее двумерный массив точек  $P: P_i(x_i, y_i)$  в положение, соответствующее минимальному расстоянию от заданной непрерывной кривой  $l$ . Могут быть сформулированы следующие три допущения:

1. Множество  $P$  и фрагмент кривой  $l$  являются объектами с приблизительно схожей формой
2. Точки множества  $P$  идут друг за другом достаточно «густо», т. е. множество  $P$  формой похоже на непрерывную кривую
3. Множество  $P$  не имеет выбросов, либо выбросы были предварительно устранины.

В двумерном случае на кривой возможно выделить начальную и конечную точки и пронумеровать точки от первой до последней. Тогда приближённое решение может быть найдено путем совмещение кривой  $l$  с треугольником максимального периметра, построенного на точках множества  $P$ , что представляет собой существенно более простую задачу, т. к. число точек равно трем. Решения данной подзадачи методами оптимизации имеет те же недостатки, что и перечисленные выше для исходной задачи, за исключением того, что неизвестных параметров в данном случае три, и число точек множества  $P$  равно трём. Использовать его нет смысла еще и потому, что при обобщении результатов решения двумерной задачи на трехмерный случай задача поиска минимума вновь требует определения 6 параметров. В связи с этим, найдём соответствие между треугольником максимального периметра и кривой  $l$  путем геометрических построений (назовем этот подход «методом окружностей»). Для удобства обозначим  $\rho(P_a, P_b) = r_{ab} \forall a, b = \overline{1, N}$  и далее везде будем использовать данное обозначение. Пусть кривая  $l$  параметризована  $l: \{x = x(t), y = y(t)\}, t \in [0, 1]$ . Организуем перебор по параметру  $t$  с шагом  $\Delta t$  (см. рис. 1), отмечая на фрагменте кривой  $l$  точку  $P_a^*$  (**первая** точка треугольника) и проводя окружность с центром в этой точке радиуса, равного  $r_{ab}$ .

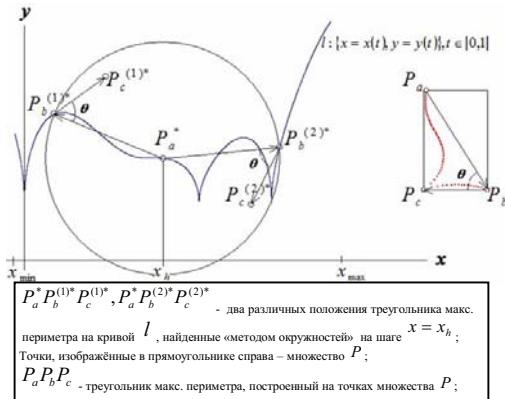


Рис. 1. Поиск решений 1-го приближения двумерной задачи «методом окружностей»

Точку пересечения  $P_b^*$  (**вторая** точка треугольника) окружности с кривой  $l$  найдем методом биссекции, разбив окружность на дуги такого малого размера  $\Delta l_{okp}^{(1)}$ , чтобы на каждой из дуг было не более одной точки пересечения. В общем случае, точек  $P_b^*$  может быть несколько. Для каждой из найденных точек пересечения  $P_b^*$  найдём точку  $P_c^*$  (**третья** точка треугольника), исходя из условия равенства треугольников  $P_a^* P_b^* P_c^*$  и  $P_a P_b P_c$ . Если третья точка отстоит от кривой  $l$  на расстояние, не превышающее заданную величину  $\delta_1$ , ищем параметры двумерного преобразования движения и пополняем ими множество решений 1-го приближения  $G^{(1)}$ . Преобразование движения в двумерном пространстве однозначно определяется тремя параметрами, при условии, что задан центр поворота. Для повышения точности будем брать в качестве центра поворота точку пересечения медиан треугольника максимального периметра. Найдя множество  $G^{(1)}$ , организуем процедуру уточнения, которая будет основана на следующих положениях.

1) На каждом шаге итерационного процесса согласно заданному правилу помимо точек  $P_a, P_b, P_c$  в рассмотрение вводятся добавочные точки из множества  $P$ . Пусть  $\Omega_k$  - множество точек, введенных в рассмотрение на  $k$ -м шаге приближения. Тогда  $\Omega_1: \{P_a, P_b, P_c\}$ . По определенному правилу каждому из двух отрезков  $P_a P_b$ ,  $P_b P_c$  ставится в соответствие по одной новой точке из множества  $P$ , и на 2-м шаге приближения рассматривается уже 4 отрезка:  $P_a P_\mu$ ,  $P_\mu P_b$ ,  $P_b P_v$ ,  $P_v P_c$ . Далее эта процедура применяется ко всем отрезкам рекурсивно. То есть на  $k$ -м шаге приближения имеется  $N_k = (2^k + 1)$  точек и  $2^k$  отрезков, соединяющих эти точки. В качестве правила может быть выбрано условие: каждым двум точкам  $P_i, P_j$  ставится в соответствие треугольник наибольшего периметра среди всех треугольников  $P_i P_s P_j$ , где  $i < s < j$ .

2) Для каждого из множества решений  $k$ -го приближения  $G^{(k)}$  выполняется варьирование (см. рис. 2) каждой из трёх точек-образов  $P_a^*, P_b^*, P_c^*$  в своих  $\delta_k$ -окрестностях. Вариации  $P_{am}^*$  первой точки определяются сдвигом первой точки  $P_a^*$  с шагами  $\Delta\varphi$  и  $(\Delta r)_k = \delta_k / N_{div}$  ( $\Delta\varphi, N_{div}$  - заданы) соответственно по углу  $\varphi$  и радиус-вектору  $r$  в полярных координатах ( $0 \leq \varphi < 2\pi, 0 \leq r \leq \delta_k$ ). Вариации  $P_{bm}^*$  второй точки определяются как точки, взятые на дуге, полученной пересечением  $\delta_k$ -окрестности точки  $P_b^*$  и окружности радиуса  $r_{ab}$  с центром в  $P_a^*$ . Итерирование по дуге происходит по параметру  $\kappa = \overline{0, L-1}$ , где  $L = \Delta l / \Delta l_{okp}^{(2)}$  - число точек, взятых на дуге через заданное расстояние  $\Delta l_{okp}^{(2)}$ ,  $\Delta l$  - длина дуги. Вариации  $P_{cm}^*$  третьей точки определяются из условия равенства треугольников  $P_a^* P_b^* P_c^*$  и  $P_{am}^* P_{bm}^* P_c^*$ .

3) Для каждой из вариаций  $P_{am}^*, P_{bm}^*, P_{cm}^*$  вычисляются параметры двумерного преобразования движения, а затем - положения, в которые отобразятся этим преобразованием остальные точки из множества  $\Omega_k$ . Если образы всех точек из  $\Omega_k$  удовлетворяют условию

$$\forall s = \overline{1, N_k} : \rho(l, P_{sm}^*) < \delta_k \quad (2),$$

тогда преобразование запоминается в список элементов множества  $G^{(k+1)}$ . Это шаг итерационного процесса.

4) Выбор критериев останова. Процесс должен остановиться, если выполнено одно из условий:

- Достигнута заданная точность  $\delta_\infty$ , т. е. условие (2) выполнено при  $\delta_k = \delta_\infty$
- Дальнейшие итерации не приводят к уменьшению числа элементов множества  $G^{(k)}$
- Выполнено более  $N_{fin}$  итераций ( $N_{fin}$  - задано)

Формулы для вычисления координат трёх варьируемых точек приведены в [5]. Таким образом, имеем всего 5 параметров, влияющих на точность и скорость алгоритма:  $\Delta t, \Delta\varphi, N_{div}, \Delta l_{okp}^{(1)}, \Delta l_{okp}^{(2)}$ . Описанный выше алгоритм был реализован в программном модуле «2DAnalyzer» [5] и протестирован на кривых различного вида.

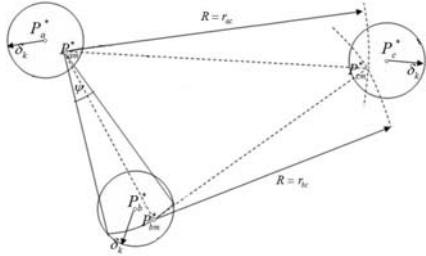


Рис 2: Процедура уточнения для двумерной задачи.

**Тест №1 (точного соответствия между множествами нет).** Кривая  $l$  задана уравнением  $f(x) = x \cos(0.04x)$  на отрезке  $[-50; 30]$ . Значения параметров:  $\delta_1 = 3$ ,  $\Delta t = 0.00625$ ,  $\Delta\phi = 0.2\pi$ ,  $N_{div} = 10$ ,  $\Delta l_{osp}^{(1,2)} = 0.001$ ,  $n_{fin} = 3$ . В качестве множества  $P$  были взяты точки кривой  $g(x) = 0.0005x^3$  на отрезке  $[-20; 40]$  через расстояние 0.9, сдвинутые затем на вектор  $\{100; 50\}$  и повёрнутые на угол  $-\frac{\pi}{3}$  относительно начала координат.

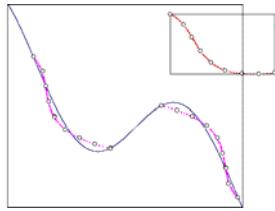


Рис 3: Совмещение кривых  $f(x) = x \cos(0.04x)$  и  $g(x) = 0.0005x^3$  с помощью модуля «2DAnalyzer»

На рис. 3 изображена последняя (третья) итерация алгоритма. В большем по размеру прямоугольнике изображена кривая  $l$  и точечное множество, совмещённое с ней. В меньшем по размеру прямоугольнике изображено множество  $P$ . Малыми кругами отмечены точки множества  $\Omega_3$ . Время работы программы – около 5 с.

**Тест №2 (точное соответствие между множествами)** Исходные данные для теста такие же, как для теста №1, кроме того, что: а) Кривая  $l$  задана уравнением  $f(x) = x \ln(2.5 + \sin(0.1x))$  на отрезке  $[-30; 0]$ ; б)  $\delta_1 = 0.2$ , в)  $g(x) = f(x)$ . На рис. 4 изображена последняя (третья) итерация алгоритма. Время работы программы – около 5 с.

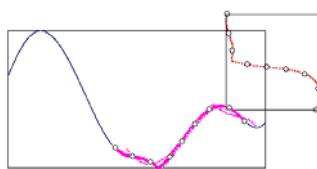


Рис 4: Совмещение двух фрагментов кривой  $f(x) = x \ln(2.5 + \sin(0.1x))$  с помощью модуля «2DAnalyzer»

Эти и другие (не приведённые здесь) тесты показали, что алгоритм для двумерного случая работает эффективно.

### 3. РЕШЕНИЕ ТРЕХМЕРНОЙ ЗАДАЧИ

Попытаемся обобщить алгоритм на трехмерный случай. Обобщению будут подвергнуты принцип треугольника максимального периметра и процедура уточнения. Три допущения, сделанные для двумерной задачи, остаются в силе. Пусть  $P_a P_b P_c P_d$  – тетраэдр с максимальной суммой

длин сторон,  $P_a^* P_b^* P_c^* P_d^*$  – образ тетраэдра  $P_a P_b P_c P_d$ , соответствующий одному из искомых положений последнего на поверхности  $\Sigma$ . Пусть поверхность  $\Sigma$  задана параметрически в ограниченной двумерной компактной области  $D$ , то есть  $\Sigma : \{x = x(u, v), y = y(u, v), z = z(u, v)\}, (u, v) \in [0, 1] \times [0, 1], (x(u, v), y(u, v)) \in D \forall (u, v) \in [0, 1] \times [0, 1]$ . По аналогии с двумерной задачей, организуем перебор по каждому из параметров  $u, v$  с заданными шагами  $\Delta u, \Delta v$  соответственно. Из каждой итерируемой точки  $(x(u, v), y(u, v), z(u, v))$  (назовем ее **первой точкой**  $P_a^*$  тетраэдра) на поверхности  $\Sigma$  очертим сферу радиуса  $r_{ab}$ .  $r_{ab}$ -сфера имеет в общем случае несколько линий пересечения с поверхностью  $\Sigma$ , которые могут быть как замкнутые, так и разомкнутые (в случае, если  $r_{ab}$ -сфера пересекает поверхность  $\Sigma$  на её границе, либо если поверхность  $\Sigma$  не односвязна). Поставим задачу определить число  $\Lambda$  линий пересечения и отметить на каждой из линий пересечения точки через приблизительно равные расстояния, измеренные вдоль линии (далее будем называть линии пересечения  $r_{ab}$ -сферы с поверхностью  $\Sigma$   $l$ -линиями). Введём для этого понятие «матрица сферы». В сферических координатах разделим  $r_{ab}$ -сферу на параллели и меридианы с заданными шагами  $\Delta\zeta, \Delta\eta$  по горизонтали и по вертикали соответственно:

$$\zeta \in [0, 2\pi], \eta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right], m = \left\lceil \frac{2\pi}{\Delta\zeta} \right\rceil + 1, n = \left\lceil \frac{\pi}{\Delta\eta} \right\rceil + 1 \quad (3)$$

$r_{ab}$ -сфера разделится на  $nm$  участков, при этом каждому  $i, j$ -участку ( $i = \overline{0, n-1}; j = \overline{0, m-1}$ ) будут поставлены в соответствие значения

$$\zeta_j = j\Delta\zeta, \eta_i = -\frac{\pi}{2} + i\Delta\eta, i = \overline{0, n-1}; j = \overline{0, m-1} \quad (4)$$

Тогда определим матрицу сферы для  $r_{ab}$ -сферы как матрицу  $n \times m$ , значение  $i, j$ -элемента которой равно 1, если через  $i, j$ -участок сферы проходит хотя бы одна из  $l$ -линий, и равно 0, если ни одна из  $l$ -линий не проходит через  $i, j$ -участок. Для того, чтобы определить, проходит ли одна из  $l$ -линий через конкретный  $i, j$ -участок, достаточно определить, пересекает ли один из 4-х (или 3-х) криволинейных отрезков  $i, j$ -участка поверхность  $\Sigma$ . Будем полагать, что размеры  $\Delta\zeta, \Delta\eta$  достаточно малы для того, чтобы судить о пересечении/не пересечении криволинейным отрезком поверхности  $\Sigma$  через условие расположения концов этого отрезка по разные/по одну сторону от поверхности  $\Sigma$  соответственно. Таким образом, матрица сферы может быть вычислена.

#### 3.1 Поиск второй точки тетраэдра (метод «мини-сфер»)

Далее матрица сферы будет использована для поиска  $l$ -линий. Будем в цикле итерировать по  $i, j$ -участкам, в которых замечено пересечение  $r_{ab}$ -сферы с поверхностью  $\Sigma$  (они соответствуют элементам матрицы со значением 1). Критерий выхода из цикла – все элементы матрицы сферы стали равны 0. На каждом шаге цикла будем пытаться выстроить ряд точек на  $l$ -линии, проходящей через текущий  $i, j$ -участок, считая в качестве первой точки  $\lambda_1^l$  на  $l$ -линии ту точку, в которой одно из рёбер  $i, j$ -участка пересекает поверхность  $\Sigma$  (точка может быть найдена

методом бисекции). Зная первую точку  $\lambda_1^l$  на  $l$ -линии и матрицу сферы, выполним «следование» по  $l$ -линии методом «мини-сфер», который заключается в следующем. Очертим сферу достаточно малого радиуса  $\tau$  с центром в  $\lambda_1^l$  и определим все точки пересечения  $\tau$ -сферы с  $l$ -линией. Поскольку  $l$ -линия есть результат пересечения поверхности  $\Sigma$  с  $r_{ab}$ -сферой, то искомые точки являются результатом пересечения трёх объектов:  $r_{ab}$ -сферы, поверхности  $\Sigma$  и  $\tau$ -сферы. А т. к. пересечение есть операция коммутативная, то будет выгоднее сначала найти пересечение двух сфер:  $r_{ab}$ -сферы и  $\tau$ -сферы (которое есть окружность), а затем найти искомые точки  $\xi_{2,s}^l$  как пересечение поверхности  $\Sigma$  окружностью. Такой подход выгоден, так как окружность можно параметризовать и использовать для поиска пересечения метод бисекции (как и в случае двумерной задачи), разбив окружность на дуги достаточно малого заданного размера  $\Delta l_{okp}^{(1)}$ . После того, как найдены точки пересечения  $\xi_{2,s}^l$   $\tau$ -сферы с  $l$ -линией, выберем среди них ту, которая максимально отстоит от

точки  $\lambda_1^l$ , т. е.  $\rho(\lambda_2^l, \lambda_1^l) = \max_s \rho(\xi_{2,s}^l, \lambda_1^l)$  (5)

Так определяется вторая точка  $\lambda_2^l$ . На  $k$ -м шаге «следования» из точки  $\lambda_k^l$  вновь проводим  $\tau$ -сферу и ищем её точки пересечения с  $l$ -линией. Но в отличие от первого шага «следования», среди найденных точек пересечения  $\xi_{k,s}^l$  выберем одну, которая дальше всех отстоит от  $\lambda_{k-1}^l$ , найденной на  $(k-1)$ -м шаге «следования», или строго:

$$\rho(\lambda_k^l, \lambda_{k-1}^l) = \max_s \rho(\xi_{k,s}^l, \lambda_{k-1}^l), k \geq 3 \quad (6)$$

На каждом шаге «следования» вычисляется, в какой  $i, j$ -участок  $r_{ab}$ -сферы попадает точка  $\lambda_k^l$ , и значение соответствующего  $i, j$ -элемента матрицы сферы устанавливается равным 0. Это значит, что данный  $i, j$ -участок  $r_{ab}$ -сферы уже был затронут одной из  $l$ -линий, проходящих через него, и на следующем шаге внешнего цикла (по  $i, j$ -участкам) данный  $i, j$ -участок уже не будет рассматриваться как возможный для того, чтобы выстраивать из него  $l$ -линию методом мини-сфер.

«Следование» по  $l$ -линии должно остановиться, если либо а) на  $k$ -м шаге «следования» не было найдено ни одной точки пересечения  $\xi_{k,s}^l$ , либо б) на  $k$ -м шаге «следования»

обнаружилось, что точки обхода начали повторяться ( $l$ -линия уже пройдена целиком, и мы вернулись в её начало):

$$\rho(\lambda_k^l, \lambda_1^l) < \tau \quad (7)$$

Таким образом  $l$ -линия проходится в положительном направлении. Таким же способом  $l$ -линия должна быть пройдена и в отрицательном направлении от точки  $\lambda_1^l$  (это имеет смысл для не одно связных поверхностей). Каждая из точек  $\lambda_k^l$  рассматривается как одна из возможных **вторых точек**  $P_b^*$  тетраэдра. Метод мини-сфер учитывает случаи, когда несколько  $l$ -линий пересекаются или касаются друг друга в одной точке (несмотря на то, что из одного  $i, j$ -участка может быть выстроена лишь одна  $l$ -линия, другие  $l$ -линии, пересекающиеся с ней или касающиеся её в этом же  $i, j$ -участке будут выстроены «следованием» из других  $i, j$ -участков, т. к. элементы матрицы сферы для смежных  $l$ -линий не «зануляются» при обходе данной  $l$ -линии).

### 3.2 Поиск третьей точки тетраэдра

Проведём в треугольнике  $P_a P_b P_c$  высоту  $P_c H$ . Для поиска образа  $H^*$  точки  $H$  можно воспользоваться пропорцией:

$$\frac{P_a^* H^*}{P_a^* P_b^*} = \frac{P_a H}{P_a P_b} \Rightarrow \overline{P_a^* H^*} = \frac{|P_a^* P_b^*| |P_a H|}{|P_a P_b|^2} = \frac{|P_a H|}{r_{ab}} \frac{P_a^* P_b^*}{r_{ab}} \quad (8)$$

Проведём в точке  $H^*$  плоскость, перпендикулярную отрезку  $P_a^* P_b^*$ , а в этой плоскости очертим окружность с центром в точке  $H^*$  радиуса  $|P_c H| = h$ . Тогда точка  $P_c^*$  может быть получена как пересечение этой окружности с поверхностью  $\Sigma$ . Для поиска точек пересечения вновь используем метод бисекции, разбив окружность на дуги достаточно малого заданного размера  $\Delta l_{okp}^{(2)}$ . Координаты точек, разделяющих окружность на равные дуги, могут быть вычислены по формулам:

$$x_{i0} = x_a^* + (x_b^* - x_a^*) \mu + h \frac{(-z_b^* - z_a^*) (x_b^* - x_a^*) \cos \theta_i - r_{ab} (y_b^* - y_a^*) \sin \theta_i}{r_{ab} \sqrt{(x_b^* - x_a^*)^2 + (y_b^* - y_a^*)^2}} \quad (9),$$

$$y_{i0} = y_a^* + (y_b^* - y_a^*) \mu + h \frac{(-z_b^* - z_a^*) (y_b^* - y_a^*) \cos \theta_i + r_{ab} (x_b^* - x_a^*) \sin \theta_i}{r_{ab} \sqrt{(x_b^* - x_a^*)^2 + (y_b^* - y_a^*)^2}}$$

$$z_{i0} = z_a^* + (z_b^* - z_a^*) \mu + h \frac{\sqrt{(x_b^* - x_a^*)^2 + (y_b^* - y_a^*)^2}}{r_{ab}} \cos \theta_i$$

$$\text{где } \mu = \frac{1}{2} \left( \left( r_{ac}^2 - r_{bc}^2 \right) / r_{ab}^2 + 1 \right) \theta_i = 0, \sqrt{\frac{2\pi h}{l_{okp}^{(2)}}} - 1 \quad (10)$$

Каждая из найденных точек пересечения окружности с поверхностью  $\Sigma$  рассматривается как одна из возможных **третьих точек**  $P_c^*$  тетраэдра.

### 3.3 Поиск четвёртой точки тетраэдра

Четвёртая точка тетраэдра может быть восстановлена по высоте тетраэдра, опущенной из точки  $P_d^*$ , поскольку тетраэдры  $P_a P_b P_c P_d$  и  $P_a^* P_b^* P_c^* P_d^*$  равны. Опуская промежуточные выкладки, приведём лишь результат:

$$\begin{cases} x_d^* = x_a^* + \chi_{ab} (x_b^* - x_a^*) + \chi_{ac} (x_c^* - x_a^*) \pm A^* |p| \\ y_d^* = y_a^* + \chi_{ab} (y_b^* - y_a^*) + \chi_{ac} (y_c^* - y_a^*) \pm B^* |p| \\ z_d^* = z_a^* + \chi_{ab} (z_b^* - z_a^*) + \chi_{ac} (z_c^* - z_a^*) \pm C^* |p| \end{cases} \quad (11),$$

$$\text{где } p = -\frac{1}{A^2 + B^2 + C^2} \begin{vmatrix} x_a & y_a & z_a & 1 \\ x_b & y_b & z_b & 1 \\ x_c & y_c & z_c & 1 \\ x_d & y_d & z_d & 1 \end{vmatrix} \quad (12),$$

$$A = \begin{vmatrix} y_a & z_a & 1 \\ y_b & z_b & 1 \\ y_c & z_c & 1 \end{vmatrix}, B = \begin{vmatrix} z_a & x_a & 1 \\ z_b & x_b & 1 \\ z_c & x_c & 1 \end{vmatrix}, C = \begin{vmatrix} x_a & y_a & 1 \\ x_b & y_b & 1 \\ x_c & y_c & 1 \end{vmatrix} \quad (13),$$

$$A' = \begin{vmatrix} y_a^* & z_a^* & 1 \\ y_b^* & z_b^* & 1 \\ y_c^* & z_c^* & 1 \end{vmatrix}, B' = \begin{vmatrix} z_a^* & x_a^* & 1 \\ z_b^* & x_b^* & 1 \\ z_c^* & x_c^* & 1 \end{vmatrix}, C' = \begin{vmatrix} x_a^* & y_a^* & 1 \\ x_b^* & y_b^* & 1 \\ x_c^* & y_c^* & 1 \end{vmatrix} \quad (14),$$

а значения  $\chi_{ab}, \chi_{ac}$  могут быть найдены из системы:

$$\begin{cases} \chi_{ab} (x_b - x_a) + \chi_{ac} (x_c - x_a) = x_d - x_a - Ap \\ \chi_{ab} (y_b - y_a) + \chi_{ac} (y_c - y_a) = y_d - y_a - Bp \\ \chi_{ab} (z_b - z_a) + \chi_{ac} (z_c - z_a) = z_d - z_a - Cp \end{cases} \quad (15)$$

(следует взять такие два из уравнений системы 15, чтобы выбранная подсистема имела наибольший определитель). Для определения знака в (11) следует воспользоваться

$$\text{критерием: } \text{sign} \begin{vmatrix} x_a^* & y_a^* & z_a^* & 1 \\ x_b^* & y_b^* & z_b^* & 1 \\ x_c^* & y_c^* & z_c^* & 1 \\ x_d^* & y_d^* & z_d^* & 1 \end{vmatrix} = \text{sign} \begin{vmatrix} x_a & y_a & z_a & 1 \\ x_b & y_b & z_b & 1 \\ x_c & y_c & z_c & 1 \\ x_d & y_d & z_d & 1 \end{vmatrix} \quad (16)$$

Далее найденная точка  $(x_d^*, y_d^*, z_d^*)$  также должна быть проверена на близость к поверхности  $\Sigma$ . Если выполнено условие  $\rho(\Sigma, (x_d^*, y_d^*, z_d^*)) < \delta_k$  (17),

тогда считаем, что  $(x_d^*, y_d^*, z_d^*)$  - **четвёртая точка**  $P^*$  тетраэдра, ищем преобразование движения, переводящее тетраэдр  $P_a P_b P_c P_d$  в  $P_a^* P_b^* P_c^* P_d^*$  [4], и заносим параметры преобразования в память. После проверки условия близости, независимо от результата проверки переходим к рассмотрению следующего кандидата на **третью точку**.

Таким образом, имеем всего 13 параметров, влияющих на точность и скорость алгоритма -  $\Delta u, \Delta v, \Delta \zeta, \Delta \eta, \tau, \Delta l_{okp}^{(1)}, \Delta l_{okp}^{(2)}, N_{div}, \Delta \phi, \Delta \psi, L_1, L_2, L_3$ , где

$N_{div}, \Delta \phi, \Delta \psi, L_1, L_2, L_3$  - параметры, используемые в процедуре уточнения. Процедура уточнения для трёхмерного случая отличается от последней для двумерного случая тем, что окрестности точек являются не кругами, а сферами. Соотношения для координат варьируемых точек будут выглядеть следующим образом:

**1-я варьируемая точка тетраэдра**  $P_{am}^*(x_{am}^*, y_{am}^*, z_{am}^*)$

$$x_{am}^* = x_a^* + r \cos \psi \cos \phi; y_{am}^* = y_a^* + r \cos \psi \sin \phi; z_{am}^* = z_1^* + r \sin \psi \quad (18)$$

**2-я варьируемая точка тетраэдра**  $P_{bm}^*(x_{bm}^*, y_{bm}^*, z_{bm}^*)$

$$\left\{ \begin{array}{l} x_{bm}^* = x_{am}^* + \frac{\Delta x}{d} r_{ab} \cos \left( \frac{ia}{L_1} \right) + \frac{\sin \left( \frac{ia}{L_1} \right)}{\sqrt{\Delta x^2 + \Delta y^2}} \left( -\Delta x \Delta z \cos \frac{2\pi j}{L_2} + R \Delta y \sin \frac{2\pi j}{L_2} \right) \\ y_{bm}^* = y_{am}^* + \frac{\Delta y}{d} r_{ab} \cos \left( \frac{ia}{L_1} \right) + \frac{\sin \left( \frac{ia}{L_1} \right)}{\sqrt{\Delta x^2 + \Delta y^2}} \left( -\Delta y \Delta z \cos \frac{2\pi j}{L_2} - R \Delta x \sin \frac{2\pi j}{L_2} \right) \\ z_{bm}^* = z_{am}^* + \frac{\Delta z}{d} r_{ab} \cos \left( \frac{ia}{L_1} \right) + \sin \left( \frac{ia}{L_1} \right) \sqrt{\Delta x^2 + \Delta y^2} \cos \frac{2\pi j}{L_2}, i = \overline{0, L_1}, j = \overline{0, L_2 - 1} \end{array} \right. \quad (19)$$

где  $d = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$ ;  $\Delta x = x_b^* - x_{am}^*$ ;  $\Delta y = y_b^* - y_{am}^*$ ;  $\Delta z = z_b^* - z_{am}^*$  (20),

$$\cos \alpha = \frac{1}{2} \left( \frac{d^2 + r_{ab}^2 - \delta_k^2}{r_{ab} d} \right)^2 - 1 \quad (21)$$

**3-я варьируемая точка тетраэдра**  $P_{cm}^*(x_{cm}^*, y_{cm}^*, z_{cm}^*)$

$$\left\{ \begin{array}{l} x_{cm}^* = \Delta x + h \frac{(-z_{bm}^* - z_{am}^*) (x_{bm}^* - x_{am}^*) \cos \chi_i - r_{ab} (y_{bm}^* - y_{am}^*) \sin \chi_i}{r_{ab}} \quad (22), \\ y_{cm}^* = \Delta y + h \frac{(-z_{bm}^* - z_{am}^*) \sqrt{(x_{bm}^* - x_{am}^*)^2 + (y_{bm}^* - y_{am}^*)^2} + r_{ab} (x_{bm}^* - x_{am}^*) \sin \chi_i}{r_{ab} \sqrt{(x_{bm}^* - x_{am}^*)^2 + (y_{bm}^* - y_{am}^*)^2}}, \\ z_{cm}^* = \Delta z + h \frac{\sqrt{(x_{bm}^* - x_{am}^*)^2 + (y_{bm}^* - y_{am}^*)^2}}{r_{ab}} \cos \chi_i \end{array} \right.$$

где

$$\Delta x = x_{am}^* + \mu (x_{bm}^* - x_{am}^*) \Delta y = y_{am}^* + \mu (y_{bm}^* - y_{am}^*) \Delta z = z_{am}^* + \mu (z_{bm}^* - z_{am}^*) \quad (23),$$

$$\chi_i = -\arccos \frac{r_{ab} \Delta z - q (z_{bm}^* - z_{am}^*)}{r_{ab} \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2 - q^2}} + (24),$$

$$+ \left( \frac{2i}{L_3} - 1 \right) \arccos \frac{h^2 - \delta_k^2 + \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}}{2h \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2 - q^2}}, i = \overline{0, L_3}$$

$$q = (\Delta x (x_{bm}^* - x_{am}^*) + \Delta y (y_{bm}^* - y_{am}^*) + \Delta z (z_{bm}^* - z_{am}^*)) / r_{ab} \quad (25),$$

а  $\mu$  определяется из (10).

Для выражения координат четвёртой точки следует использовать формулы, приведённые в разделе 3.3, при условии, что к нижнему индексу всех координат добавлена буква "m" (признак варьируемой точки).

Алгоритм для трёхмерной задачи был реализован в программном модуле «3DAnalyzer». Ниже приведены два примера тестирования модуля на поверхностях.

**Тест №3 (точного соответствия между множествами нет).** Поверхность  $\Sigma$  задана уравнением  $f(x) = 50 \sin(0.1 \sqrt{x^2 + y^2})$  на отрезке  $[-60; 60]$ . В качестве множества  $P$  были взяты точки поверхности  $g(x) = 0.015(x^2 + y^2) - 150$  на отрезке  $[-30; 30]$  через расстояние 0.9, сдвинутые затем на вектор  $\{100; 320; 200\}$  и повёрнутые на углы  $(\frac{\pi}{2} - 0.2; -\frac{\pi}{4}; \frac{\pi}{5})$  относительно начала координат. Параметры алгоритма:  $\Delta u = 1/12$ ,  $\Delta v = 1/12$ ,  $\Delta \zeta = \Delta \eta = 0.1$ ,  $\tau = 2$ ,  $\Delta l_{okp}^{(1)} = \Delta l_{okp}^{(2)} = 0.04\pi h$  (такая связь задавалась внутри программы),  $N_{div} = 3$ ,  $\Delta \phi = \Delta \psi = \delta_k / (r N_{div})$  (такая связь задавалась внутри программы),  $L_1 = L_2 = L_3 = 10$ . Время работы программы - около 2 мин.

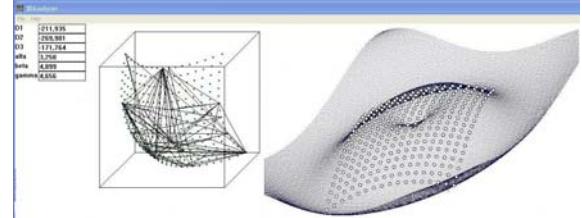


Рис 5: Совмещение поверхностей  $f(x) = 50 \sin(0.1 \sqrt{x^2 + y^2})$  и  $g(x) = 0.015(x^2 + y^2) - 150$  с помощью модуля «3DAnalyzer»

Слева на рис.5 (в кубе) изображены точки множества  $P$ . Справа изображена поверхность  $\Sigma$  и точки множества  $P$ , совмещённые с ней (обозначенные белым цветом).

#### 3.4 Параметры алгоритма

данний момент вопрос о влиянии параметров  $\Delta u, \Delta v, \Delta \zeta, \Delta \eta, \tau, \Delta l_{okp}^{(1)}, \Delta l_{okp}^{(2)}, N_{div}, \Delta \phi, \Delta \psi, L_1, L_2, L_3$  на эффективность алгоритма мало исследован. Но экспериментальным путём были установлены некоторые более предпочтительные соотношения между параметрами. 1)  $\Delta u = \Delta v \approx 0.05$ . Меньшие значения существенно снижают скорость алгоритма, но не сильно повышают точность. Тем более, что в ходе процедуры уточнения решения всё равно уточняются. 2)  $\Delta \zeta, \Delta \eta$  следует выбирать такими, чтобы в пределах каждого  $i, j$ -участка  $r_{ab}$ -сферы поверхность имела малые перепады кривизны. 3)  $\tau, \Delta l_{okp}^{(1)}, \Delta l_{okp}^{(2)}$  следует, как и два предыдущих параметра, выбирать, исходя из кривизны: между любыми двумя точками поверхности, отстоящими на  $\tau, \Delta l_{okp}^{(1)}, \Delta l_{okp}^{(2)}$ , кривизна не должна иметь сильных скачков. Излишнее уменьшение параметров  $\Delta \zeta, \Delta \eta, \tau, \Delta l_{okp}^{(1)}, \Delta l_{okp}^{(2)}$  существенно снижает скорость алгоритма, но не сильно повышают точность. 4)  $N_{div}$  сильно влияет на скорость. Достаточно брать равным 3. То же относится и к  $L_1, L_2, L_3$  - достаточно взять их равными 10. 5)  $\Delta \phi, \Delta \psi$  лучше всего брать не константами, а функциями:  $\Delta \phi = \Delta \psi = \delta_k / (r N_{div}), r \neq 0$  (при  $r = 0 \Delta \phi = \Delta \psi = 0$ ). При таком выборе  $\delta_k$ -окрестность «заполняется» точками более равномерно.

#### 4. СОВМЕЩЕНИЕ ИЗОБРАЖЕНИЙ

В код модуля «3DAnalyzer» были добавлены функции для чтения/записи файлов формата «bmp» с преобразованием данных файла в поверхность и обратно (поверхность строится по регулярной сетке, а для получения координаты  $z$  значение, читаемое из файла, делится на нормирующий коэффициент  $\omega$ ). После совмещения полученных поверхностей в результирующий файл записывались обе поверхности, а на их границе – точки чёрного цвета (чтобы совмещённый фрагмент было проще увидеть).

##### Тест №4 (Поиск фрагмента изображения на нём же самом)

На рис. 6(в) приведён результат совмещения фрагмента изображения с ним же. Размеры изображения: 473x800, размеры фрагмента: 326x300. Нормирующий коэффициент  $\omega = 100000$  (был подобран эмпирически). Параметры алгоритма:  $\Delta u = 0.0528$ ,  $\Delta v = 0.0313$ ,  $\Delta \zeta = \Delta \eta = 0.1$ ,  $\tau = 10$ ,

$$\Delta l_{\text{окр}}^{(1)} = \Delta l_{\text{окр}}^{(2)} = 0.04\pi h, \quad N_{\text{div}} = 3, \quad \Delta \phi = \Delta \psi = \delta_k / (rN_{\text{div}}), \quad L_1 = L_2 = L_3 = 10 \cdot$$

Время работы программы – около 2 мин 25 с.



Рис 6 (а, б): Исходные данные теста №4 – изображение (слева) и его фрагмент (справа)



Рис 6 (в): Результат теста №4 – совмещённое изображение

##### Тест №5 (Совмещение двух разных изображений)

На рис. 7(в) приведён результат совмещения двух изображений, представляющими вид на один объект с разных ракурсов. Размеры первого изображения: 473x800, размеры второго изображения: 397x670. Параметры алгоритма такие же, как для предыдущего теста. Время работы программы – около 2 мин 45 с.



Рис 7 (а, б): Исходные данные теста №5 – два разных изображения



Рис 7 (в): Результат теста №5 – совмещённое изображение

#### 5. ЗАКЛЮЧЕНИЕ

Разработанный алгоритм показал удовлетворительные результаты как в применении для совмещения 3D геометрических фигур, так и для совмещения 2D изображений. Учитывая тот факт, что на данный момент разработка модуля «3DAnalyzer» не завершена, сделать окончательные выводы об эффективности его работы в плане скорости и точности не представляется возможным. Также остаётся не исследованным вопрос об оптимальных значениях параметров алгоритма в зависимости от типа поверхности или изображения (в случае с изображением добавляется ещё один параметр – нормирующий

коэффициент  $\omega$ ). В качестве недостатков алгоритма можно отметить следующие. Во-первых, на текущей стадии разработки алгоритм не учитывает преобразования масштабирования и симметрии. Во-вторых, алгоритм требует предварительной фильтрации шумов.

Между тем, алгоритм обладает серьёзными достоинствами, среди которых следует отметить, что во-первых, алгоритм способен работать на неодносвязных поверхностях (пример: тор) и поверхностях с самопересечением, во-вторых, рассматривает существование нескольких решений (пример: рис. 3). Задачи, которые предстоит решить в будущем в рамках затронутой проблемы – провести полное исследование влияния параметров алгоритма на скорость и точность, построив соответствующие графики, а также усовершенствовать алгоритм, устранив перечисленные выше недостатки.

#### 6. ССЫЛКИ

- [1] ПоповИвашин, Д. И. **Математические методы и комплекс программ частотно-пространственного анализа стереопар изображений** / Д. И. Ивашин // Диссертация на соискание степени кандидата технических наук
- [2] Самарина, О. В. **Использование инвариантов при поиске соответствия изображений** / О. В. Самарина // Вестник Югорского государственного университета. - Ханты-Мансийск, 2008. выпуск 1(8). – С. 110-113.
- [3] Самарина, О. В. **Программный комплекс обработки одноканальных и трёхканальных изображений** / О. В. Самарина // Вестник ЮГУ.- Ханты-Мансийск, 2008. выпуск 3(10). – С. 67-71.
- [4] Самойлов, А. А. **Определение параметров трёхмерного преобразования движения по 4-м заданным точкам и их образам**/ А. А. Самойлов // Сборник трудов аспирантов, магистрантов и соискателей. Технические науки. – Н. Новгород: ННГАСУ, 2012. ISBN 978-5-87941-663-3
- [5] Самойлов, А. А. **Поиск оптимального совмещения кривой и точечного множества в двумерном пространстве**/ Е. В. Попов, С. И. Ротков, А. А. Самойлов // Приволжский научный журнал №4(20). Периодическое научное издание. Н. Новгород, ННГАСУ, 2011. ISSN 1995-2511
- [6] M. Thomas, Irene. **A Cross-Correlation Method for Merging Electron Crystallographic Image Data**/ Irene M. Thomas, Michael F. Schmid // Verna and Marrs McLean Department of Biochemistry and W. M. Keck Center for Computational Biology. Baylor College of Medicine, Houston, TX 77030
- [7] Tao XU. **Research of Real-time Wide Field Image Merging Based on Multi-Cameras**/ XU Tao, CEN Zhao-feng, LI Xiao-tong // State Key Lab of Modern Optical Instrumentation, Zhejiang University, Hang Zhou 310027, China

#### Об авторах

Александр Александрович Самойлов – аспирант ННГАСУ.

E-mail: [greatlimit@rambler.ru](mailto:greatlimit@rambler.ru).

Евгений Владимирович Попов – д.т.н., профессор ННГАСУ. E-mail: [popov@sandy.ru](mailto:popov@sandy.ru).

## Интерактивное моделирование автомобильных красок

А.Г. Волобой, С.В. Ершов, С.Г. Поздняков

Институт прикладной математики им. М.В.Келдыша РАН, Москва

### Аннотация

В работе описан подход к моделированию многослойных красок в интерактивном режиме. Была поставлена и решена задача генерации внешнего вида краски, исходя из ее внутренней структуры. Для достижения интерактивности были разработаны оригинальные подходы к решению задачи моделирования света в красящем слое, быстрой визуализации сцены, предложена архитектура программного комплекса. Также разработанные средства были успешно использованы для задачи автоматического подбора состава краски по ее внешнему виду.

**Ключевые слова:** реалистичная визуализация материалов, взаимодействие света с красящим слоем, физически корректное моделирование, ДФО, оптические свойства поверхности.

### 1. ВВЕДЕНИЕ

Компьютерная графика изначально зародилась как эффективное и мощное средство связи между человеком и вычислительной машиной. С появлением методов синтеза реалистичных изображений, основанных на физически корректном моделировании распространения света, область применения компьютерной графики существенно расширилась. Созданные алгоритмы и программные средства стали представлять интерес для использования в архитектуре, проектировании систем освещения, в автомобильной промышленности и других областях.

Важным направлением, получившим развитие в последние годы, стало моделирование и визуализация новых оптически сложных материалов, таких как многослойные краски со сложной микроструктурой (типа перламутровых и «металлик») в автомобильной промышленности. Разработанные программные средства позволяют моделировать и визуализировать краски, состоящие из плоскопараллельных однородных слоев, каждый из которых составлен из прозрачного лака и взвешенных в нем пигментных частиц и интерференционных чешуек.

На рис. 1 представлен автомобиль, окрашенный такой краской, при дневном освещении. Цвет этих красок зависит от угла зрения и направления освещения. Он может резко меняться («переключаться») даже при плавном изменении этих углов.



Рис. 1. Автомобильная краска при дневном освещении.

Светорассеивающие свойства таких красок не могут быть заданы скалярными коэффициентами, а требуют более сложного описания с помощью двунаправленных функций отражения света. Другим аспектом визуального восприятия краски является различное освещение, она выглядит различно утром и в середине дня, при солнечной или пасмурной погоде. На рис. 1 освещение задано изображением с высоким диапазоном яркостей (HDRI – High Dynamic Range Image).

Гораздо дешевле смоделировать на компьютере внешний вид краски и исследовать ее при различных условиях освещение, чем произвести опытную партию краски и покрасить ею реальный автомобиль. В настоящее время такой подход к моделированию красок применяется не только в автомобильной промышленности, но и для окраски большого спектра других товаров (мобильные телефоны, персональные компьютеры, бытовая техника и пр.) Именно внешний вид изделия часто определяет его коммерческий успех. Используя средства компьютерной графики, можно понять, как будет выглядеть изделие уже на начальных стадиях его разработки.

Моделирование современных красок с металлическими и перламутровыми эффектами, чей внешний вид изменяется в зависимости от условий освещения и наблюдения, является нетривиальной задачей. Иногда краски разрабатываются непосредственно для данного изделия с учетом его формы и размеров. Чтобы разработка красок, базируясь на физических и химических свойствах их составляющих, была эффективной, моделирование внешнего вида должно выполняться с интерактивной скоростью.

### 2. СУЩЕСТВУЮЩИЕ РЕШЕНИЯ ДЛЯ МОДЕЛИРОВАНИЯ КРАСОК

Одни из подобных первых разработок были проведены для акварельных красок [1]. Авторы применяют широко известную модель Кубелка-Мунка, которая является упрощенным решением уравнения переноса излучения (света) [2]. В настоящее время, с развитием компьютеров и математических методов стал возможным непосредственный расчет распространения света в красящих слоях с использованием уравнения переноса.

В последние годы в литературе по компьютерной графике опубликовано достаточное число работ, относящихся к моделированию оптически сложных красок с перламутровым и металлическим эффектами. Большинство из них используют подходы, сильно отличающиеся от представленного ниже. Значительная их часть ориентирована лишь на объяснение и качественное воспроизведение какого-либо одного эффекта на предельно упрощенной модели. Например, интерференционные модели, разработанные для моделирования мыльных пузырей, масляных пятен [3], или жемчуга [4], которые описывают физические явления для непрерывных тонких пленок, малопригодны для аккуратного расчета многократного рассеяния света на относительно небольших перламутровых чешуйках, расположенных в связующем веществе и окруженных зернами поглощающего пигмента и металлическими чешуйками.

По тем же самым причинам не могут быть применены методы, использующие множество упрощений таких как: изотропия пленок [5], однородное освещение внутри пленки [6], ограничивающиеся только случаем однослойной краски [5, 7] или фиксирующие ориентацию чешуек параллельно поверхности краски [8]. Существуют работы по вычислению характеристик краски для формально сколь угодно сложной (и потому реалистичной) внутренней структуры, но они вычислительно дороги и сложны в реализации [9].

Существуют работы, применяющие непосредственное моделирование распространения света внутри объема краски, описываемого на макроскопическом уровне. Использованные методы позволяют моделировать все важные для настоящего приложения оптические эффекты. Некоторые из таких подходов были специально разработаны для аккуратного расчета радиальных и перламутровых эффектов [10, 11]. Однако, требуется весьма длительные вычисления, поскольку огромное число лучей должно быть прорассчитано через объем краски, чтобы вычислить распределение отраженного света с малыми погрешностями. Те же из них, что используют явное моделирование внутренней структуры краски, еще менее пригодны, т.к. на стадии разработки неизбежно частое изменение внутренней структуры объема краски. Этот недостаток может быть частично преодолен с помощью неявного статистического описания структуры краски [12-14] (т.е. введения соответствующей длины свободного пробега, с помощью которой разыгрывается длина пробега света между событиями рассеяния на чешуйках). Все эти методы ресурсозатратны и без выполнения полных вычислений они не позволяют на качественном уровне оценить результирующую двунаправленную функцию отражения (ДФО), описывающую оптические свойства поверхности. Поэтому они непригодны для интерактивного программного комплекса.

Интерактивная скорость визуализации возможна для таких теоретических моделей отражения, как Cook-Torrance [15]. Они основаны на структурах, использующих микрофасетки, и весьма эффективны на вычислительном уровне и дают хорошие приближения ДФО для типичных материалов. Однако, некоторые параметры, использующиеся в этих моделях, не имеют простой физической интерпретации. Отсутствие простых соотношений между реальными физическими величинами и параметрами моделей сильно усложняют их применение для разработки красок с заданным внешним видом. Этот недостаток делает рассматриваемые методы неприемлемыми в нашем случае.

Таким образом, существующие подходы непригодны для нашей задачи либо из-за низкой вычислительной эффективности (методы прямой трассировки), или из-за слишком сильного упрощения реальной структуры краски (теоретические модели).

### 3. ВЫЧИСЛЕНИЕ ОПТИЧЕСКИХ СВОЙСТВ МНОГОСЛОЙНЫХ КРАСОК

Задача моделирования цвета краски может быть поставлена в формулировке: требуется вычислить распределение отраженного света (двунаправленную функцию отражения – ДФО), аккуратно учитывающее многократное рассеяние света в объеме краски и позволяющее изменять состав краски в рамках технологических возможностей. В этой формулировке освещение предполагается параллельным однородным некогерентным пучком, а наблюдение – в дальней зоне

(регистрируется только угловое распределение, а текстура, если она вообще наблюдается, утрачивается). Так как часто краски бывают достаточно сложными, и имеют нетривиальную зависимость цвета от угла наблюдения и падения света, то возможно использовать только наиболее гибкое, табличное представление ДФО.

«Искристость», вообще говоря, может быть описана в похожих терминах. Это некая вторая ДФО для амплитуды флюктуаций плюс закон их распределения (в большинстве моделей достаточно тривиальный).

С другой стороны, модель краски также определяется входными данными моделирования. Говоря об оптической задаче, мы в качестве «входа» предполагаем готовую структуру высохшей краски<sup>1</sup>. Она содержит «чешуйки», для которых мы используем распределение их по размерам, ориентации и положению (т.е. локальную концентрацию), а равно оптические свойства их поверхности (вычисляемой, скажем, как отражение интерференционного покрытия слоя  $\text{TiO}_2$ ). Форма чешуек не используется, ибо обычно она весьма изрезанна, и построение даже статистического описания такой фигуры – отдельная непростая задача. Помимо чешуек, есть еще «обычные» (мелкие) пигменты. Они описываются просто объемным поглощением и рассеянием (последнее часто очень незначительно и может быть заменено эмпирическим законом Хенни-Гринстейна). Все эти частицы считаются взвешенными в прозрачном лаке, так что «в среднем» геометрия изотропна, но не обязательно однородна т.к. концентрация может меняться по глубине.

В предлагаемом методе модель краски состоит из плоскопараллельных однородных слоев, каждый из которых составлен из прозрачного лака и взвешенных в нем пигментных красящих частиц и интерференционных чешуек, ответственных за «искрение» (рис. 2).

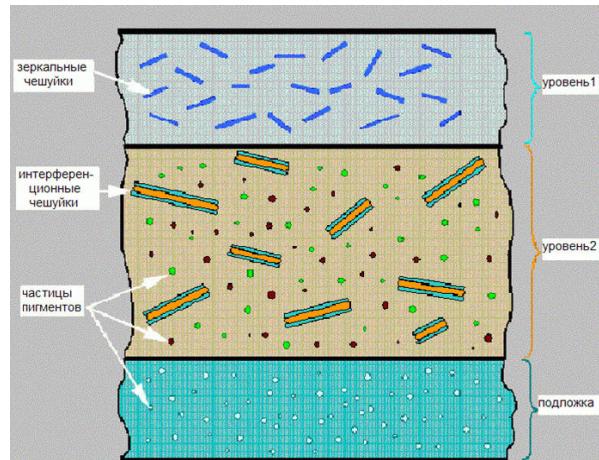


Рис. 2. Поперечное сечение используемой модели краски.

Каждый слой краски расщепляется на множество тонких субслоев таким образом, чтобы внутри каждого субслоя можно было бы пренебречь многократным рассеянием. Предполагается, что все события рассеяния света происходят в разных субслоях. Это позволяет описывать рассеяние в тонких субслоях в аналитическом виде. Такая модель рассеяния основана на статистическом подходе и аккуратно описывает взаимодействие света внутри

<sup>1</sup> Есть модели формирования краски (т.е. как капельки оной, разбрызгиваясь из пульверизатора, растекаются по поверхности и далее высыхают, лишаясь летучих компонент). Однако, это не оптическая задача.

краски, включая радужные и перламутровые эффекты. Рассеяние света каждым слоем краски вычисляется с помощью распределения рассеяния тонкими субслоями методом удвоения [16, 17].

Для расчета светоотражающей характеристики краски необходимо промоделировать движение света сквозь заданные слои ее структуры до отражения от подложки и движение его обратно. Рассеяние света всей краской вычисляется с помощью рассеяния на каждом слое методом сложения [16-18]. Стадия сложения дополняет стадию удвоения, которая вычисляется ДФО реального слоя, используя аналитические выражения для ДФО тонких воображаемых слоев. Такой подход делает возможным аккуратный учет многократного рассеяния в сложных средах. Он позволяет сформулировать в аналитическом виде весьма точное приближение для ДФО красок с перламутровым и металлическим эффектами.

Методы сложения и удвоения, изначально разработанные для расчета рассеяния в атмосферах планет, применяются к уравнению переноса излучения, преобразованного к форме, удобной для таких задач. В этом смысле метод Hanrahan and Kueger [14] близок, поскольку использует ту же теорию переноса излучения. В работе [14] интегральное уравнение переноса решалось методом Монте-Карло, в предложенном же подходе применяется прямой сеточный метод, который значительно быстрее.

Границы между тонкими субслоями являются воображаемыми – на них не происходит отражения и преломления света, в отличие от границ реальных слоев в многослойных красках. Кроме этого, оказывается, что с помощью дальнейших упрощений, модель может аккуратно учитывать рассеяние света на металлических чешуйках, при этом эффективная ДФО также может быть вычислена в аналитическом виде. Ошибки, вносимые такими упрощениями, пренебрежимо малы при тех слабых плотностях чешуек, которые применяются в реальных красках (объемная концентрация частиц PVC ~5-10%).

Данная аналитическая модель была специально разработана для описания рассеяния света на ансамблях гладких металлических или перламутровых пластинок, состоящих из слюдяных частиц, покрытых двуокисью титана ( $TiO_2$ ) [19]. В зависимости от толщины покрытия слюды двуокисью титана перламутровая частица отражает свет различного цвета. Она весьма точна, когда пластинки расположены внутри почти прозрачного вещества с низкой концентрацией пигментов. Последнее условие выполняется в случае реальных красок, которые практически прозрачны для того чтобы подчеркнуть металлические эффекты, например, искрение (sparkles).

К недостаткам предложенного подхода, использующего приближение непрерывной среды, следует отнести:

- приближение независимости событий рассеяния, которое не всегда является достаточно точным. Например, свет, рассеянный одной чешуйкой, может быть отражен другой чешуйкой (или подложкой), а потом опять рассеяться на первой чешуйке. Однако стоит заметить, что вероятность таких событий мала;
- использование концентрации площади чешуек в единице объема вместо использования размеров и концентрации самих чешуек. В частности, ДФО для случая небольшой концентрации крупных чешуек и соответствующей концентрации мелких чешуек будут идентичны. Такое упрощение можно считать достаточно точным только при некоторых условиях,

налагаемых на максимальный размер чешуек и их концентрацию.

Так как существуют случаи, когда данные ограничения существенны, то была реализована также «полная» модель вычисления ДФО, свободная от этих недостатков. Она основана на Монте-Карло трассировке лучей в прозрачной среде лака с использованием явной геометрии. «Явность» тут применима только к крупным частицам (размеры которых  $>>$  длины волны). Мелкие же пигменты (красители) описываются в терминах поглощения и рассеяния среды, потому как бессмысленно говорить о взаимодействии луча с частицей с размерами  $<<$  длины волны. Мелкие пигменты, таким образом, описываются неявной геометрией. Этот подход является комбинацией классической прямой Монте-Карло трассировки лучей и уравнения переноса. Между частицами прохождение луча описывается уравнением переноса, которое решается методом Монте-Карло. Взаимодействие же луча с крупными частицами обрабатывается, как в классической трассировке, т.е. находится точка попадания, разыгрывается направление рассеяния и пр.

Было произведено сравнение между быстрой и «полной» моделями вычислений для различных концентраций, размеров и типов частиц. Для прозрачных диэлектрических частиц отличия невелики и редко превышают 10%. Для металлических чешуек отличие моделей при предельно высоких концентрациях крупных чешуек может достигать даже 50%, но типичная ошибка редко превосходит 15% - 20%. При этом более точная «полная» модель вычисляется за минуты, в то время как быстрая модель – практически в реальном времени (за доли секунды).

Другое явление, явно зависящее от размера чешуек, – это грануляция или флуктуации яркости [14]. По мере увеличения размера чешуек, отдельные чешуйки становятся различимы глазом и действуют как маленькие сияющие зеркальца (эффект искрения). Искры могут быть описаны как специальная текстура, существенно зависящая от условий освещения и наблюдения. В дальнейшем текстура накладывается на поверхность окрашенного объекта, увеличивая правдоподобность визуализации краски «металлик».

Вычисление этого эффекта в первом приближении несложно. Основное приближение связано с тем, что «искрение» хорошо видно тогда, когда концентрация искорок невелика, иначе они сливаются и пропадают. При этом можно пренебречь многократными переотражениями. Точка на поверхности краски видна как искорка, если там есть чешуйка и если её нормаль направлена приблизительно по биссектрисе направления освещения и наблюдения (чтобы было зеркальное отражение). Допустимое отклонение направления нормали равно угловому разрешению глаза. Зная распределение частиц по направлению (обычно оно близко к гауссовому) и их концентрацию, мы можем вычислить вероятность того, что данная точка краски «искрится». В противном случае она чёрная. Таким образом и синтезируется случайная текстура.

В случае, когда есть прочие, не бликующие пигменты, полученную текстуру необходимо сложить с ДФО прочих пигментов.

#### 4. СТРУКТУРА ПРОГРАММНОГО КОМПЛЕКСА МОДЕЛИРОВАНИЯ КРАСОК

Модель краски является важной составляющей разработанного интерактивного программного комплекса

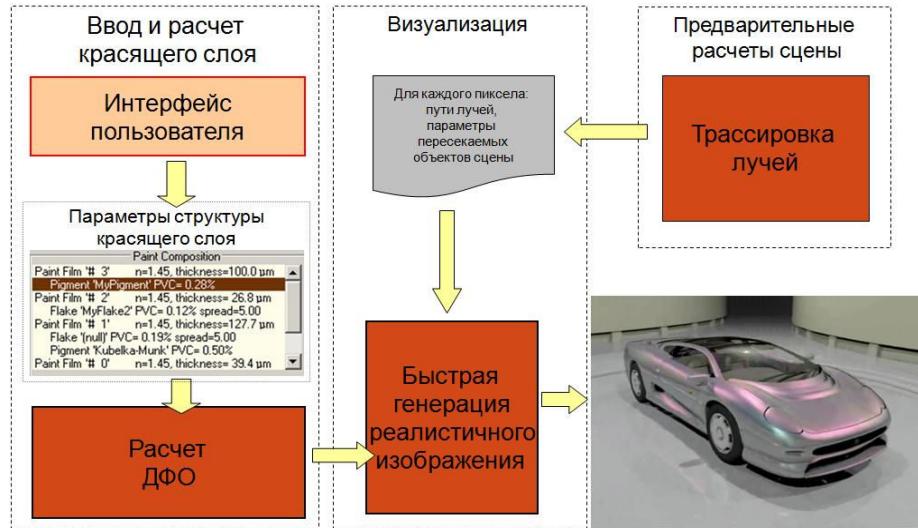


Рис. 3. Схема программного комплекса интерактивного моделирования многослойной краски.

моделирования внешнего вида красок, исходя из их состава. Структура комплекса представлена на рис. 3.

Интерфейсная часть программного комплекса позволяет полностью задать структуру моделируемой краски. Для каждого слоя прозрачного лака пользователь задает характеристика лака, виды содержащихся в слое поглощающих частиц и рассеивающих чешуек, которые могут быть либо зеркальными, либо интерференционными. Для каждого вида частиц или чешуек задается их концентрация в слое и размер. При задании зеркальных чешуек можно указать коэффициент отражения в виде спектрограммы. Для интерференционных чешуек необходимо специфицировать толщины слюдяного слоя и покрытия двуокисью титана. Также управляемым параметром является разброс значений нормалей к чешуйкам, позволяющий контролировать их ориентацию в слое. Параметры могут быть изменены в интерактивном режиме, естественно, в технологически допустимых пределах.

Используя эти параметры и структуру краски, вычисляется ДФО для описанной многослойной модели. Полученная ДФО поступает в модуль быстрой визуализации виртуальных сцен, содержащих объекты, окрашенные разрабатываемой краской. Оптическое моделирование с помощью трассировки лучей обычно требует значительного времени. Эта проблема особенно критична, когда результаты трассировки лучей зависят от параметров моделирования, например, положения или ориентации камеры.

Основная идея интерактивной визуализации состоит в сохранении всего массива трассируемых лучей в совокупности с необходимыми вспомогательными данными на стадии предварительного расчета освещенности сцены и его последующим использованием для быстрой визуализации. Количество трассируемых лучей в рассматриваемых задачах может достигать миллионов или даже миллиардов в зависимости от сложности визуализируемой сцены, размерности изображения и требований к устранению лестничного эффекта. Поэтому карты трассируемых лучей [20] сохраняются на диске в процессе предварительного расчета. Свойства всех поверхностей, за исключением

окрашенных разрабатываемой краской, также сохраняются.

На этапе интерактивной визуализации карты трассируемых лучей загружаются по мере необходимости, оптические свойства поверхностей, окрашенных краской, заменяются на рассчитанную ДФО, и формируется итоговое реалистичное изображение. Время визуализации постоянно, зависит только от разрешения изображения и не зависит от сложности геометрии сцены. Например, время, необходимое для полного вычисления ДФО двухслойной краски с перламутровым эффектом и визуализации изображения с разрешением 640 x 480 пикселей, составило всего около 0.14 секунды на компьютере с процессором Pentium III, 500 MHz. Для «полной» модели краски то же время составляет примерно 0.5–5 секунд в зависимости от параметра краски. Время счета увеличивается с усложнением структуры краски. Например, расчет «полной» модели для трехслойной краски, содержащей два вида чешуек и два вида пигментных зерен, требует от 1 до 9 секунд. В итоге предлагаемый программный комплекс также является вполне пригодным для интерактивной разработки новых красок. В таком случае в качестве результата можно сохранить полученный состав краски в формате, пригодном для ее изготовления.

## 5. ПОДБОР СТРУКТУРЫ КРАСКИ ПО ЕЕ ВНЕШНЕМУ ВИДУ

Задача автоматического подбора состава краски, применимого для ее изготовления, по внешнему виду существующей краски является одной из востребованных в автомобильной промышленности. Современные краски имеют достаточно сложный внутренний состав, ингредиенты краски, их концентрации и процесс ее создания часто являются закрытой информацией. Но даже когда изначальный состав краски известен, это не гарантирует совпадение цвета при кузовном ремонте автомобиля потому, что краска меняет свой цвет под воздействием солнечного света, осадков и химических реагентов, применяемых в дорожных работах.

Подбор состава для достижения желаемого цвета — это классическая обратная задача. В первом приближении

можно построить даже аналитическое решение. Но точность его будет невысока, а ограничений — много. Поэтому применялся подход «минимизации ошибки». В процессе подбора выбирался некий «пробный» состав и вычислялась его ДФО. На основе сравнения вычисленной и желаемой ДФО строится вариация состава, уменьшающая ошибку. Процесс повторяется итеративно. На каждом шаге приходится, по меньшей мере, один раз находить целевую функцию, и, стало быть, вычислять ДФО. Поэтому здесь существенно применение быстрого метода.

Для проверки возможностей программного комплекса моделирования красок решать задачу подбора состава краски компанией Merck были предоставлены 4 образца реальных красок, созданных с помощью чешуек Iridin: WF 67/99/250, WF 67/99/192, WF 9/00/78 и WF 9/00/27. ДФО этих образцов были измерены на измерительном комплексе [21]. После этого была предпринята попытка воссоздать состав этих красок в предположении, что реальная краска состоит из одного слоя.

Результат моделирования (рис. 4):

Удалось получить достаточно близкий цвет восстановленной краски, однако во всех случаях полное совпадение достигнуто не было. Данный эксперимент можно рассматривать, как первый шаг к решению задачи автоматического подбора состава краски, которая является весьма востребованной производителями автомобильных красок. На рисунках левая сторона каждой сферы окрашена измеренной ДФО реального образца, а правая сторона — ДФО, рассчитанной с помощью программного комплекса моделирования краски.

Первоначально было предположено, что краска однородна по толщине. Однако никакими вариациями состава не удавалось воспроизвести её ДФО. Для выяснения причин отклонения были произведены дополнительные измерения слоя краски на прозрачной подложке, и оказалось, что ДФО с разных сторон существенно отличается. Введение неоднородности по толщине позволило удовлетворительно воспроизвести ДФО с обеих сторон. В итоге была получена вполне удовлетворительная точность. Однако дальше улучшить её не удавалось.

Анализ показал, что для голубых красок одной из важнейших причин отклонения служит голубой краситель. Это вещество, в идеале, не должно бы вовсе иметь собственного рассеяния (а только поглощение). Поэтому рассеяние не паспортизуется и не

выдерживается изготовителем пигмента. Измерения двух образцов голубого красителя показали, что рассеяние пигмента в них отличается более чем вчетверо. Производитель краски подтвердил, что существенная часть ошибки связана с нестабильностью изготовления краски (как перемешали, сколько подождали до нанесения, при какой влажности сушили, и пр.) Эти факторы вообще не входят в оптическую модель, которая трактует образцы номинально одного состава как тождественные. На самом же деле они — разные.

## 6. РЕЗУЛЬТАТЫ

На основании построенной модели многослойной краски были предложены алгоритмы расчета двунаправленной функции отражения света, в общем виде описывающей внешний вид краски с различных направлений и при различных условиях освещения. Был реализован программный комплекс, позволяющий моделировать ДФО краски и реалистично визуализировать ее в некоторой сцене с интерактивной скоростью.

Разработанное решение дает приемлемые результаты при визуализации сложных автомобильных красок при различных условиях освещения, а также показывает хорошее совпадение рассчитанных и измеренных ДФО красок.

Использование комплекса расчета краски в режиме интерактивной визуализации позволяет перейти к задачам моделирования новых красок. Разработанные подходы можно в дальнейшем использовать для решения наиболее востребованной задачи в области моделирования оптически сложных красок — задачи восстановления структуры и состава краски по ее внешнему виду, так называемая задача подгонки цвета.

Проведенные эксперименты с красками фирмы Merck показали хорошее соответствие предложенной модели с реальностью. Как основной цвет результирующей краски, так и цвета таких сложных зон, как блик и около бликового яркое пятно, близки к цветам реального образца. Полученные в результате моделирования ДФО красок могут быть использованы для физически корректных расчетов освещенности в трехмерных сценах, содержащих окрашенные объекты. Например, они были использованы для анализа и выбора окраски автомобилей, мобильных телефонов и ноутбуков.

Работа поддержана грантом РФФИ № 11-01-00870, а также фирмой Integra Inc. (Япония).

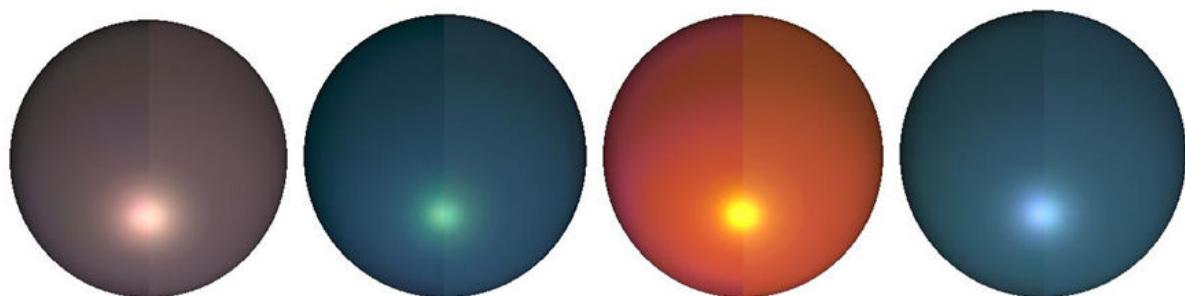


Рис. 4. Визуализация измеренной (левая половина сферы) и рассчитанной (правая половина) ДФО перламутровых красок.

## 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] B.E. Smits and G. Meyer. Newton's Colors: Simulating Interference Phenomena in Realistic Image Synthesis // 1<sup>st</sup> Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics, 1990, pp. 185-194.
- [4] N. Nagata, T. Dobashi, Y. Manabe, T. Usami and S. Inokuchi. Modelling and Visualization for a Pearl-Quality Evaluation Simulator // IEEE Transactions on Visualization and Computer, Vol. 3, No. 4, 1997, pp. 307-315.
- [5] I. Icart and D. Arques. An Illumination Model for a System of Isotropic Substrate – Isotropic Thin Film with Identical Rough Boundaries // 10<sup>th</sup> Eurographics Rendering Workshop, 1999, pp. 260-272.
- [6] P. Kubelka and F. Munk. Zeits. Tech. Physik (1931), 12, 593.
- [7] P. Callet. Pertinent Data for Modelling Pigmented Materials in Realistic Rendering // Computer Graphics Forum, Vol. 15, No. 2, 1996, pp. 119–128.
- [8] P. Callet. Physically Based Rendering of Metallic Paints and Coated Pigments // Visualization and Modelling (ed. R. Earnshaw et al.), Academic Press, 1997, pp. 287–301.
- [9] Т.А Сушкевич. Математические модели переноса излучения. Москва, 2004.
- [10] J.S. Gondek, G.W. Meyer and J.G. Newman. Wavelength Dependent Reflectance Functions // Proceedings of SIGGRAPH'94, 1994, pp. 213–220.
- [11] M. Schramm, J. Gondek and G. Meyer. Light Scattering Simulations using Complex Subsurface Models // Graphics Interface '97, 1997, pp. 56–67.
- [12] J. Dorsey and P. Hanrahan. Modelling and Rendering of Metallic Patinas // Proceedings of SIGGRAPH'96, 1996, pp. 387–396.
- [13] J. Dorsey, A. Edelman, J. Legakis, H. Wann Jensen and H.K. Pedersen. Modelling and Rendering of Weathered Stone // Proceedings of SIGGRAPH'99, 1999, pp. 225–234.
- [14] P. Hanrahan, W. Krueger. Reflection from Layered Surfaces Due to Subsurface Scattering // In SIGGRAPH' 93 Conference Proceedings, California, August 1993, pp. 165-174.
- [15] R.L. Cook, K.E. Torrance. A Reflectance Model for Computer Graphics // ACM Transaction on Graphics, Vol. 1, No. 1, 1982, pp 7-24.
- [16] J.E. Hansen. Radiative Transfer by Doubling Very Thin Layers // Astrophys. J., 155, 1969, pp. 565-574.
- [17] J.E. Hansen and L. Travis. Light Scattering in Planetary Atmospheres // Space Science Reviews, 16, 1974, pp. 527–610.
- [18] M. Pharr and P. Hanrahan. Monte Carlo Evaluation of Non-Linear Scattering Equations for Subsurface Reflection // In Computer Graphics Proceedings, Annual Conference Series, 2000, pp. 75-84.
- [19] 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.
- [20] Boris Barladyan, Lev Shapiro, Alexey Voloboy. Ray maps technique for effective interrogation of results of MCRT simulation // Conference proceedings of 21-th International Conference on Computer Graphics and Vision GraphiCon-2011, Moscow, Russia, pp. 46-49.
- [21] А.Г. Волобой, В.А. Галактионов, С.В. Ершов, А.А. Летунов, И.С. Потемин. Аппаратно-программный комплекс для измерения светорассеивающих свойств поверхностей // «Информационные технологии и вычислительные системы», № 4, 2006, с. 24-39.

## Abstract

The interactive lighting simulation of pearlescent paint basing on paint structure is described. The task of calculation of paint appearance from paint composition was formulated and solved. To provide interactive speed of simulation program the original approaches to calculation and fast rendering algorithms were elaborated. The simulation program structure is described in the paper. The example of program application to the color matching problem of automotive paint is presented.

**Keywords:** *realistic rendering, lighting simulation of paint, physically based lighting, BRDF, optical properties of surface.*

## Authors:

Alexey G. Voloboy, PhD, senior researcher, Keldysh Institute of Applied Mathematics RAS (KIAM RAS). E-mail: [voloboy@gin.keldysh.ru](mailto:voloboy@gin.keldysh.ru)

Sergey V. Ershov, PhD, senior researcher, KIAM RAS. E-mail: [sergey\\_65@mail.ru](mailto:sergey_65@mail.ru)

Sergey G. Pozdnyakov, researcher, KIAM RAS. E-mail: [mephi32@rambler.ru](mailto:mephi32@rambler.ru)

# Спектральная трассировка лучей в задачах моделирования цветового сдвига, возникающего в флюоресцентных средах

Жданов Д.Д.

Государственный Оптический Институт им. С.И. Вавилова, Санкт-Петербург

Галактионов В.А., Ершов С.В., Поздняков С.Г., Потемин И.С.

Институт Прикладной Математики им. М.В.Келдыша РАН, Москва

## Аннотация

Рассмотрены вопросы компьютерного моделирования и расчета распространения света во флюоресцентных рассеивающих средах. Предлагаемое решение обеспечивает эффективную и физически аккуратную модель распространения света во флюоресцентных средах и позволяет осуществлять компьютерное моделирование Стоксова сдвига. Рассматривается пример моделирования цветового сдвига в белом диоде с флюоресцентным наполнителем.

**Ключевые слова:** *рассеивающая среда, флюоресценция, спектральная трассировка лучей, выход флюоресценции, светодиод, Стоксов сдвиг, расчет цвета светодиода.*

## 1. ВВЕДЕНИЕ

Физически корректная модель изображения сложной сцены включает в себя две основные составляющие. Во-первых, это модель физически корректного распространения света от источников до приемников изображения, а во-вторых, это физически корректная модель передачи изображения от приемника излучения до средства его визуализации (фотопленка, экран монитора и т.п.). Данная работа посвящена исключительно первому аспекту, а именно, построению физически корректных изображений сложных сцен, содержащих объекты со специальными свойствами пропускания, отражения, рассеивания, поляризации и флюоресценции. Решение данной задачи требует применения как соответствующих физически аккуратных моделей распространения света, так и соответствующих моделей оптических объектов сцены.

Физически корректные модели не могут базироваться на RGB представлении света и оптических свойств объектов, используемых в большинстве программ компьютерной графики [4, 7]. RGB модель не только ограничена видимой спектральной областью, но и некорректна для моделирования эффектов распространения и рассеивания света, имеющих спектрально-волновую природу. Конечно, для ряда физических эффектов, таких как дифракционное рассеивание и дисперсия, могут быть (в первом приближении) созданы в RGB модели, но применение этих моделей носит очень ограниченный характер. Кроме того эти модели имеют, как правило, «локальное» применение. Например, можно построить корректную RGB модель одной рассеивающей пластины, освещаемой белым светом, однако, использование этой модели как элемента сложной сцены приводит к некорректному результату моделирования. Как правило, искажение цветопередачи наблюдается при построении изображений сложных сцен, в которых возникает многократное переотражение света между ее объектами [1].

Отличие флюоресценции от приведенных выше эффектов заключается в принципиальной невозможности построения RGB модели распространения света во флюоресцентной среде. В случае флюоресценции спектральный диапазон света, выходящего из флюоресцентной среды, может отличаться от спектрального диапазона света, входящего в данную среду [5]. Этот эффект приводит к возникновению цветового сдвига, который, как правило, смещает спектральный диапазон в область больших длин волн. Кроме того, вторичное излучение флюоресцентной среды в видимой области спектра может быть вызвано ультрафиолетовым освещением, которое невозможно представить в RGB пространстве. Поэтому, единственно возможное решение, которое обеспечивает физически корректную передачу цвета, заключается в трассировке спектральных лучей.

Для построения физически корректного изображения используются различные техники трассировки лучей (прямые и обратные, детерминистические и стохастические) и для каждой из них спектральное представление луча и методы его трассировки имеют определенные особенности. Флюоресценция, в данном случае, вносит дополнительную специфику, связанную с переизлучением света в диапазоне, отличном от спектрального диапазона поглощенного излучения. Кроме того, модель флюоресцентного объекта может быть как непосредственной, когда воспроизводится эффект захвата фотонов одной длины волн и их переизлучения на другой длине волн, так и косвенной, когда флюоресцентный объект ассоциируется с двунаправленной функцией подповерхностного рассеивания (измеренной или вычисленной заранее), осуществляющей цветовой сдвиг. Для моделирования эффекта флюоресценции данная работа рассматривает, главным образом, возможность применения прямой спектральной трассировки лучей методом Монте-Карло и построения непосредственной модели переизлучения света внутри флюоресцентной среды.

Наряду с построением физически корректных изображений сцен, содержащих флюоресцентные объекты, разработанная модель может найти применение при проектировании световых устройств, основанных на эффекте флюоресценции. Например, моделей белых светодиодов [8, 9], рассмотренных в главе 4, или концентраторов света солнечных батарей, основанных на эффекте флюоресценции света [11].

## 2. ПРЯМАЯ СПЕКТРАЛЬНАЯ СТОХАСТИЧЕСКАЯ ТРАССИРОВКА ЛУЧЕЙ В СЦЕНАХ, СОДЕРЖАЩИХ ФЛЮОРЕСЦЕНТНЫЕ ОБЪЕКТЫ

Как отмечалось в работе [1], спектральная трассировка луча может выполняться одним из 3 способов. Во-первых, возможно создание ряда «монохроматических» сцен,

оптические свойства которых определены для одной длины волны. Трассировка лучей выполняется независимо в каждой из отдельных «монохроматических» сцен. Результат моделирования для всей спектральной области есть сумма результатов «монохроматического» моделирования для отдельных сцен. Несмотря на свою простоту и привлекательность этот метод неприменим для моделирования эффекта флюоресценции, поскольку он не может обеспечить цветовой сдвиг на монохроматической сцене. Второй способ близок к предыдущему за исключением того, что монохроматические лучи трассируются в сцене, определенной во всем спектральном диапазоне, и каждый из лучей вносит вклад в общий спектральный результат. Данный подход позволяет моделировать эффект флюоресценции, однако эффективность данного решения невысока. Третий способ позволяет трассировать полихроматические лучи, определенные во всем спектральном диапазоне, и поэтому наиболее эффективен для моделирования эффекта цветового сдвига, возникающего во флюоресцентной среде.

За основу спектральной трассировки лучей во флюоресцентных средах был выбран третий подход. В случае прямой трассировки источник света испускает лучи, определенные во всем спектральном диапазоне, и преобразование лучей на оптических объектах сцены аналогично цветному преобразованию лучей в RGB пространстве, когда вместо R, G и B компонент цвета выступает спектральная плотность излучения. В случае попадания луча на флюоресцентный объект, спектральный состав выходного луча претерпевает сдвиг, как правило, в длинноволновую область. Рис. 1 демонстрирует различие во взаимодействии лучей с обычными и флюоресцентными объектами.

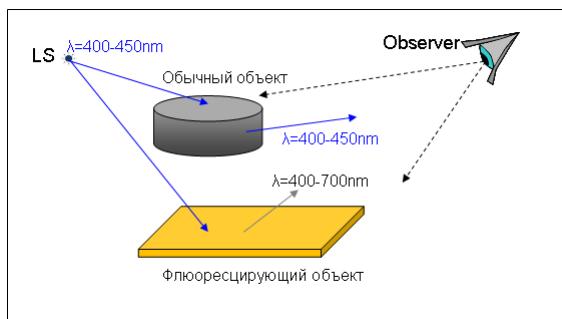


Рис 1: Различие во взаимодействии лучей с обычными и флюоресцентными объектами

С точки зрения макрообъекта сцены флюоресцентный объект определяется двунаправленной функцией флюоресцентного рассеивания (ДФР). В общем виде, специфика данной функции по отношению к обычной двунаправленной функции рассеивания (ДФР) заключается в том, что для каждой длины волны падающего света существует отдельная ДФР, определяющая рассеивание света на заданной длине волны. То есть, свет захватывается на одном спектральном интервале, а затем переизлучается в другом спектральном диапазоне. Рис. 2 демонстрирует данный эффект флюоресцентного рассеивания.

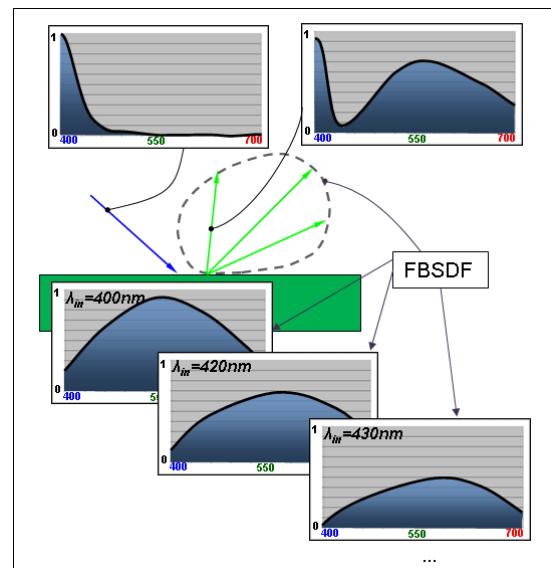


Рис 2: Модель ДФР

Кроме того, специфика рассеивания во флюоресцентной среде заключается в том, что луч должен переносить спектральную мощность луча, а не его спектральную плотность. Эта специфика объясняется природой эффекта флюоресценции, когда происходит захват излучения в одном спектральном интервале и его переизлучение в другом спектральном диапазоне. Спектральная плотность луча  $\varphi(\lambda_k)$  и его спектральная мощность  $P(\lambda_k)$  в интервале  $\Delta\lambda$  связаны простым соотношением:

$$\varphi(\lambda_k) = \frac{P(\lambda_k)}{\Delta\lambda_k} \quad (1)$$

$$\Delta\lambda_k = \frac{(\lambda_{k+1} - \lambda_{k-1})}{2}$$

Мощность рассеянного излучения, возникающего в флюоресцентном объекте, выражается следующим образом:

$$P'(\vec{s}', \lambda) = \sum_{k=1}^N P(\vec{s}, \lambda) FBSDF_{\lambda_k}(\vec{s}, \vec{s}', \lambda) \quad (2)$$

где:

$\vec{s}$  и  $\vec{s}'$  – направления падающего и выходящего излучения соответственно,

$N$  – количество длин волн в спектре,

$FBSDF_{\lambda_k}(\vec{s}, \vec{s}', \lambda)$  – ДФР.

Применительно к сценам, содержащим флюоресцентные объекты, идея прямой стохастической трассировки лучей выглядит следующим образом. Во-первых, выполняется предварительная подготовка сцены к моделированию. Для этого все параметры всех оптических объектов переводятся в единый спектральный диапазон, который охватывает диапазон длин волн первичных источников света и диапазон длин волн, в который происходит вторичное переизлучение флюоресцентных объектов. Только после этого выполняется собственно стохастическая трассировка лучей.

Процесс испускания и трассировки луча состоит из следующих шагов. Сначала, вероятностным образом в соответствии со спектральными мощностями  $M$  источников света выбирается  $i$ -ый источник света, который будет испускать луч.

$$\frac{\sum_{j=1}^{i-1} \sum_{\lambda} P_j(\lambda)}{\sum_{j=0}^M \sum_{\lambda} P_j(\lambda)} \leq \xi < \frac{\sum_{j=1}^i \sum_{\lambda} P_j(\lambda)}{\sum_{j=0}^M \sum_{\lambda} P_j(\lambda)} \quad (3)$$

где:

$\xi$  – случайное число, равномерно распределенное между 0 и 1.

Аналогичным вероятностным образом выбирается точка начала, направление и спектральный состав луча. Затем спектральная энергия луча нормируется к единичной энергии:

$$p(\lambda_k) = \frac{P_i(\lambda_k)}{\sum_{i=1}^N P_i(\lambda_i)} \quad (4)$$

Необходимо отметить, что при отсутствии дисперсии на границе раздела сред, 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. Единственное отличие спектральной трассировки луча в сцене с флюоресцентными объектами это применение ДФФР преобразования лучей (2) вместо ДФР. В случае стохастической трассировки лучей направление флюоресцентного рассеивания выбирается вероятностным образом, аналогично модели ДФР рассеивания. Единственное отличие заключается в том, что флюоресцентное рассеивание в спектральном диапазоне, вызванное освещением в диапазоне  $\Delta\lambda$ , заменяется мощностью спектрального излучения в данном диапазоне. После определения направления рассеивания спектральный состав определяется исходя из спектрального состава луча и ДФФР (2). В финале спектральная мощность луча нормируется к его единичной мощности (4).

Данное решение было реализовано в программном комплексе INSPIRER2 [3] для режимов прямой стохастической трассировки лучей. Это позволило осуществлять физически корректный и эффективный рендеринг сцен, содержащих флюоресцентные объекты.

### 3. АДАПТАЦИЯ МОДЕЛИ ФЛЮОРЕСЦЕНТОЙ СРЕДЫ В СПЕКТРАЛЬНУЮ СТОХАСТИЧЕСКУЮ ТРАССИРОВКУ ЛУЧЕЙ

В общем виде эффект флюоресценции означает преобразование рассеянного света в спектральный диапазон, отличный от диапазона падающего света. В рассматриваемой модели стохастической трассировки лучей эффект флюоресценции означает, что материал захватывает и поглощает электроны одних длин волн и переизлучает поглощенные электроны на других, как правило, более длинных волнах. Поэтому, модель флюоресцентной среды в стохастической трассировке луча состоит из двух основных компонентов.

Во-первых, это поглощение фотона. Для стохастической трассировки лучей вероятность того, что фотон пройдет дистанцию  $L$  без поглощения, определяется следующим образом.

$$\Pr(L) = e^{-\left(\frac{L}{L_{MFP}}\right)} \quad (5)$$

где:

$L_{MFP}$  – средняя длина свободного пробега фотона в среде.

Во-вторых, это переизлучение поглощенного фотона. В данной модели мы не предполагаем наличия каких-либо полей, влияющих на параметры переизлучаемого фотона, поэтому угловое распределение испускаемых фотонов предполагается изотропным, некогерентным и полностью деполяризованным. Оставшимися количественными характеристиками флюоресцентного излучения являются спектральный состав излучения и его энергетический выход. С большой долей вероятности можно утверждать, что спектральный состав излучения не зависит от характеристик (поляризации, длины волны, мощности) падающего излучения. То есть, в случае флюоресцентного переизлучения его спектр постоянен в некотором диапазоне  $\lambda_1 \dots \lambda_N$ . Энергетический выход флюоресцентного материала для света, падающего на длине волны  $\lambda$ , определяется следующим образом:

$$Q(\lambda) = \frac{N_{out}(\lambda_1 \dots \lambda_2)}{N_{in}(\lambda)} \quad (6)$$

где:

$N_{in}(\lambda)$  – число фотонов, поглощенных на длине волны  $\lambda$ .

$N_{out}(\lambda)$  – число фотонов, переизлученных в интервале длин волн  $\lambda_1 \dots \lambda_N$ . Поскольку спектральный состав выходного излучения постоянен, то с вероятностью, пропорциональной выходной спектральной плотности излучения, фотон может приобрести любую длину волны в интервале  $\lambda_1 \dots \lambda_N$ .

Поскольку не вся поглощенная энергия уходит на переизлучение (часть энергии фотонов уходит в тепло), то энергия переизлученных фотонов ниже энергии поглощенных фотонов и, следовательно, длина волны переизлученного фотона выше, чем поглощенного. Это явление называется Стоксовым сдвигом.

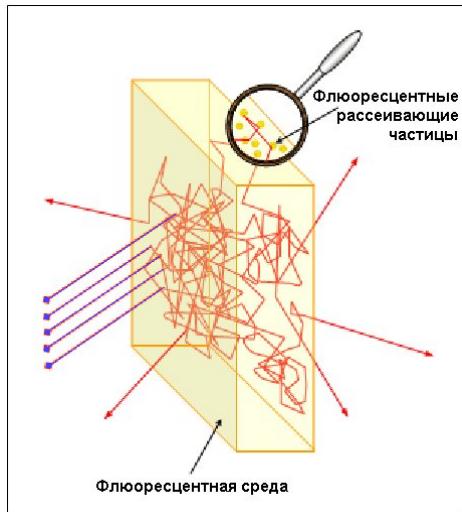
Алгоритмически приведенная выше модель флюоресцентной среды очень близка к модели распространения света в среде с мелкоструктурными рассеивающими частицами. Она состоит из 3 аналогичных шагов:

- Определение длины свободного пробега в флюоресцентной среде (случайная величина, определяемая выражением (5)).
- Перенос луча на дистанцию свободного пробега и определение (вероятностным образом в соответствии с (6)) события поглощения луча.
- Переизлучение луча в новом спектральном диапазоне.

Принимая во внимание близость алгоритмов трассировки луча в среде объемного рассеивания [2] и флюоресцентной среде, оба алгоритма были объединены в один программный интерфейс трассировки луча в среде.

Флюоресцентные материалы, как правило, являются не цельными прозрачными средами, а представляют собой мелкозернистую структуру (частицы фосфора), связанную прозрачной средой. Поэтому, компьютерная модель объемного рассеивания была соответствующим образом расширена. Модель флюоресценции для рассеивающих частиц аналогична модели, рассмотренной выше, и к модели рассеивания частицами добавляется только изотропное флюоресцирующее переизлучение. Как в среде объемного рассеивания, так и во флюоресцентной среде трасса луча представляет собой ломаную линию, где спектральный состав каждого сегмента луча может

отличаться от спектрального состава предыдущих сегментов. Рис. 3 иллюстрирует процесс распространения стохастических лучей в прозрачной среде с рассеивающими флюоресцентными микрочастицами.



**Рис 3:** Распространение стохастических лучей во флюоресцентной среде

Приведенные выше алгоритмы были реализованы в программном комплексе INSPIRER2 [3]. Как было указано в предыдущей главе, специфика модели спектральной трассировки лучей, используемой в программном комплексе INSPIRER2, заключается в том, что фотоны могут переносить не одну длину волны, а целый спектральный диапазон. Поэтому модель поглощения «полихроматических» фотонов была соответствующим образом адаптирована на поглощение всего спектрального диапазона, а модель переизлучения практически сохранилась (фотон несет весь спектр переизлучения, приведенный к единичной мощности). Такое решение наиболее эффективно для задач флюресценции, когда моделирование происходит в очень широком спектральном интервале, включающем в себя как короткие длины волн первичных источников излучения, так и длинные волны, возникающие в результате флюресценции.

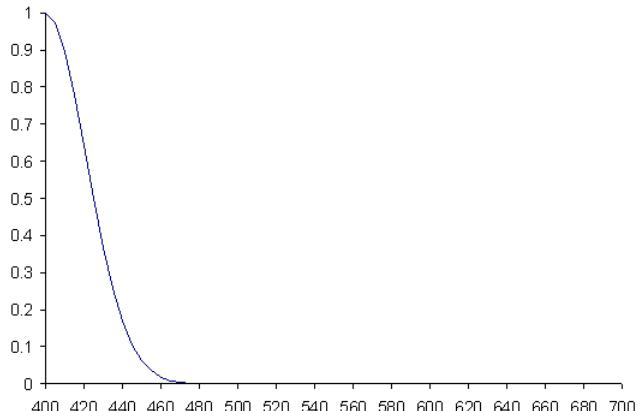
#### 4. ПРИМЕР МОДЕЛИРОВАНИЯ ЦВЕТОВОГО СДВИГА В БЕЛОМ СВЕТОДИОДЕ

В данном разделе приводится пример моделирования белого светодиода, полученного в результате сдвига собственного сине-фиолетового излучения диода в красную область спектра за счет рассеивания света в флюоресцентной среде колпачка диода. Светодиоды это современные, высокоэффективные источники света, которые излучают в очень узком спектральном диапазоне (как правило, в сине-фиолетовой области), что делает такие источники не пригодными во многих областях применения. Задача генерации светодиодного излучения белого цвета очень важна и имеет различные технические решения [8]. Одним из таких решений является генерация белого излучения в результате цветового сдвига излучения, генерируемого обычным синим светодиодом, в красную часть спектра [9]. В этом случае выходное излучение, являющееся результатом смешивания синей и красной зон спектра, становится близким к белому. Целью данного примера является демонстрация результатов компьютерного моделирования эффекта цветового сдвига в изначально синем светодиоде и расчет белого светодиода

посредством оптимального выбора концентрации рассеивающих флюоресцентных частиц.

#### 4.1 Описание модели

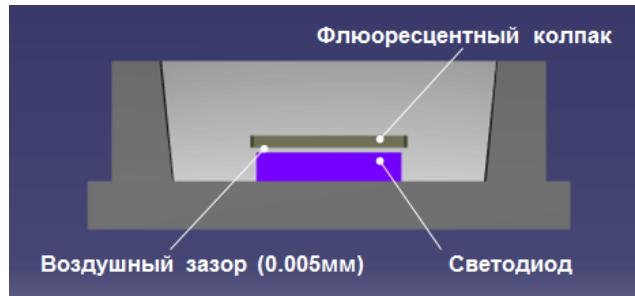
В исследуемом примере флюоресцентный колпак преобразует изначально синее излучение светодиода в выходное излучение белого цвета. Собственная поверхность светодиода моделировалась как квадратный источник света размером 1x1 мм, излучающий по закону Ламберта. Спектрограмма первичного излучения диода представлена на рис. 4.



**Рис 4:** Спектрограмма первичного излучения светодиода

Флюоресцентный колпак, показанный на рис. 5, представляет собой цилиндр высотой 0.25 мм и радиусом 1.25 мм, размещенный над светодиодом. Колпак и светодиод разделены пятимикронным воздушным зазором.

Материал флюоресцентного колпака состоит из «пассивного» связующего вещества (с показателем преломления 1.5) и распределенными в нем сферическими частицами диаметром 15 мкм. Частицы состоят из флюоресцентного вещества с показателем преломления 2.0.



**Рис 5:** Геометрия модели светодиода

Коэффициент энергетического выхода флюоресцентного материала, использованный в данном моделировании, показан на рис. 6.

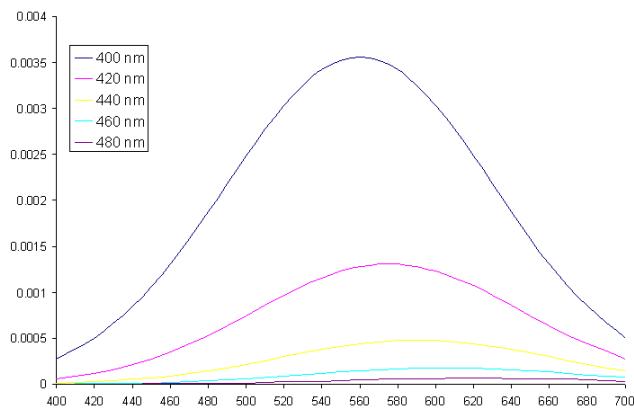


Рис 6: Коэффициент энергетического выхода используемых частиц для диапазона 400-480 нм

Видно, что с увеличением длины волны первичного излучения, показанного на рис. 6 цветными линиями, максимум энергетического выхода смещается в красную область, а его значение уменьшается. Такая функция хотя и является в данном случае искусственной, по свойствам близка к флуоресцентным материалам типа Ce:YAG [5, 6] и удовлетворяет основным физическим свойствам выхода флуоресценции (правилу Стокса).

В данном примере реальное поглощение света отсутствует. При поглощении света частицами он полностью переизлучается в другом спектральном диапазоне. Такая модель является некоторым упрощением, так как обычно поглощенные фотоны полностью не переизлучаются и часть поглощенной энергии идет на нагрев. Такой эффект несколько снижает общую эффективность флуоресцентного колпака при общем сохранении его цвета.

## 4.2 Результаты моделирования

В данном примере были произведены расчеты для различных концентраций частиц. При отсутствии частиц (нулевая концентрация) на рис. 7 мы видим исходный цвет светодиодного излучения (синий). Если концентрация частиц отличается от нулевой, то рассеивание света во флуоресцентном колпаке распределяет его по всей поверхности выходного диска. Рис. 7 демонстрирует результат изменения цвета светодиода в зависимости от концентрации флуоресцентных частиц. Для удобства сравнения цвета, каждое изображение было нормировано к максимальному значению яркости данного изображения. Таким образом, все изображения визуально выглядят с одинаковой максимальной яркостью. Это удобно при исследовании изменения цветового тона, но не показывает насколько эффективно применение колпака с заданной концентрацией флуоресцентных частиц. Реально, яркость уменьшается с ростом концентрации. Это связано как с поглощением света в колпаке и на корпусе диода, так и с более равномерным «размазыванием» света по всей выходной поверхности светодиода.

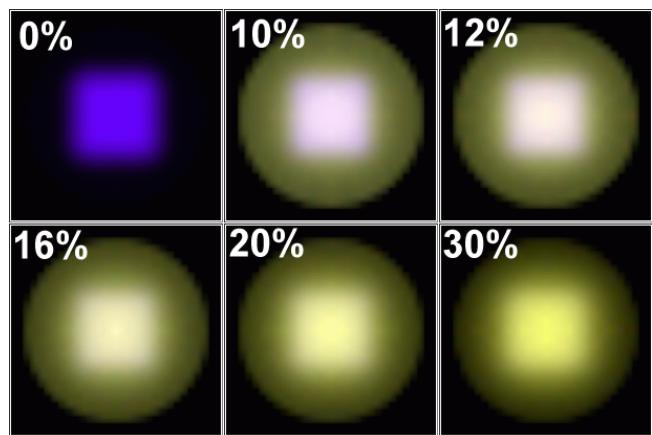


Рис 7: Распределение яркости по верхней поверхности колпака для различных концентраций частиц

Заметим, что при низких плотностях световое пятно на колпаке воспроизводит форму излучающей поверхности светодиода, в то время как при более высоких плотностях частиц, из-за многократного рассеивания света в материале колпака, световое пятно расширяется, достигая симметричного распределения яркости.

## 4.3 Оптимизация белого излучения светодиода

Целью оптимизации является достижение белого цвета излучения светодиода. Для автоматической оптимизации нужен численный критерий. Простым, но вполне приемлемым критерием является отклонение цвета «центральной зоны» (над светодиодом, см. рис.8) от белого. Для вычисления этого критерия мы вычисляем усредненные значения R, G и B компонентов яркости в данной области светодиода и рассчитываем их относительную разность:

$$d = ((R - W)^2 + (G - W)^2 + (B - W)^2)^{1/2} / W, \quad (7)$$

$$W = (R+G+B)/3$$

Затем минимизируем  $d$ , подбирая оптимальные параметры системы. На рис.8 представлен график изменения искомой величины в зависимости от концентрации частиц.

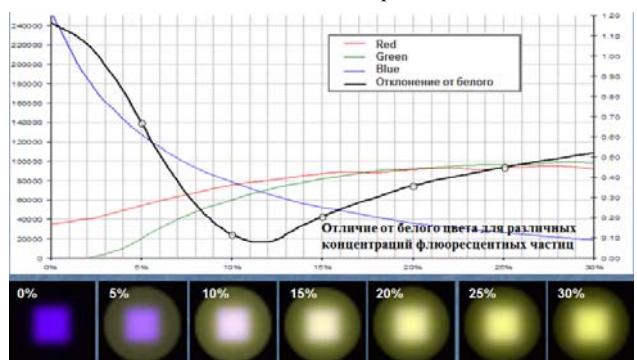


Рис 8: RGB компоненты яркости над колпаком и «отклонение от белого»  $d$  (вторая ось) как функция концентрации

Минимум оценочной функции лежит в области 12%-ной объемной концентрации частиц. Такое значение концентрации дает лучшее приближение цвета к белому.

Значения яркости излучения светодиода очень высоки, что соответствует единичному световому потоку излучения светодиода с поверхности площадью 1  $\text{мм}^2$ .

## 5. ЗАКЛЮЧЕНИЕ

Выработано программное решение для физически корректного моделирования распространения света во флюoresцентных средах, в том числе в средах, содержащих флюoresцентные микрочастицы. Разработанное программное решение было интегрировано в программный комплекс INSPIRER2, что позволило расширить круг задач, решаемых данным программным комплексом, включая задачи моделирования цветового сдвига, возникающего во флюoresцентных средах, в том числе белых светодиодах. Разработанные программные решения были применены для задач моделирования и проектирования белых светодиодов, основанных на эффекте флюoresценции.

## 6. БЛАГОДАРНОСТИ

Работа поддержана грантами РФФИ № 10-01-00302, 11-01-00870, 12-01-00560, а также компанией Integra Inc.

Авторы выражают свою признательность В.Г. Соколову и А.А. Гарбулю за предоставленные примеры.

## 7. Abstract

Computer investigation and design of light propagation in fluorescent scattering media is considered. Suggested solution provides efficient and physically accurate model of light propagation in the media that allows simulating Stokes color shift effect and design of white LED.

**Keywords:** scattering media, fluorescence, spectral raytracing, fluorescence efficiency, LED, Stokes color shift, LED color design.

## 8. ССЫЛКИ

- [1] Б.Х. Барладян, К.А. Востряков, В.А. Галактионов, Д.Д. Жданов, И.С. Потемин, Л.З. Шапиро. Спектральная трассировка лучей в задачах построения фотorealистичных изображений // "Программирование", № 5, 2011, с.13-26.
- [2] А.Г.Волобой, В.А.Галактионов, Д.Д.Жданов. Технология оптических элементов в компьютерном моделировании оптико-электронных приборов // "Информационные технологии в проектировании и производстве", № 3, 2006, с.46-56.

[3] <http://www.integra.jp/en/specter/index.html>

[4] Henrik Wann Jensen, Per Christensen. High quality rendering using ray tracing and photon mapping /SIGGRAPH '07.

[5] S. M. Kaczmarek, G. Domaniak-Dzik, W. Ryba-Romanowski, J. Kisielewski, J. Wojtkowska. Changes in Optical Properties of Ce: YAG Crystals under Annealing and Irradiation Processing // Crystal Research and Technology, Volume 34, Issue 8, September 1999, p.1031-1036.

[6] M. Kirma, A. Lushchikb, Ch. Lushchikb, G. Zimmerer. Investigation of luminescence properties of pure and Ce<sup>3+</sup>

[7] Matt Pharr, Greg Humphreys. Physically Based Rendering - From Theory to Implementation // Morgan Kaufmann, 2004.

[8] Y.Shimizu, Development of White LED light source, in Rare earths, 40, The Rare Earth Society of Japan, Osaka, 2002, p.150-151.

[9] S.Tanabe, S.Fujita, S.Yoshihara, A.Sakamoto, S.Yamamoto. "YAG glass-ceramic phosphor for white LED (II): luminescence characteristics". Proc. of SPIE 5941: 594112 (2005). doi:10.1117/12.614681

[10] A. J. Welch, Craig Gardner, Rebecca Richards-Kortum, Eric Chan, Glen Criswell, Josh Pfefer, Steve Warren. Propagation of fluorescent light // Lasers in Surgery and Medicine, Volume 21, Issue 2, 1997, p.166-178.

[11] M. Bendig, J. Hanika, H. Dammertz, J. C. Goldschmidt, M. Peters, M. Weber. Simulation of Fluorescent Concentrators // Interactive Ray Tracing August 10, 2008, p.93-98.





# Young Scientists School

---

GraphiCon'2012

October 01–05, 2012  
Moscow, Russia

# An evaluation of finger alphabet intelligibility using quality assessment of video with masked content

Petra Heribanova<sup>+</sup>, Jaroslav Polec<sup>\*</sup>, Darina Tarcsiova<sup>^</sup>

<sup>+</sup>Department of Algebra, Geometry and Didactics of Mathematics, FMFI, Comenius University

<sup>\*</sup>Institute of Telecommunications, FEI, Slovak University of Technology

<sup>^</sup>Institute of Special Education Studies, Faculty of Education, Comenius University

Bratislava, Slovakia

petra.heribanova@fmph.uniba.sk, polec@ktl.elf.stuba.sk, darina.tarcsiova@fedu.uniba.sk.

## Abstract

This paper discusses the finger alphabet recognition and evaluating the requirements for image quality and definition of the criteria of automatical, real-time objective evaluation without respondent involvement for speech intelligibility in the video and electronic communications. Tests with respondent were made under the logatom recognizability, as it is the most precise, because we may be able to identify voices or do not know, we can not infer from their context, so it causing people to avoid the tendency to repair improperly admitted syllables. This methodology is based on the intelligibility according to variable transmission channel capacity for different video formats. The aim is to determine video degradation threshold, at which the signs of one handed alphabet are still correctly understood, the degree of degradation of particular alphabet signs and, alternatively, mutual sign exchangeability. The results obtained were applied a standard scale for subjective evaluation of image quality and percentages evaluation of recognizability as used in acoustics. Based on this results of objective evaluation of logatom recognizability with respondent involvement we search the method, which correlate best with intelligibility. The aim of objective methods for evaluation of video quality is to design algorithms whose quality prediction is in good agreement with the results of objective human evaluation and therefore could represent a method for automatic evaluation of video intelligibility with finger alphabet.

**Keywords:** finger alphabet, intelligibility, logatom, video, quality, metrics, geometry.

## 1. INTRODUCTION

Today is not problem make the high definition (quality) video or image. High definition (quality) video requires considerable volume of data that needs to be transferred (and paid). Therefore, we always try to find the best compromise between acceptable video quality and cost. Means of determining the compromise are coding and compression, closely related to the quality evaluation criteria. There are many methods and metrics for objective evaluation of video quality, where the main criterion is "lovely" of video regardless of content. However in the evaluating of video with different methods of implementation of augmentative and alternative communication (AAC) or specific method of communication people with hearing impairments [10] we can not ignore content – intelligibility of video. The main difference between the terms quality and intelligibility is that the term "quality" describes the appearance of decoded video signal ("how" the viewer sees it) and the "intelligibility" is just one aspect of quality saying if the received information gives any sense ("what" the viewer sees in it). High-quality video signal is likely to be intelligible. Conversely, of course it may or may not apply. Anyway, unintelligibility is an indicator of poor quality. In the acoustics,

intelligibility threshold is defined as a point, after which one does hear, but one does not understand [1].

Subjective tests show that sound tends to reduce people's ability to recognize video image degradation. Hearing-impaired people do not rely that much on video quality, as the most important thing to them is whether they are able to understand the meaning. Their subjective video quality evaluation can differ from hearing people. Actually, the biggest difference of video of sign language is its purpose - it is the equivalent of sound channel in normal audiovisual recordings. Our aim is to find criteria for video signal quality encoded in various bit-rates, to achieve full intelligibility of Slovak (or other) sign language and finger alphabet.

There is no recommendation ITU (International Telecommunication Union) for evaluating the quality and intelligibility of the video containing alternative and augmentative means of communication.

Our purpose is to modify the criteria for objective evaluation of quality and create a method for automatic evaluation of speech intelligibility (one - handed finger alphabet) based on [3]. In order to be made automatic objective evaluation of the intelligibility of sign language or finger alphabet, it is necessary to do testing with human factors – respondents. Any such non-automatic testing is not only challenging in terms of time, but also the needs of a large number of respondents. Is it problem. Sign language and finger alphabet is not international, but it is a speech which is divided by nationality and binds to a specific territory, and thus the community of deaf people speak SPJ is size limited. Is there a maximum number of people's of evaluation team, and he is not big or concentrated in one place.

## 2. SIGN LANGUAGE, FINGER ALPHABET

Sign language is the primary communication tool of deaf and/or hard of hearing people. It is visual and spatial language with its own grammar and sign vocabulary. It has visual motor modality and it is independent of spoken language. But it is not international, Slovakia used Slovak sign language. All sign languages used three-dimensional space (the sign space) for communication, which is defined horizontally and vertically. In sign languages, we have two types of components (parameters), which we can be analyzed :

- manual parameters = location, handshape and movement
- non-manual parameters = facial expression, position of eyes, head, upper body, mouth movement

The basic communication element is sign. It is given by configuration (shape and placement) of the hands in sign space, by palm and finger orientation, and also by hand movements themselves. It is quite difficult to learn the sign from books or

static images, because even slight difference in movement and location of the hand can change the meaning. Hence, personal demonstration, or understandable video preview is needed.

Finger alphabet was not created naturally and spontaneously by deaf people. It was adapted from monasteries for the purpose of teaching children with hearing impairments. It is a system of finger and movement configurations that represent letters of the alphabet. The number of finger alphabet signs is related to the number of letters (graphemes) of the language. It is commonly used for purposes of clarification, such as unfamiliar words, names of persons, geographical names, or with words, for whose the asking person does not know the appropriate sign. An advantage of the finger alphabet is that its adoption is not difficult or time-consuming. It helps to express the words in correct grammatical form and thus it is the tool for obtaining a richer vocabulary. In the world, there are two widely used systems of the finger alphabet [10].

- One – handed finger alphabet (Figure 1)
- Two – handed finger alphabet

In some countries, using both (for example Slovakia), in some countries only two-handed (for example UK), or one – handed (for example USA).

In Slovakia, the situation is as follows:

One -handed finger alphabet is used to teach pupils at schools for children with hearing impairment. It is more widespread in the world. On international meetings, the only used finger-spelling alphabet is the one approved by The World Federation of The Deaf.

Two-handed finger alphabet tends to be used by older people, because it is slower. Despite its slowness, it is also used at lectures and seminars because of its better intelligibility and visibility [2].

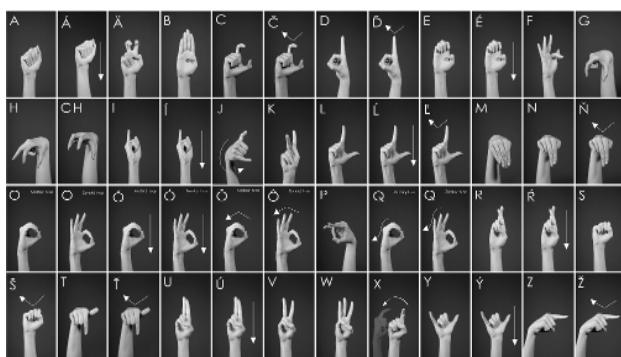


Figure 1: Example of one -handed finger alphabet [2]

### 3. THE INTELLIGIBILITY

In acoustics, the intelligibility of the language (Z) defines the percentage of correctly received elements or parts of speech (a) divided by their total number (b):

$$Z = \frac{a}{b} * 100 \quad (1)$$

We distinguish consonant, logatom, word, and sentence based intelligibility. Logatomes are artificial words designed to look alike words of given language, but they do not have the meaning. The term recognizability is used in recognition of speech sounds (phonemes) and logatomes, as one can either recognize or not recognize them, but there is nothing to be understood [5].

Similarly, we can explore the intelligibility of video: sentence and word intelligibility using sign [7], while logatom and consonant recognizability using the finger alphabet. It is possible to create a sort of "sign logatomes" for the deaf, because one sign in finger alphabet represents a one speech sound in logatom.

### 4. SUBJECTIVE AND OBJECTIVE METHODS FOR THE QUALITY AND INTELLIGIBILITY AND TESTING

The quality evaluation criteria are closely linked with the encoding and compression, as a means for the intended destination boundaries, as it is possible to reduce the size of the data stream. In basically used to evaluate two groups of methods:

1. Subjective methods of measuring reaction observers pursuing a tested system. These methods are very time consuming and to implement [4].

2. Objective methods are automated methods without the participation of observers and the implementation of any distance gap metrics. The aim is to find appropriate application method with the highest correlation with the results of subjective methods [4, 9,11].

Objectively, intelligibility is measured by statistical methods. In the simplest case, it is the percentage of correctly recognized elements. For sentence intelligibility, recognition is considered successful, when the reproduced sentence has correct context and makes sense. Logatom recognizability is expressed as the percentage of correct consonants and vowels from all speech sounds in transmitted logatomes. Resulting from this, it is clear that logatom based recognizability is much more demanding than sentence or word based one, because the meaning cannot be guessed from the context [1].

#### 4.1 Logatom recognizability

To evaluating the requirements for image quality and speech intelligibility in the video we used the logatom recognizability. Logatom (consonant) recognizability was tested using artificial monosyllabic words without meaning, so called logatom. Using logatoms in our tests, to mitigate people's tendency to correct the incorrectly understood consonants or words according to the meaning. We define the criteria for logatom recognizability in one - handed finger alphabet analogically to acoustics. Sign language and finger alphabet has own character and is incompatible with language for hearing people. We use the finger alphabet signs to create so-called "sign logatomes". Every speech sound in logatom is represented by an appropriate sign from Slovak one-handed alphabet. It is a new evaluation methodology of video signal quality in transmissions of sign language in videoconferencing.

This methodology is based on the intelligibility according to variable transmission channel capacity. The aim is to determine video degradation threshold, at which the signs of one handed alphabet are still correctly understood, the degree of degradation of particular alphabet signs and, alternatively, mutual sign exchangeability.

#### 4.2 Testing

Based on this methodology we created the following experiment. We produced 2 video previews with seven different logatoms in Slovak single-handed finger alphabet (one with 41 consonants, one with 42 consonants). The length of the video previews is about one minute. For the whole experiment we used different video formats with 25 frames per second.

Subsequently, these recordings were encoded by the H.264 codec in various bit rates (QP = 30, 40, 50 that corresponds to rates from 390 kbit/s to 4.5 kbit/s respectively). Testing was realized according to subjective ACR method on groups of hearing impaired volunteers. A random sequence of consonants is quite hard to remember; therefore some sequences were shown multiple times to the same people (in different bit-rate and/or video format) without mentioning it in advance.



a)



b)

**Figure 2:** Picture taken from the experiment: a) original (704x576); b) "cif" (352x288) H.264 decoded frame with parameter QP=40

The whole test consists of two parts:

1. Subjective, where the respondent evaluate (by their subjective feelings) the quality and intelligibility of the stream, according to a defined scale.
2. Objective, where the respondent had to rewrite the consonants organized into logatomes to the letters of the Slovak alphabet. While the sentence intelligibility evaluation was based on subjective rating, the logatom recognizability expresses the correctness of all consonants in logatom in percents.

The results obtained were applied a standard scale for subjective evaluation of image quality and percentages evaluation of recognizability as used in acoustics. From the results determine the dependency of recognizability of

transmission rate for different video formats. With decreasing recognizability there was an increasing number of consonant interchanges, mostly between 'a' and 's', 'o' and 'f', and there was also higher frequency of missed or extra added consonants.

The test results confirmed that conventional subjective methods to evaluate video quality with markings 1 (nice) - 5 (ugly) are irrelevant and pointless, because do not say anything about intelligibility. In some case subjective feelings of evaluating respondents were contrary to the results of objective evaluating and the results can not be taken as correct evaluation of intelligibility.

The results of objective evaluation of logatom recognizability are the percentage of correctly received signs from all "sign logatoms" in video stream and this results can be taken as correct evaluation of intelligibility.

Based on this results of objective evaluation of logatom recognizability with respondent involvement we test the method from objective methods [4,6,9,11] which correlate best with intelligibility, and therefore could represent a method for automatic evaluation of video intelligibility with finger alphabet.

## 5. MAIN RESULTS

We work on the basis of full reference method (FR) with differential metrics to evaluate image quality and video between the original and processed video. As the original videos we used the primal videos in format "4-cif" (704x576). The processed video stream in other format "cif" (352x288) and "qcif" (176x144) we resized bilinear interpolation to size of format "4 - cif". Using the software MSU [8] we test the selected metrics, as PSNR, VQM, SSIM, 3SSIM, MSAD and MSE. According to ITU-T recommendations for ratio of objective metrics in regard of the subjective evaluation of a correlation coefficient

$$\rho_{x, y} = \frac{\text{Cov}(X, Y)}{\sigma_x \sigma_y} \quad (2)$$

we find the value of the correlation between existing metrics and intelligibility.



**Figure 3:** Picture taken from the experiment: Region of interest – elliptical mask

Format	QP	Transmission rate [kbit/s]	Logatom recognizability		MSAD	VQM	SSIM	PSNR	MSE
			[%]	MSAD					
<b>4cif</b>	30	199,707	94,84	1,651	1,115	0,956	38,414	12,036	
<b>4cif</b>	40	74,662	73,40	2,671	1,642	0,905	33,764	29,146	
<b>4cif</b>	50	27,172	59,23	4,701	2,589	0,848	29,743	70,780	
<b>Correlation</b>				-0,953	-0,961	0,989	0,997	-0,938	
<b>cif</b>	30	113,205	91,46	2,431	1,380	0,922	34,958	21,889	
<b>cif</b>	40	30,680	73,77	3,528	2,062	0,871	31,497	46,869	
<b>cif</b>	50	11,108	39,39	5,809	3,205	0,835	28,151	101,021	
<b>Correlation</b>				-1,000	-0,999	0,961	0,981	-1,000	
<b>qcif</b>	30	37,923	86,99	3,277	1,866	0,889	32,456	38,191	
<b>qcif</b>	40	12,004	64,01	4,559	2,667	0,849	29,595	72,717	
<b>qcif</b>	50	5,086	0,00	7,353	3,881	0,824	26,220	158,290	
<b>Correlation</b>				-0,999	-0,989	0,921	0,976	-1,000	
<b>Total correlation</b>				-0,959	-0,955	0,829	0,874	-0,982	

Table 1: The results of correlation without mask

Format	QP	Transmission rate [kbit/s]	Logatom recognizability		MSAD	VQM	SSIM	PSNR	MSE
			[%]	MSAD					
<b>4cif</b>	30	199,707	94,840	0,618	0,595	0,987	43,729	5,223	
<b>4cif</b>	40	74,662	73,400	0,915	0,862	0,977	39,504	9,013	
<b>4cif</b>	50	27,172	59,230	1,741	1,503	0,962	34,984	21,682	
<b>Correlation</b>				-0,928	-0,939	0,978	0,991	-0,913	
<b>cif</b>	30	113,205	91,460	0,783	0,685	0,981	40,683	5,983	
<b>cif</b>	40	30,680	73,770	1,153	1,085	0,969	37,382	12,312	
<b>cif</b>	50	11,108	39,390	2,117	1,965	0,958	33,250	31,651	
<b>Correlation</b>				-0,998	-1,000	0,981	0,993	-0,995	
<b>qcif</b>	30	37,923	86,990	1,069	0,986	0,972	37,990	10,935	
<b>qcif</b>	40	12,004	64,010	1,552	1,523	0,963	35,112	20,805	
<b>qcif</b>	50	5,086	0,000	2,729	2,424	0,956	31,166	51,596	
<b>Correlation</b>				-1,000	-0,993	0,936	0,985	-1,000	
<b>Total correlation</b>				-0,970	-0,964	0,850	0,893	-0,986	

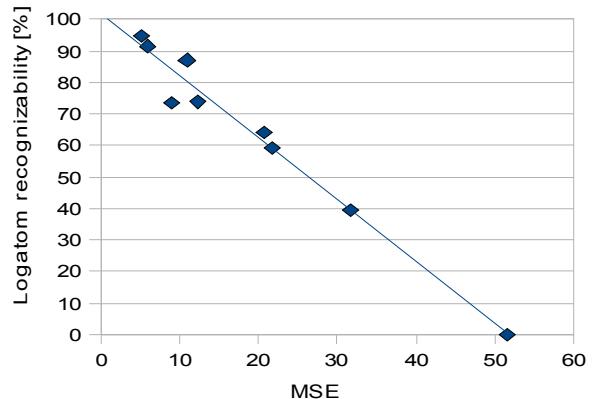
Table 2: The results of correlation with elliptical mask

The results of correlation between logatom recognizability and relevant metrics show Table 1. A comparison between the original and processed video was made for all the pixels with the same value (weight). The best result of correlation for "4cif" format had the metric PSNR, for "cif" format the metrics MSE and MSAD and for "qcif" format had MSE the best value of correlation. The best results of total correlation had the metric MSE with value of correlation coefficient -0,982.

Since the video contains areas that are for us in terms of intelligibility finger alphabet irrelevant, we divided the video into regions of interest according to their importance. Figure 3 shows the main region of interest (ROI) for the recognizability of one-handed alphabet. It is an area in the dominant hand (in this case right) showing the signs.

In some case important region can be a mouth, which are especially important to communicate with lip-reading (cued speech) and be used deaf people, too. This region would be subject to the least degradation of the image in coding and compression. The gray part of image contain background (BG). It is not important for intelligibility finger alphabet as wall, clothes, the rest of the face and hair, or second hand and may be of greater degradation by video processing.

By creating masks, we only tested the intelligibility on ROI. The mask have elliptical shape, which is unvaried during testing. In the experiments were used followed settings: ROI = 1, BG = 0. The results of correlation with the elliptical mask between logatom recognizability and relevant metrics show Tab. 2.



Graph 1: Correlation with mask between MSE and intelligibility

The best result of correlation for "4cif" format had the metric PSNR, for format "cif" the metrics VQM and for "qcif" format had MSE and MSAD the best value of correlation. The best results of total correlation had the metric MSE (Graph 1) with value of correlation coefficient -0,986.

Comparison of the results of correlation with and without the mask is clear to see improvement results using masks for region of interest.

Format	QP	Transmission rate [kbit/s]	Logatom recognizability		MSAD	VQM	SSIM	PSNR	MSE
			[%]	Logatom recognizability					
<b>4cif</b>	30	199,707	94,84	0,301	0,435	0,994	47,061	2,686	
<b>4cif</b>	40	74,662	73,40	0,462	0,663	0,988	42,648	5,113	
<b>4cif</b>	50	27,172	59,23	0,766	1,216	0,981	38,578	11,645	
<b>Correlation</b>				-0,957	-0,939	0,988	0,996	-0,930	
<b>cif</b>	30	113,205	91,46	0,400	0,520	0,990	43,906	3,606	
<b>cif</b>	40	30,680	73,77	0,592	0,859	0,984	38,676	7,467	
<b>cif</b>	50	11,108	39,39	0,971	1,648	0,980	36,817	17,461	
<b>Correlation</b>				-1,000	-0,999	0,961	0,900	-0,998	
<b>qcif</b>	30	37,923	86,99	0,541	0,772	0,986	41,401	6,299	
<b>qcif</b>	40	12,004	64,01	0,778	1,245	0,981	38,448	12,038	
<b>qcif</b>	50	5,086	0,00	1,016	2,056	0,978	34,702	28,405	
<b>Correlation</b>				-0,965	-0,993	0,916	0,981	-1,000	
<b>Total correlation</b>				-0,908	-0,966	0,829	0,863	-0,986	

Table 3: The results of correlation with dynamic mask



Figure 4: Picture taken from the experiment: Dynamic mask within ROI (frame 327)

The next table 3. shows the results of correlation with dynamic mask within ROI. On the video, we applied threshold with value about 130 and subsequently elliptical mask. By creating more accurately masks within region of interest, we made mask for each frame in the video (Figure 4). In the same experiments were used settings: the black part of ROI = 1, the white part of ROI and BG = 0. The best result of correlation for "4 cif" format had the metric PSNR, for format "cif" the metrics MSAD and for "qcif" format had MSE the best value of correlation. The best results of total correlation had the metric MSE with value of correlation coefficient -0,986.

From comparison of the results of correlation for elliptical and dynamic mask is to see improvement results for VQM using dynamic masks for region of interest. The value of MSE remained the same and other results have deteriorated.

## 6. CONCLUSION

This paper describes the technique of evaluating the quality of video signals based on logatom recognizability using so-called sign logatomes, where it is not possible to guess missed consonants from the context and shows our obtained results in one - handed finger alphabet. The results of logatom recognizability was acquired based on objective evaluation of logatom recognizability with respondent involvement.

Therefore the next part of paper describe evaluating the requirements for video quality and definition of the criteria of automatical objective evaluation without respondent involvement for speech intelligibility (finger alphabet for the deaf) in the video and in the electronic communications. Show the result of correlation between existing relevant metrics and logatom recognizability without mask and with two types of mask use to region of interest in video.

## 7. AKNOLEDGMENTS

Research described in the paper was financially supported by the Slovak Research Grant Agency (VEGA) under grant No. 1/0602/11 and by Foundation Tatrabanka under projekt No. 11Sds078 and by the Comenius University under project No. UK/106/2012 the Program for support of young researcher for year 2012.

## 8. REFERENCES

- [1] Granat, M. (2009) *Objective methods for evaluation of audio signal quality* (in Slovak), Brno University of Technology, Brno.
- [2] Hefty, Michal : Finger alphabet (in Slovak). The organization I think - Development of thinking not only for hearing impaired, 2009. [www.zzz.sk](http://www.zzz.sk)
- [3] Heribanová, P., Polec, J., Ondrušová, S., Hosövecký, M.: *Intelligibility of Cued Speech on Video*. In: World Academy of Science, Engineering and Technology. - ISSN 2010-376X. - Iss. 79 (2011), pp. 492-496
- [4] ITU-R Recommendation BT.1683: 2004, *Objective perceptual video quality measurement techniques for standard definition digital broadcast television in the presence of a full reference*.
- [5] Makáň, F.: *Elektroacoustics* (in Slovak), Publisher STU Bratislava, 1995.
- [6] Mardiak, M., Polec, J.: *Novel Method for Objectively Measuring Video Quality*. In: Proceedings ELMAR-2010: 52nd International Symposium ELMAR-2010. Zadar, Croatia, 15.-17.9.2010. - Zadar : Croatian Society Electronics in Marine, 2010. - ISBN 978-953-7044-11-4. - pp. 109-112
- [7] Mordelová, A., Polec, J., Ondrušová, S., Filanová, J. (2010) New Objective Method of Evaluation Cued Speech Recognition in Videoconferences, Proceedings Redžúr 2010, Bratislava, STU v Bratislavie FEI. 4 p., CD-Rom.

- [8] MSU *Video Quality Measurement Tool*. MSU Graphics & Media Lab (Video Group) Moskva, 2008. [Online] [Dátum: 5.2.2011]  
[http://compression.ru/video/quality\\_measure/video\\_measureme nt\\_tool\\_en.html](http://compression.ru/video/quality_measure/video_measureme nt_tool_en.html)
- [9] Ries, M. et al.: *Video quality estimation for mobile H.264/AVC video streaming*. In: Journal of Communications, vol.3, 2008, no.1, pp. 41-50
- [10] Tarciová, D.: *The communication system for deaf and ways to overcome their communication barriers. (In Slovak) Sapientia: Bratislava, 2005. ISBN 80-69112-7-9*
- [11] Winkler, S.: *Digital video quality vision model and metrics*. 1. vyd. Chichester : John Wiley & Sons Ltd., 2005. ISBN 0-470-02404-6

### About the author

P. Heribanová was born in 1986 in Kremnica, Slovak Republic. She received M.Sc. degree in Geometry from the Faculty of Mathematics, Physics and Informatics, Comenius University in Bratislava in 2010. She is a PhD. student of Geometry and Topology at the same university. Her research interests include image coding, reconstruction and quality evaluation. Department of Algebra, Geometry and Didactics of Mathematics, FMFI, Comenius University. Her contact email is [petra.heribanova@fmph.uniba.sk](mailto:petra.heribanova@fmph.uniba.sk).

J. Polec was born in 1964 in Trstená, Slovak Republic. He received the M.Sc. and PhD. degrees in telecommunication engineering from the Faculty of Electrical and Information Technology, Slovak University of Technology in 1987 and 1994, respectively. From 2007 he is professor at Department of Telecommunications of the Faculty of Electrical and Information Technology, Slovak University of Technology and at Department of Applied Informatics of Faculty of Mathematics, Physics and Informatics of Comenius University. His research interests include Automatic-Repeat-Request (ARQ), channel modeling, image coding, reconstruction and filtering. Institute of Telecommunications, FEI, Slovak University of Technology. His contact email is [polec@ktl.elf.stuba.sk](mailto:polec@ktl.elf.stuba.sk).

D. Tarciová was born in 1963 in Levoča, Slovak republic. She received the M.Sc. and PhD. degrees in Special Education from the Faculty of Education, Comenius University. She is professor at Institute of Special Education Studies of the Faculty of Education, Comenius University. His research interests include special education for deaf people (sign language, finger alphabets, and specific method of education deaf and hard of hearing). Her contact email is [darina.tarciová@fedu.uniba.sk](mailto:darina.tarciová@fedu.uniba.sk).

# Выделение объектов в видеопотоке на основе разрезов графов

Антон Конушин, Никита Четвериков

Лаборатория Компьютерной Графики и Мультимедиа  
Московский Государственный Университет, Москва, Россия  
[{ktosh,nchetverikov}@graphics.cs.msu.ru](mailto:{ktosh,nchetverikov}@graphics.cs.msu.ru)

## Аннотация

Алгоритмы поиска объектов переднего плана широко используются в видеонаблюдении. Однако большинство существующих алгоритмов работает либо быстро, но неточно, либо точно, но медленно. В данной статье предлагается новый алгоритм выделения объектов в видеопотоке, основанный на «прицеливании», т.е. предварительном выборе областей интереса [5]. К выделенным областям интереса затем применяется качественный, но медленный алгоритм на основе минимизации функционала энергии с помощью разрезов графов [1]. Такой подход позволяет достичь большей скорости по сравнению с применением алгоритма разрезов графов ко всему кадру, но сохранить точность определения границ объекта интереса.

**Ключевые слова:** поиск объектов в видео, трекинг, разрезы графов.

## 1. ВВЕДЕНИЕ

В статье рассматривается задача выделения объектов переднего плана в видеопоследовательности, полученной со статичной камеры. Основным применением алгоритмов выделения переднего плана являются системы видеонаблюдения со стационарными камерами. В таких системах наблюдаемая сцена мало меняется от кадра к кадру в видеопотоке, поэтому называется «фоном». Все новые объекты, попадающие в области видимости, отличаются от фона и называются «передним планом». Формально задача выделения объектов переднего плана в видеопотоке ставится как определение для каждого пикселя каждого кадра, принадлежит ли он фону, либо объекту на переднем плане (см. рис. 1).

Для последующего анализа выделенных объектов необходимо как можно точнее определить их границы. Если в выделенные области объектов попадает фон, то анализ таких объектов, например, поиск похожих в видеоархиве, затруднен. Другой проблемой является разделение одного наблюдаемого объекта на несколько несвязанных областей. Наиболее высокую точность выделения объектов показывают методы на основе минимизации функционала энергии специально вида, заданного на всем изображении. Однако даже вычислительно эффективные методы минимизации энергии, подходящие для решения только для случая бинарных переменных, такие как методы на основе разрезов графов, оказываются недостаточно быстрыми для обработки видеопотока высокого разрешения в реальном времени.

В данной работе предлагается применить идею «прицеливания» [5] для выбора областей интереса, и применять методы на основе минимизации функционала энергии только к ним. За счёт этого удается обеспечить высокую точность границ объектов при высокой скорости работы алгоритма.



**Рис. 1:** Задача выделения объектов в видеопоследовательностях. Сверху: исходный кадр. Снизу: обработанный кадр (пиксели, отнесённые к фону, затемнены, к объекту переднего плана – «подсвечены»).

## 2. СУЩЕСТВУЮЩИЕ МЕТОДЫ

Основную часть существующих методов выделения объектов переднего плана в видеопотоке можно отнести к трём основным категориям:

- Попиксельные методы [2] [3] [9];
- Поблочные методы [11], [13];
- Методы, основанные на минимизации функционала энергии по всему изображению [7].

Попиксельные методы обрабатывают все точки изображения независимо. Обычно вначале они строят цветовую модель фона и во время работы оценивают, насколько текущий цвет пикселя ей соответствует. В качестве модели фона может использоваться, к примеру, нормальное распределение либо смесь из статически [2] либо динамически [9] заданного числа нормальных распределений. К этой же группе можно отнести методы, осуществляющие попиксельную классификацию, но использующие при этом статистику распределения цветов по окрестности пикселя [3]. Подобные методы обычно обладают достаточно высокой скоростью работы, однако

чувствительны к шуму и не учитывают связанность разметки (среди большинства пар соседних пикселей либо оба принадлежат фону, либо переднему плану).

Методы из второй категории обрабатывают независимо не отдельные пиксели, а группы пикселей, объединенные в блоки. Несмотря на то, что использование таких методов позволило в некоторых случаях достичь высокой скорости работы [11], подобный подход даже теоретически не позволяет добиться идеальной точности, т.к. к фону либо переднему плану относится сразу весь блок пикселей. Если блок содержит границу, то в таких областях граница объекта определяется заведомо неточно.

Методы на основе минимизации энергии функционала специального вида позволяют использовать информацию со всего изображения в совокупности, включая информацию о градиенте яркости между соседними пикселями. Такой подход позволяет учесть, например, тот факт, что граница между объектом чаще разделяет пиксели, сильно отличающиеся по цветам, чем похожие по цвету. За счёт учета пикселов в совокупности методы оказываются менее чувствительными к шуму в изображении, чем попиксельные методы, и дают большую точность, чем поблочные методы. Недостатком методов на основе функционала энергии является низкая скорость работы, определяемая низкой скоростью методов минимизации энергии.

Одним из наиболее известных методов, использующих для поиска объектов в видеопотоке минимизацию энергии по всем пикселям изображения, является метод *BackgroundCut* [7]. Он учитывает цвет пикселя и его контраст с «соседями», а также принимает во внимание возможные изменения фона: изменение освещения, движение в фоне, появление новых объектов на заднем плане и дрожание камеры. Однако скорость его работы составляет 12-15 кадров в секунду для разрешения 320x240 (при обработке на процессоре 3.2 GHz) и является недостаточной для многих задач видеонаблюдения.

Отдельно стоит отметить различные методы ускорения и улучшения качества работы алгоритмов. Одним из распространённых методов увеличения скорости является использование многомасштабных представлений [10], когда обрабатывается не само исходное изображение, а его уменьшенные в несколько раз копии. Ряд методов учитывает специфику предметной области для добавления в используемую модель дополнительных условий. Например, при решении задачи отслеживания лабораторной мыши может быть учтена форма тела животного [12]. Другим примером является использование информации о наиболее вероятном положении камеры и типичном виде отслеживаемых объектов: если камера находится на высоте 1-2 метра над землёй и направлена параллельно ей, то подходящие к ней спереди люди занимают, как правило, несколько непрерывных соседних столбцов точек на кадре [4].

### 3. ПРЕДЛОЖЕННЫЙ АЛГОРИТМ

Основная идея предлагаемого алгоритма заключается в том, чтобы применять метод выделения объектов на основе минимизации функционала энергии не ко всему изображению в целом, а только к некоторым областям. Для этого вначале необходимо выбрать области интереса, которые потенциально содержат объекты переднего плана, т.е. «прицелить» алгоритм. Как и другие алгоритмы выделения объектов на основе вычитания фона, предлагаемый алгоритм состоит из двух этапов: построения модели фона и выделения объектов переднего плана.

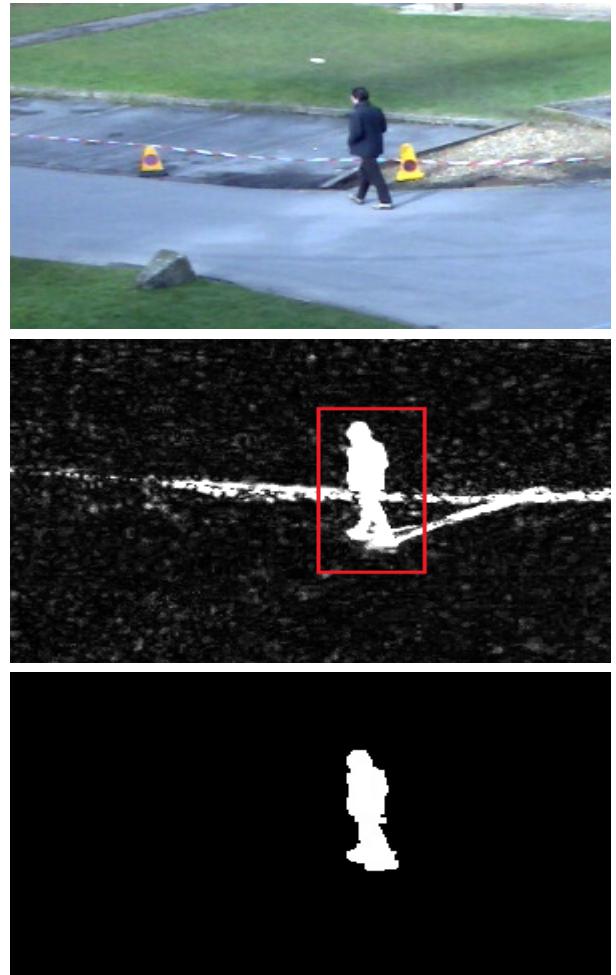


Рис. 2: Основные шаги алгоритма. Сверху вниз: исходный кадр, карта вероятностей с выбранной областью интереса, результат.

#### 3.1 Модель фона

Здесь и далее предполагается, что цвет каждой точки изображения задаётся в формате RGB, то есть вектором из трёх значений (по одному для каждой составляющей цвета).

Предлагаемый алгоритм использует достаточно простую модель фона. Предполагается, что цвет  $c_{xy}$  в каждой отдельно взятой точке фона  $(x, y)$  определяется нормально распределённой случайной величиной  $c_{xy} \sim N(\mu_{xy}, \Sigma)$ . Отметим, что векторы математических ожиданий задаются в каждой точке отдельно, в то время как матрица ковариации считается одинаковой для всех точек. Такой выбор обусловлен предположением о том, что цвет каждой точки фона достаточно стабилен, и его изменения на разных кадрах вызваны в основном шумом, одинаковым для всех точек изображения.

#### 3.2 Построение модели фона

На этапе обучения алгоритм получает на вход неупорядоченный набор кадров, содержащих изображение фона без каких-либо посторонних объектов. Допускается различное освещение сцены на разных кадрах, а также незначительные изменения фона (например, колебания лёгких предметов из-за ветра).

По полученным данным производится расчёт средних значений цвета в каждой точке, а также матрицы ковариации, единой для всех точек. Кроме того, для каждого пикселя сохраняется последовательность значений

цветового вектора, которая впоследствии используется при динамическом обновлении модели фона.

### 3.3 Поиск объектов в видеопотоке

На данном этапе алгоритм последовательно получает и обрабатывает кадры видеопотока, которые могут содержать либо не содержать объекты переднего плана. Работа с кадром состоит из следующих основных шагов (рис. 2):

1. Для кадра по модели фона рассчитывается карта вероятностей, то есть матрица, размеры которой совпадают с разрешением кадра, и значение каждого элемента равно вероятности того, что соответствующая точка кадра принадлежит фону;
2. По полученной карте производится поиск областей, содержащих достаточно большое число точек, не принадлежащих фону, т.е. проводится «прицеливание»;
3. Задача поиска объекта в каждой из найденных областей сводится к задаче минимизации энергии определённого функционала, которая, в свою очередь, сводится к поиску минимального разреза некоторого графа;
4. Производится обновление модели фона с использованием информации о том, какие части кадра были отнесены к переднему плану, а какие – к фону.

Так как шаг 1 достаточно очевиден, рассмотрим более подробно лишь шаги 2-4.



Рис 3: Регион с объектом, найденный алгоритмом «прицеливания» (слева), и объект после уточнения границ (справа).

#### 3.3.1 Поиск областей интереса

Для быстрого поиска регионов изображения, содержащих объекты переднего плана, используется метод «скользящего окна». В ходе его работы по карте вероятностей  $P$  с некоторым шагом перемещается окно переменного размера, и для каждого его положения и размера вычисляется функция

$$f(x, y, w, h) = \frac{\sum_{i=x}^{x+w} \sum_{j=y}^{y+h} P_{ij}}{w * h}$$

где

- $(x, y)$  – координаты верхнего левого угла окна;
- $w$  и  $h$  – ширина и высота окна.

Если значение  $f$  для какого-либо положения и размера окна ниже заданного порога, это окно считается содержащим объект переднего плана. Если два или более таких окон пересекаются либо расстояние между ними меньше заданного порога, для дальнейшей работы выбирается ограничивающий прямоугольник их совокупности.

Для ускорения работы этой части алгоритма используется представление карты вероятностей в виде интегрального изображения (такой подход применяется, например, в алгоритме Виолы-Джонса [8]).

#### 3.3.2 Уточнение границ объектов

Для уточнения границ объектов, найденных на предыдущем этапе, используются метод минимизации функционала энергии специального вида.

Каждому пикслю изображения с координатами  $(i, j)$  сопоставляется случайная переменная  $x_{ij}$ . Кроме того, вводится множество меток  $L = \{0, 1\}$ ; метка 0 отождествляется с фоном, 1 – с объектом переднего плана. Задача заключается в поиске разметки, минимизирующую энергию вида

$$E(x) = \sum_{i=0}^W \sum_{j=0}^H \varphi_{ij}(x_{ij}) + \sum_{i=0}^W \sum_{j=0}^H \sum_{(i', j') \in N(i, j)} \psi_{ij i' j'}(x_{ij}, x_{i' j'})$$

где

- $W, H$  – ширина и высота изображения;
- $N(i, j)$  – множество таких координат  $(i', j')$ , что точки  $(i, j)$  и  $(i', j')$  являются соседними;
- $\varphi_{ij}(x_{ij})$  – унарные потенциалы, отвечающие за «стоимость» назначения переменной определённой метки;
- $\psi_{ij i' j'}(x_{ij}, x_{i' j'})$  – парные потенциалы, соответствующие тому факту, что у соседних пикселей, скорее всего, одинаковые или похожие метки.

В предлагаемом алгоритме унарные потенциалы получаются из соответствующих значений карты вероятности вычитанием последних из 1 (так как каждая точка обязательно принадлежит либо объекту, либо фону, то сумма соответствующих вероятностей равна 1).

Для задания парных потенциалов используется модель Поттса:

$$\psi_{ij i' j'} = \begin{cases} 0 & \text{если } x_{ij} = x_{i' j'} \\ w * \text{abs}(p_{ij} - p_{i' j'}) & \text{если } x_{ij} \neq x_{i' j'} \end{cases}$$

где  $w$  – параметр.

Для поиска минимума энергии используется алгоритм Бойкова-Колмогорова на основе поиска разреза минимальной стоимости [12].

На рис. 3 показан пример результата работы алгоритма уточнения границ объекта.

#### 3.3.3 Обновление модели фона

После окончательного определения границ всех найденных объектов производится обновление модели фона. При этом для каждого пикселя, отнесённого к фону, удаляется самое «старое» значение из истории его значений, в конец её добавляется значение из текущего кадра, после чего производится пересчёт среднего значения цвета в данной точке. Такой подход позволяет динамически адаптировать модель к достаточно плавным изменениям фона, например, изменению уровня освещения.

### 4. ЭКСПЕРИМЕНТЫ И СРАВНЕНИЕ

Предложенный алгоритм был протестирован на нескольких видеопоследовательностях, взятых из базы PETs 2010 [6], содержащей набор городских сцен с пешеходами. Для проведения численной оценки одна из видеопоследовательностей была попиксельно размечена вручную для оценки точности работы алгоритма.

В ходе тестирования производилось сравнение описанного алгоритма с каждым из базовых для него методов: отдельно взятым алгоритмом поиска области интереса (найденным объектом считались точки  $(x, y)$  из области интереса, для

которых  $P_{xy} < \frac{1}{2}$ ) и алгоритмом, использующим минимизацию функционала энергии по всему изображению. Оценивалась скорость работы, а также точность и полнота нахождения объектов переднего плана. Результаты для одной из последовательностей можно увидеть в таблице 1.

	Точность	Полнота	F-оценка	Время (мсек/кадр)
Предложенный алгоритм	0.488228	0.835179	0.965023	112
Марковские поля для всего изображения	0.275857	0.838707	0.863122	281
Алгоритм поиска области интереса	0.329894	0.77707	0.914774	110

**Таблица 1:** Результаты тестирования

Нужно сразу отметить, что основной целью тестирования было не получение как можно более высокой точности результатов или скорости работы, а сравнение предложенного алгоритма с каждым из двух алгоритмов, лежащих в его основе. Как следствие, результаты могут быть улучшены как по качеству (более тонкой настройкой параметров алгоритмов или использованием другой модели фона), так и по скорости (например, распараллеливанием отдельных частей алгоритма).

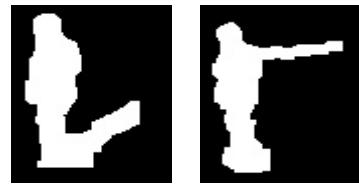
Основным достижением предложенного алгоритма, как можно увидеть из результатов тестирования, является скорость работы, эквивалентная скорости работы алгоритма поиска областей интереса, при качестве результатов, достигающем и даже превосходящем качество результатов метода, использующего случайные Марковские поля.

Отдельно стоит пояснить относительно невысокую точность результатов. Одна из причин заключается в том, что алгоритм часто не отделяет движущегося человека от его тени и отмечает их как один объект. Так как тень занимает на изображении площадь, сравнимую с площадью самого человека, то точность существенно падает. Другой распространённой ошибкой является попадание в область интереса посторонних предметов, например ленты, колеблющейся на ветру (см. рис. 4). Этой же причиной обусловлена ещё более низкая точность работы алгоритма, использующего минимизацию энергии для всего изображения: так как областью интереса фактически является всё изображение, то к переднему плану относится больше посторонних объектов.

Алгоритм был реализован на C++; тестирование производилось на компьютере с процессором Core 2 Duo 2.0 GHz. Разрешение всех кадров, использованных для тестирования – 768x576. Построение модели фона по 50 кадрам занимает около 3 секунд.

## 5. ЗАКЛЮЧЕНИЕ

В статье предложен новый алгоритм выделения объектов переднего плана, основанный на вычислении пописельной карты вероятности принадлежности пикселя фону, выборе областей интереса с помощью скользящего окна по этой карте, и применению метода на основе минимизации функционала энергии специального вида к выбранным областям.



**Рис. 4:** Примеры ошибок алгоритма: тень, отнесённая к переднему плану (слева); часть ленты, попавшая в область интереса и также отнесённая к найденному объекту.

## 6. БЛАГОДАРНОСТИ

Работа выполнена при поддержке гранта Президента Российской Федерации для молодых ученых - кандидатов наук МК-4644.2012.9 и проекта МРЛ-2010-050 с компанией Microsoft Research.

## 7. ССЫЛКИ

- [1] Y. Boykov, V. Kolmogorov, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy-BK Minimization in Vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004.
- [2] P. KaewTraKulPong, R. Bowden, "An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection," *Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems*, 2001.
- [3] T. Ko, S. Soatto, D. Estrin, "Background Subtraction on Distributions," *ECCV '08 Proceedings of the 10th European Conference on Computer Vision: Part III*, 2008, pp. 276-289..
- [4] V. Konushin, A. Konushin, "Improvement of background subtraction by mask constraints," *Proc. GraphiCon*, 2010, pp. 96-99.
- [5] I. Laptev, P. Pérez, "Retrieving actions in movies," in *Proc. ICCV'07*, 2007..
- [6] PETS 2010. <http://www.cvg.rdg.ac.uk/PETS2010/a.html#s0>.
- [7] J. Sun, W. Zhang, X. Tang, H.-Y. Shum, "Background Cut," *ECCV Proceedings*, Berlin, 2006, pp. 628–641.
- [8] P Viola, M Jones, "Rapid object detection using a boosted cascade of simple features," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, pp. I-511-I-518.
- [9] Z. Zivkovic, "Improved Adaptive Gaussian Mixture Model for Background Subtraction," *ICPR '04 Proceedings of the Pattern Recognition*, 2004, pp. 28-31
- [10] Н.В. Птицын, "Встроенная видеоаналитика для детектирования и сопровождения объектов при помощи многомасштабных признаков," *Proc. GraphiCon*, 2010, pp. 200-205.
- [11] Р. Салимзабаров, А. Разумков, А. Коробков, "Адаптивный метод выделения движущихся объектов," *Proc. GraphiCon*, 2010, pp. 333-334.
- [12] М. Синдеев, А. Конушин, "Отслеживание контура лабораторной мыши в видеопоследовательности на основе метода ветвей и границ," *GraphiCon*, 2010, pp. 214-217.
- [13] Е. Шальнов, В. Кононов, В. Конушин, "Алгоритм вычитания фона, основанный на поблочных классификаторах," *Proc. GraphiCon*, 2011, pp. 227-230

## Критерий перезахвата на ложный объект при корреляционном методе сопровождения

Ефимов Андрей Сергеевич, Козлитин Иван Алексеевич  
ЗАО «Научно-технический центр «ЭЛИНС», Зеленоград, Россия  
dvveller@gmail.com, kozlitin@elins.ru

### Аннотация

Для корреляционного алгоритма сопровождения предложен метод обнаружения перезахвата на ложный объект. Метод основан на анализе зависимости коэффициента корреляции объекта и эталона от номера кадра при сопровождении. Суть метода состоит в обнаружении на этой зависимости участков особого вида, соответствующих перезахвату.

**Ключевые слова:** Сопровождение, Обнаружение, Корреляция, Критерий срыва сопровождения.

### 1. ВВЕДЕНИЕ

Для любого метода слежения за объектом практически важно иметь критерий срыва сопровождения. В случае обнаружения срыва сопровождения можно переключиться на другой метод или перейти к экстраполяции.

Выделим две основные причины срыва сопровождения:

1. Исчезновение объекта из поля зрения, либо падение его контраста ниже некоторого критического уровня;
2. Перезахват на ложный объект.

При срыва сопровождения первого типа факт срыва может быть обнаружен путем сопоставления изображения в стробе с аддитивной моделью фона. При срыва второго типа такой способ работает плохо из-за того, что перезахват обычно происходит на сложном фоне, адекватную модель которого построить затруднительно.

В случае, когда метод сопровождения предусматривает отделение объекта от фона, решение о срыва сопровождения принимается исходя из результата сопоставления области изображения, предположительно принадлежащей объекту, с аддитивной моделью объекта, построенной ранее. Если же метод сопровождения не предусматривает такого разделения фона и объекта, то такой подход непосредственно неприменим.

Практически важным примером подобного алгоритма является корреляционный метод сопровождения с обновляемым эталоном. Единственной непосредственно доступной характеристикой качества слежения для этого алгоритма является коэффициент корреляции объекта и эталона, известный для каждого кадра видеопоследовательности. Существенно, что вычисление этого коэффициента не требует никаких дополнительных вычислительных операций.

В данной работе путем анализа временной зависимости коэффициента корреляции от времени удалось выработать критерий перезахвата на ложный объект.

### 2. КОРРЕЛЯЦИОННОЕ СОПРОВОЖДЕНИЕ

Суть этого метода состоит в поиске на изображении участка наиболее коррелированного с эталоном, создаваемом при захвате объекта. Поскольку объект обычно меняется со временем, то и эталон также должен постепенно меняться, подстраиваясь под объект. Обычно эталон обновляется по следующему правилу:

$$E_n = (1 - \alpha)E_{n-1} + \alpha I_n, \quad (1)$$

где  $E_n$  - эталон на  $n$ -ом шаге,  $I_n$  - изображение объекта на  $n$ -ом шаге и  $\alpha$  - коэффициент обновления эталона. Значение  $\alpha$  подбирается исходя из скорости изменения объекта, за которым осуществляется слежение, и находится в диапазоне от 0 до 0.1.

### 3. АНАЛИЗ ЗАВИСИМОСТИ КОЭФФИЦИЕНТА КОРРЕЛЯЦИИ ОТ ВРЕМЕНИ

Типичная зависимость коэффициента корреляции от времени приведена на рис. 1. В процессе нормального сопровождения коэффициент корреляции  $\rho$  меняется достаточно случайно. В момент времени  $t_0$  происходит перезахват. На графике наблюдается резкий спад коэффициента корреляции, за которым следует нарастание этого коэффициента по кривой определенной формы. Начальный участок зашумлен, но затем  $\rho$  начинает возрастать плавно. Скорость нарастания зависит от  $\alpha$ .

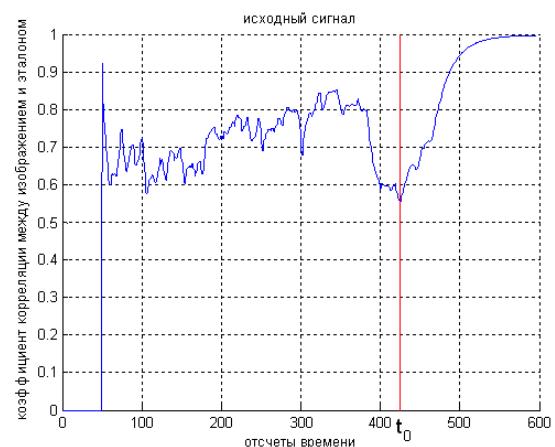


Рис. 1 Зависимость коэффициента корреляции от времени. Красной чертой показан момент времени, в который происходит перезахват.

Зависимость коэффициента корреляции от времени после перезахвата можно определить аналитически. Пусть  $E_0$  - эталон в момент перезахвата,  $I$  - часть изображения, на которую произошел перезахват, тогда  $I_n = I$  для всех  $n$ .

Для этого частного случая равенство (1) можно записать следующим образом:

$$\mathbf{E}_n = (\mathbf{E}_0 - \mathbf{I})(1-\alpha)^n + \mathbf{I}. \quad (2)$$

Можно показать, что в этом случае коэффициент корреляции  $\mathbf{E}_n$  и  $\mathbf{I}$  вычисляется по формуле

$$\rho(\mathbf{E}_n, \mathbf{I}) = \rho((\mathbf{E}_0 - \mathbf{I})(1-\alpha)^n + \mathbf{I}, \mathbf{I}) =$$

$$= \left\{ 1 + \frac{(1-\rho^2(\mathbf{E}_0, \mathbf{I}))(1-\alpha)^{2n}}{\left( \rho(\mathbf{E}_0, \mathbf{I})(1-\alpha)^n + \frac{\sigma_I}{\sigma_{E_0}}(1-(1-\alpha)^n) \right)^2} \right\}^{-1/2}, \quad (3)$$

где  $\sigma_I$  - стандартное отклонение ложного объекта,  $\sigma_{E_0}$  - стандартное отклонение эталона на  $n$ -ном кадре.

В реальности на изображении присутствует шум, поэтому более правильно писать  $\mathbf{I}_n = \mathbf{I} + \mathbf{\Phi}_n$ , где  $\mathbf{\Phi}_n$  - добавка, обусловленная шумом. Будем считать, что шум некоррелирован с изображением и шумовые добавки к разным кадрам некоррелированы между собой. Тогда с учетом шума формула (3) записывается следующим образом:

$$\rho_n = \rho(\mathbf{E}_n, \mathbf{I}_n) =$$

$$= \rho_\infty \left\{ 1 + \frac{(1 - (\rho_0 / \rho_\infty)^2)(1-\alpha)^{2n}}{\left( (\rho_0 / \rho_\infty)(1-\alpha)^n + k(1-(1-\alpha)^n) \right)^2} \right\}^{-1/2}, \quad (4)$$

где  $k = \sigma_I / \sigma_{E_0}$ ,  $\rho_0 = \rho(\mathbf{E}_0, \mathbf{I}_0)$ , а  $\rho_\infty$  - масштабный коэффициент, имеющий смысл коэффициента корреляции при  $n \rightarrow \infty$ . Вследствие шума всегда  $\rho_\infty < 1$ .

#### 4. ОБНАРУЖЕНИЕ ПЕРЕЗАХВАТА

Задача обнаружения перезахвата сводится к поиску на графике коэффициента корреляции от времени зависимости вида (4). Эту задачу можно свести к проблеме обнаружения на графике участка прямой с угловым коэффициентом  $-\ln(1-\alpha)$ . Для этого необходимо разрешить (4) относительно  $(1-\alpha)^{-n}$  и прологарифмировать обе части полученного равенства. В итоге получим

$$-n \ln(1-\alpha) = \ln \left( 1 + \frac{1}{k} \left( \frac{1 - (\rho_0 / \rho_\infty)^2}{(\rho_\infty / \rho_n)^2 - 1} \right)^{1/2} - \frac{\rho_0}{k \rho_\infty} \right). \quad (5)$$

Обозначим  $\gamma = -\ln(1-\alpha)$  и  $y_n$  равным правой части (5), тогда

$$y_n = \gamma n \quad (6)$$

Искомый угловой коэффициент  $\gamma$  прямой (6) можно получить с помощью линейной регрессии в скользящем окне. В нашей реализации мы взяли длину окна равной  $1/\alpha$ . Формулы для углового коэффициента  $\gamma$  и его стандартного отклонения  $\sigma_\gamma$  при использовании процедуры линейной регрессии можно найти, например, в [1]. Параметры  $\rho_0$ ,  $k = \sigma_I / \sigma_{E_0}$  вычисляются по первому кадру, попавшему в окно. Значение  $\rho_\infty$  - по кадру, непосредственно следующему за последним кадром, попавшим в окно.

Заметим, что правая часть (5) не всегда может быть вычислена, а лишь при условии  $\rho_n < \rho_\infty$  (предполагается, что  $\rho_n < \rho_\infty$ , поскольку по смыслу задачи зависимость должна быть возрастающей). Будем считать набор входных данных  $y_n$  корректным, если условие  $\rho_n < \rho_\infty$  выполняется для всех элементов скользящего окна.

Введем две гипотезы:

$H_0: \gamma = \gamma^* = -\ln(1-\alpha)$  - перезахват;

$H_1: \gamma = 0$  или набор  $y_n$  для расчета  $\gamma$  некорректен (при некотором  $n$  нарушается условие  $\rho_n < \rho_\infty$ ) - нормальное сопровождение.

Построим логарифмическое отношение правдоподобия

$$\Lambda(\gamma) = \ln \frac{e^{-\frac{(\gamma - \gamma^*)^2}{2\sigma_\gamma^2}}}{e^{-\frac{\gamma^2}{2\sigma_\gamma^2}}} = \left( \frac{\gamma^*}{\sigma_\gamma} \right)^2 \left( \frac{\gamma}{\gamma^*} - \frac{1}{2} \right). \quad (7)$$

Теперь рассмотрим  $m$  последовательных значений  $\gamma$  и построим для них общее логарифмическое отношение правдоподобия следующим образом:

$$\Lambda_m = \sum_{i=1}^m \Lambda(\gamma_i). \quad (8)$$

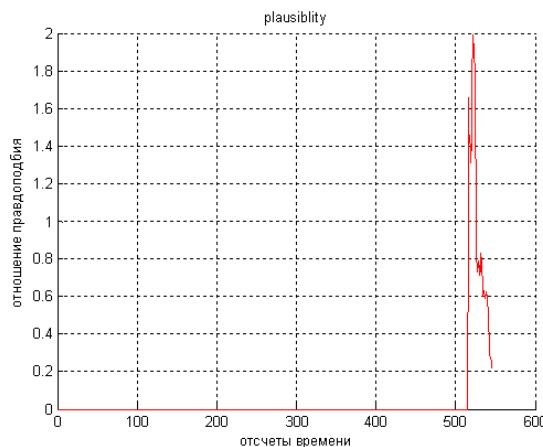
Теперь сформулируем алгоритм обнаружения. Он состоит из следующих шагов:

1. В процессе сопровождения вычисляем значение (7) и находим такой кадр, для которого  $\Lambda(\gamma_1) > 0$ .
2. Накапливаем сумму (8), если текущий  $\gamma_i$  таков, что  $\Lambda(\gamma_i) > 0$ . Если  $\gamma_i$  вычислить нельзя, или не выполняется условие  $\Lambda(\gamma_i) > 0$ , то переходим к шагу 1.
3. Если текущее значение суммы  $\Lambda_i > \eta$ , где  $\eta$  - некоторый порог, то принимаем решение об успешном обнаружении перезахвата, иначе повторяем шаг 2.

По сути описанный выше алгоритм реализует метод последовательного анализа [2]. Отличие заключается в сбросе накопления суммы (8) на шаге 2, а также в том, что не используется нижний порог для  $\Lambda_i$ .

Значение порога  $\eta$  подбирается по тестовым примерам, и вопрос о его теоретической оценке пока остается открытым.

Зависимость отношения правдоподобия (7) от времени представлена на рис. 2. Хорошо виден резкий рост (7) примерно через 100 кадров после перезахвата.



**Рис. 2** Зависимость логарифмического отношения правдоподобия (7) от времени.

Величина задержки пропорциональна длине окна, которая, в свою очередь, связана с параметром  $\alpha$ . Поэтому задержка может быть существенно уменьшена путем увеличения  $\alpha$ .

*Замечание.* Оказывается, что если в формуле (7) вместо  $\sigma_\gamma$  взять  $(\sigma_\gamma)^t$ , где показатель степени  $t > 1$  (в нашей реализации  $t = 2$ ), то эффективность обнаружения возрастает (уменьшается число ложных срабатываний при той же вероятности пропуска цели). Ясно, что при такой модификации алгоритма значение порога  $\eta$  должно быть подобрано заново.

## 5. ЗАКЛЮЧЕНИЕ

Эффективность рассмотренного алгоритма обнаружения перезахвата на ложную цель подтверждена тестированием на различных тестовых видеозаписях с существенно отличающейся фоново-целевой обстановкой.

Наиболее перспективным представляется комбинирование описанного метода обнаружения перезахвата с другими методами обнаружения срыва сопровождения, основанными на использовании адаптивных моделей объекта и фона. Для этого необходимо строить двухстадийные алгоритмы сопровождения, в которых на первой стадии определяются предварительные координаты объекта, а на второй стадии эти координаты используются в корреляционном алгоритме в качестве первого приближения. При такой реализации корреляционная стадия будет достаточно быстрой за счет того, что вместо глобальной минимизации можно использовать локальную.

В двухстадийном алгоритме доступна информация о процессе слежения как с первой, так и со второй (корреляционной) стадии. Это позволит существенно увеличить эффективность обнаружения срыва сопровождения.

Заметим, что координаты, которые будут получаться после второй стадии, можно как использовать в дальнейшем в качестве координат цели, так и не использовать. В последнем случае задачей второй стадии будет исключительно обнаружение перезахвата. В таком варианте алгоритма можно существенно увеличить значение

параметра  $\alpha$ , тем самым резко сократив величину задержки обнаружения срыва сопровождения.

## 6. ССЫЛКИ

- [1] Ивченко Г.И., Медведев Ю.И. **Введение в математическую статистику**: Учебник. М.: Издательство ЛКИ, 2010. – 600 с.
- [2] Харкевич А.А. **Борьба с помехами**. Издательство «Наука», Главная редакция физико-математической литературы, Москва, 1965.

## Об авторах

Ефимов Андрей Сергеевич – инженер-электроник ЗАО НТЦ «ЭЛИНС», студент МИЭТ. Его адрес: [dveller@gmail.com](mailto:dveller@gmail.com)

Козлитин Иван Алексеевич – научный сотрудник ЗАО НТЦ «ЭЛИНС», к.ф.-м.н. Его адрес: [kozlitin@elins.ru](mailto:kozlitin@elins.ru)

# Ускорение расчёта вторичного освещения с помощью фильтрации в пространстве экрана и уточнения на основе информации о близлежащей геометрии

Груздев А.М, Фролов В.А, Игнатенко А.В.

Московский государственный университет им. Ломоносова

факультет вычислительной математики и кибернетики, Москва, Россия

aleks-gruzdev92@yandex.ru, vfrarov@graphics.cs.msu.ru, ignatenko@graphics.cs.msu.ru

## Аннотация.

В данной работе рассмотрен алгоритм, основанный на аппроксимации точного расчета глобального освещения. Алгоритм работает в пространстве экрана и основан на идеи переиспользования информации о вторичной освещенности в соседних пикселях изображения. Программная реализация использует трассировку лучей в поле расстояний до поверхности и позволяет рассчитывать глобальное освещение интерактивно на современных графических процессорах (2-5 FPS, GTX 570).

**Ключевые слова:** GPU, трассировка лучей, глобальное освещение.

## 1. ВВЕДЕНИЕ

Одна из основных задач компьютерной графики – синтез изображений, неотличимых для наблюдателя от фотоснимков реального мира. Реалистичное изображение можно получить, вычислив интеграл освещенности.

$$I(\varphi_r, \theta_r) = \iint_{\varphi_i, \theta_i} L(\varphi_i, \theta_i) R(\varphi_i, \theta_i, \varphi_r, \theta_r) d\varphi_i d\theta_i$$

**Рисунок 1.** Интеграл освещенности I представляет из себя двойной интеграл по полусфере.  $\varphi$  и  $\theta$  задают направление в сферических координатах, L – количество входящего излучения, R – дуалучевая функция отражения [Nicodemus 1965].

Эту задачу решают методы глобального освещения [Ritschel et al. 2011] [Kaplanyan et al. 2010] [Jensen et al. 2002] [Radax 2008]. Обладая, как правило, высокой вычислительной стоимостью данные алгоритмы позволяют синтезировать высококачественные изображения.

В силу свойства определенного интеграла, мы можем раскладывать функции R и L (рис.1) на несколько компонент, вычислять для них интеграл отдельно, а результат получать как линейную комбинацию полученных значений.

В данной работе мы сосредоточимся на ускорении только непрямого (то есть претерпевшего одно и более переотражений) диффузного освещения. Под термином 'непрямое диффузное освещение' (или 'вторичное диффузное освещение') мы будем понимать компоненту функции L, обусловленную только Ламбертовскими [Basri et al. 2000] переотражениями света.

### 1.1. ОБЗОР РАБОТ ПО СМЕЖНОЙ ТЕМАТИКЕ

Эффективный подход расчета глобального освещения, предложен Wang в работе [Wang et al. 2009]. Этот метод использует фотонные карты. Основное направление данной работы - использование кэша освещенности, фотонных карт и финального сбора [Jensen et al. 2002] для быстрого расчета диффузного вторичного освещения. Прямое освещение в работе [Wang et al. 2009] рассчитывается с помощью трассировки лучей с учётом резких теней от точечных источников света. Недостаток подхода предложенного в [Wang et al. 2009] – построение

kd-дерева в реальном времени на графическом процессоре, что требует дополнительных затрат памяти. Память является одним из важнейших ресурсов для современных программ рендеринга. Это особенно верно для GPU в связи с тем, что памяти на графических процессорах, как правило, меньше.

Аналогичный подход используется в [Fabianovski and Dingliana 2009], но без кэша освещенности и финального сбора. Прямое освещение рассчитывается трассировкой лучей, а вторичное – с помощью фотонных карт. Этот алгоритм подходит для каустиков, но даёт зашумлённые результаты для вторичного освещения.

McGuire и Luebke [McGuire and Luebke 2009] скомбинировали растеризацию для первичного освещения с выполняемой на центральном процессоре трассировкой фотонов и растеризацией сплотов для расчёта вторичного освещения на графической карте.

Другой подход к расчёту глобального освещения рассмотрен в [Ritschel et al. 2009]. Этот метод основан на вычислении переноса вторичного освещения не с помощью трассировки лучей в сцене, а за счёт вычислений в экранном пространстве. Такое упрощение расчетов является большим плюсом данного метода, но он имеет и серьёзные недостатки, а именно: только одно переотражение, локальное вычисление вторичного освещения.

### 1.2. Кэш освещенности

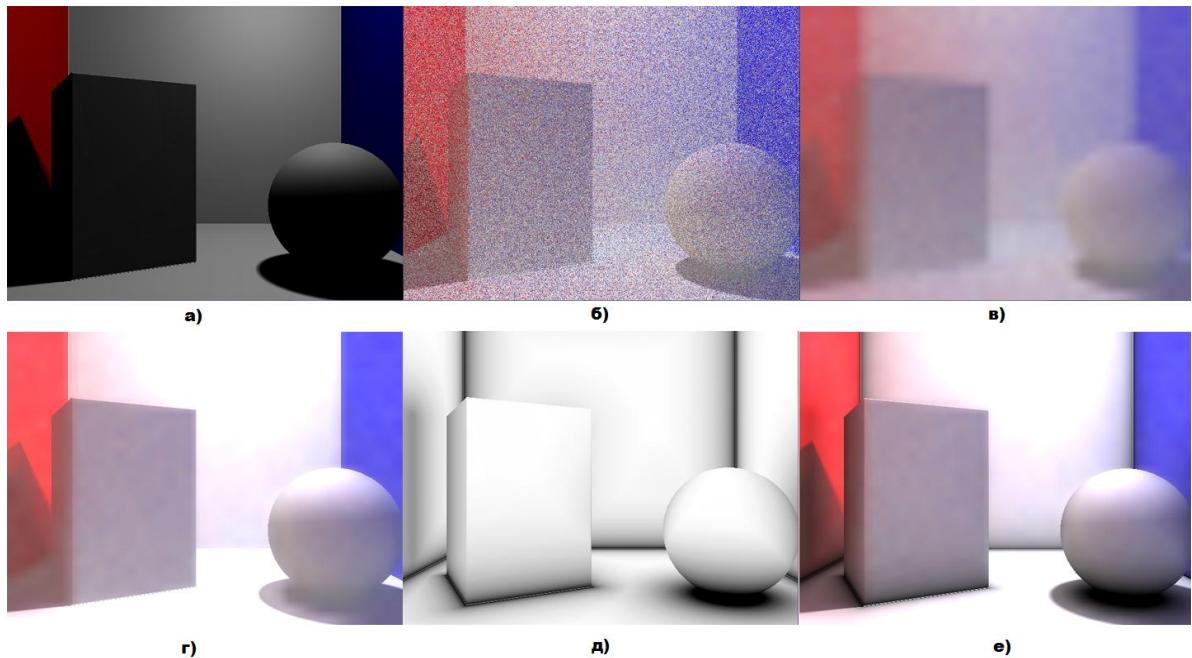
Кэширование освещенности снижает общую стоимость расчета вторичного освещения выполнением выборки по полусфере (или финального сбора) только в отмеченных точках сцены, кэшируя результаты и используя несколько раз сохранённые значения вторичного освещения за счёт интерполяции между точками кэша. Это было предложено в [Ward et al. 1988]. Алгоритм может быть кратко представлен в следующем виде:

```
if возможна интерполяция then
    используй сохранённые значения для интерполяции;
else
    вычисли новое значение;
    сохрани в кэше;
end if
```

Количество точек кэша освещенности обычно на 1-2 порядка меньше количества пикселей, таким образом, кэш довольно эффективен и может значительно ускорить весь рендеринг.

Кэширование освещения на графическом процессоре подробно описано в [Krivanek et al. 2008] и [Gauton et al. 2005]. Основная их идея – заменить финальный сбор растеризацией сплотов. Для конкретной записи кэша возможно определить, какие видимые точки вносят вклад в расщепление записи на плоскости изображения. Используя аппаратно ускоренную растеризацию для семплирования по полусфере над каждой записью и так же для расщепления записей в специальном буфере, возможно добиться высокой производительности.

Метод "irradiance hints" представленный в [Papaioannou 2011] является стабильным (для анимации) и обладает высокой скоростью при вычислении диффузного глобального освещения. Метод основан на сеточном кэшировании освещения с отражающими теневыми



**Рисунок 2.** Работа алгоритма по шагам. а) – первичное освещение по Фонгу. б) – грубый расчёт освещённости по 4 случайным лучам. в) – отфильтрованное при помощи размытия вторичное освещение. г) - изображение 'а' с наложенным вторичным освещением д) - ambient occlusion е) – результирующее изображение.

картами и может обрабатывать множественные переотражения света.

В [Arikan et al. 2005] впервые предложен метод разделяющий освещение на ближнее и дальнее. Он использует кэш вторичного освещения и представление вторичного освещения с помощью сферических гармоник. Кэш позволяет уменьшить количество точек, в которых необходимы расчёты вторичного освещения; представление освещения с помощью гармоник значительно упрощает расчет интеграла освещённости. Данный метод даёт хороший результат, но не подходит для интерактивного рендеринга из-за слишком долгого времени работы.

### 1.3. Выводы

Все рассмотренные работы предлагают эффективные методы расчета глобального освещения. Тем не менее, одни из них имеют недостатки (например, [Arikan et al. 2005] в скорости), другие применимы только для некоторых частных случаев (например, [Radax 2008] только для сцен из диффузных материалов). Задачей нашей работы не является предложить некоторый новый метод, который будет лучше рассмотренных, а разработать метод ускорения расчета вторичного освещения, который может быть применён для уже некоторого существующего алгоритма (например, [Arikan et al. 2005]) и позволит ускорить его работу, но, при этом, существенно не потеряв качество синтезируемого изображения.

## 2. ПРЕДЛОЖЕННЫЙ МЕТОД

Аналогично статье [Arikan et al. 2005] в предложенном алгоритме вторичное освещение разделяется на 2 компоненты – ближнее и дальнее освещение. На первом шаге нами вычисляется компонента, представляющая дальнее освещение.

Следуя идеи кэширования освещенности, наш алгоритм основывается на том, что вторичное освещение, обусловленное диффузными переотражениями света в пространстве экрана меняется с небольшой скоростью и при переходе между соседними пикселями информация о

вторичном освещении может быть переиспользована. Однако в областях изображения с высокой геометрической детализацией, как известно [Jensen et al. 2002], эффективность переиспользования снижается и известный алгоритм кэширования освещенности [Krivanek et al. 2008] вычисляет вторичную освещенность избыточное количество раз.

В противовес кэшу освещенности, в областях высокой геометрической детализации наш алгоритм уточняет вторичное освещение на основе информации о локальной форме поверхности при помощи аппроксимации “Ambient Occlusion” [Langer et al. 2000] Предложенный алгоритм можно разделить на три основных шага:

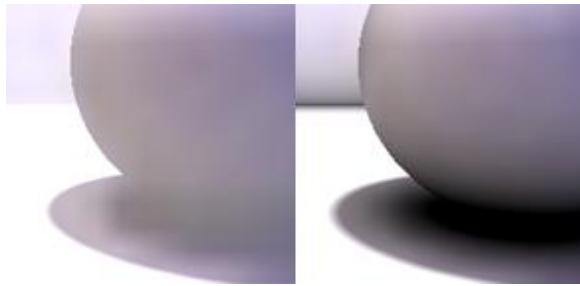
### 1. Приближенное вычисление дальнего освещения

На данном этапе для каждого пикселя производится обратная трассировка луча, находится пересечение со сценой и производится грубый расчёт интеграла освещённости методом Монте-Карло с небольшим числом сэмплов (4-16). Полученная информация записывается в текстуру экранного размера для последующего использования (рис 2.б). В текстуру сохраняется суммарное падающее освещение, без учета цвета материала текущей точки (т.е. падающее освещение на данном этапе не домножается на цвет материала).

В отдельные текстуры сохраняется информация о позициях, материалах и нормалах геометрии (т.е. строится G-буффер [Ferko et al. 2012]).

### 2. Фильтрация дальнего освещения

Так как мы использовали небольшое число Монте-Карло сэмплов, информация о вторичном освещении получается сильно зашумлённая и для получения приемлемого для глаза результата ее необходимо фильтровать. В данной работе в качестве фильтра использовалось построение цепочки mip-мап уровней с последующим размытием (рис. 2.в). Этот алгоритм был выбран нами вследствие высокой скорости работы и простоты реализации. Однако мы полагаем, что использование более сложной фильтрации (например, подавление шума на GPU, описанное в [Kharlamov et al. 2007]) в данном случае разумно и принесет более качественные результаты.



**Рисунок 3.** Уточнение освещения с помощью Ambient Occlusion

Мы также полагаем, что использование других методов, ускоряющих вычисление вторичной освещенности (например, кэш освещенности в пространстве экрана, LPV [Kaplanyan et al. 2010], instant radiosity [Kinkelin et al. 2008]) для вычисления дальнего освещения возможно вместо описанных шагов Монте-Карло сэмплирования с последующей фильтрацией. Однако в данной работе мы не исследовали этот вопрос.

### 3. Синтез итогового изображения

На финальном шаге информация о позициях, нормалях и материалах считывается из G-буфера. Первичное освещение вычисляется по модели Фонга. Вторичное освещение вычисляется (с учетом цвета материала) при помощи аппроксимации "Ambient Occlusion" (рис. 2.д) с тем отличием, что так называемый "фоновый" цвет, или компонента ambient, берется равной не белому цвету (как полагается в классической реализации "Ambient Occlusion"), а цвету, сохраненному в текстуре дальнего

освещения (рис 2.в). Уточнение освещения с помощью Ambient Occlusion позволяет избавиться от артефактов в виде ступенчатости после работы первых двух шагов (рис. 3)

## 3. ОЦЕНКА СТОИМОСТИ АЛГОРИТМА

Стоимость алгоритма складывается из:

1. Стоимости расчета вторичного освещения
2. Стоимости фильтрации текстуры вторичного освещения
3. Стоимости вычисления Ambient Occlusion

Выбирая различные алгоритмы для каждого шага, возможен выигрыш либо в скорости работы, либо в точности результата.

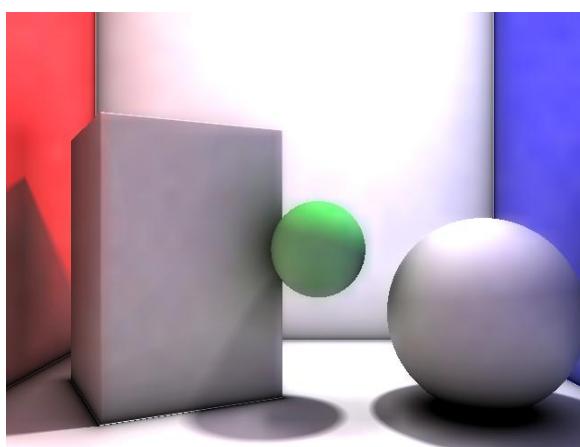
## 4. РЕАЛИЗАЦИЯ И РЕЗУЛЬТАТЫ

В программной реализации трассировка лучей производилась на основе функции расстояния [Quilez 2008] и геометрии, таким образом, задавалась аналитически. Данный вид трассировки лучей позволяет быстро и точно вычислять "Ambient Occlusion", однако не является оптимальным для простых примитивов, с которыми легко вычислять пересечение аналитически. Использованная трассировка лучей на основе функций расстояния, тем не менее, не ограничивает общность представленного в работе метода. Например, возможна реализация алгоритма для полигональной геометрии с использованием алгоритма расчёта Ambient Occlusion на основе буфера глубины, описанного в [Bavoil et al. 2008].

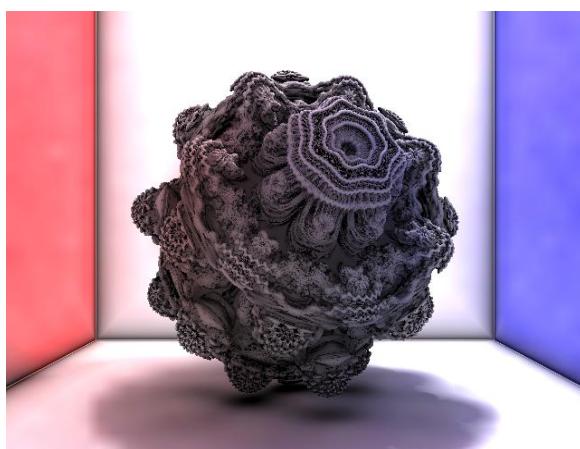
Программная реализация предложенного алгоритма выполнена на OpenGL4.0 и языке GLSL. Она протестирована на тестовых сценах (рис. 4). На сцене (рис 4. а) была достигнута скорость 2-5 FPS и на (рис. 4 б) – 1-3 FPS (в зависимости от минимальной длины шага, заданной для трассировки лучей на основе функций расстояния) на графическом процессоре NVIDIA GTX570.

## 5. ЛИТЕРАТУРА

- [1] [Arikan et al. 2005] Arikan O., Forsyth D. A. and O'Brein J. F. 2005. *Fast and Detailed Approximate Global Illumination by Irradiance Decomposition*. In Proceedings of ACM SIGGRAPH 2005, 1108-1114.
- [2] [Basri et al. 2000] Basri R. And Jacobs D. 2000, *Lambertian reflectance and linear subspaces*. Tech. Rep. MCS00-21, 2000-172R, Waizmann Institute of Science, NEC Research Institute
- [3] [Bavoil et al. 2008] Bavoil L. and Sainz M. 2008, *Screen Space Ambient Occlusion*, NVIDIA Corporation
- [4] Fabianovski and Dingliana 2009] Fabianovski, B. and Dingliana, J., *Interactive Global Photon Mapping*, Proc. of the EUROGRAPHICS Conference, 2009
- [5] [Ferko et al. 2012] Ferko M. and Valient M. 2012, *Real-time Lightning Effects using Deferred Shading*. Proceedings of CESCG 2012
- [6] [Gauton et al. 2005] Pascal Gautron, Jaroslav Krivánek, Kadi Bouatouch, and Sumanta Pattanaik. *Radiance cache splatting: A GPU-friendly global illumination algorithm*. In Proceedings of Eurographics Symposium on Rendering, June 2005.
- [7] [Jensen et al. 2002] 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).



(а)



(б)

**Рисунок 4.** Тестовые сцены

- [8] [Langer et al. 2000] Langer M.S. and Buelthoff H. H. 2000, *Depth discrimination from shading under diffuse lighting*, Perception. 29 (6) 649—660
- [9] [Kaplanyan et al. 2010] Kaplanyan A. and Dachsbacher C. 2010, *Cascaded Light Propagation Volumes for Real Time Indirect Illumination*, Crytek
- [10] [Kharlamov et al. 2007] Kharlamov A. and Podlozhnyuk V. 2005, *Image Denoising*, NVIDIA Corporation
- [11] [Kinkelin et al. 2008] Kinkelin M. and Liensberger C. 2008, *Instant Radiosity An Approach for Real-Time Global Illumination*, ICGA/TU Wien SS2008
- [12] [Krivanek et al. 2008] Krivánek, 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.
- [13] [McGuire and Luebke 2009] McGuire M. and Luebke D. 2009, *Hardware-Accelerated Global Illumination by Image Space Photon Mapping* Proc. of ACM SIGGRAPH/EuroGraphics High Performance Graphics 2009
- [14] [Nicodemus 1965] Nicodemus E. Fred, 1965. *Directional reflectance and emissivity of an opaque surface* (abstract). Applied Optics 4 (7): 767–775. DOI:10.1364/AO.4.000767
- [15] [Papaioannou 2011] G. Papaioannou, *Real-Time Diffuse Global Illumination Using Radiance Hint*. Presented at High Performance Graphics 2011, Vancouver, Canada, Aug. 2011.
- [16] [Quilez 2008] Quilez I., 2008. *Rendering Worlds with Two Triangles with raytracing on the GPU*. NVSCENE 08
- [17] [Radax 2008] Radax I., 2008, *Instant Radiosity for Real-Time Global Illumination*, Vienna University of Technology.
- [18] [Ritschel et al. 2011] Ritschel T., Dachsbacher C., Grosch T., and Kautz J., 2011, *The State of the Art in Interactive Global Illumination*, DOI: 10.1111/j.1467-8659.2012.02093.x
- [19] [Ritschel et al. 2009] Ritschel T., Grosch T. and Seidel H.P. 2009. *Approximating Dynamic Global Illumination in Image Space*. [18:40:38] Vladimir Frolov: Proceedings ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, Boston, MA, February 27—March 1, 2009
- [20] [Ward et al. 1988] Ward, G., Rubinstein, F., and Clear, R. 1988. *A ray tracing solution for diffuse interreflection*. In SIGGRAPH 1988, Computer Graphics Proceedings.
- [21] [Wang et al. 2009] 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.

## Об авторах

Алексей Груздев - студент третьего курса ВМК МГУ имени М.В. Ломоносова. Его адрес: aleks-gruzdev92@yandex.ru  
Владимир Фролов – аспирант ИПМ имени М.В. Келдыша. Он работает в NVIDIA инженером по развитию технологий. Его адрес: vfrarov@graphics.cs.msu.ru  
Алексей Игнатенко - научный сотрудник, к.ф.-м.н ВМК МГУ имени М.В. Ломоносова. Он занимается исследованиями в области фотorealистичного 3D рендеринга. Его адрес: ignatenko@graphics.cs.msu.ru

# Применение Нейронной Сети Для Распознавания Частных Признаков Дактилоскопических Изображений

Гудков Владимир Юльевич, Челябинский государственный университет, Челябинск, Россия

diana@sonda.ru

Дорофеев Константин Андреевич, Челябинский Государственный Университет, Челябинск, Россия

kostuan1989@mail.ru

## АННОТАЦИЯ

В статье рассматриваются вопросы распознавания частных признаков дактилоскопических изображений и методы их решения, а также возможность применения нейронных сетей для распознавания изображений с априорной неопределенностью.

**Ключевые слова:** биометрия, нейронные сети, отпечатки пальцев, распознавание.

## 1. ВВЕДЕНИЕ

Целью работы является исследование методов распознавания дактилоскопических изображений (ДИ) на основе нейронных сетей (НС) и методов обучения НС на эталонных векторах [1, 3] в виде структурных особенностей ДИ [4, 5]. Для достижения поставленной цели были решены следующие основные задачи:

- разработка и тестирование алгоритма функционирования и обучения НС на основе метода обратного распространения ошибки [3];
- разработка и тестирование алгоритмов предварительной обработки ДИ, которая подготавливает входные векторы для обучения НС [2];
- подготовка обучающего и тестового массива ДИ;
- обучение НС и проверка ее работы [1].

Практическая ценность работы заключается в возможности применения выработанных концепций и алгоритмов в системах контроля и управления доступом, системах верификации и идентификации личности [5].

## 2. МЕТОД ОБУЧЕНИЯ СЕТИ

Для исследования поставленных задач разработана программа, считывающая битовую карту ДИ и выполняющая его предварительную обработку:

- построение и улучшение гистограммы [2];
- применение сглаживающего фильтра для удаления высокочастотных шумов [4];
- построение векторного поля градиента ДИ в виде препараторов модуля и аргумента [4].

В качестве выходных параметров предварительной обработки выбраны именно модуль и аргумент векторного поля градиента для простой окрестности по следующим причинам.

В исходном виде изображения представляют собой массив значений яркости пикселей [2]. Соответственно, мощность пространства всех изображений составляет значение

$$|F| = 256^{N \times M},$$

где  $N$  и  $M$  – ширина и высота изображения, 256 – число градаций серого. Так для изображений размером  $1000 \times 1000$

$$|F| \sim 2^{10^7}.$$

При вычислении модуля и аргумента векторного поля размерность пространства всех изображений уменьшается.

Действительно, учитывая пределы изменения модуля (0..200) и аргумента (0..180), а также размеры небольших окрестностей векторного поля (в области  $t \times t$  пикселей изменения векторного поля незначительны), получим:

$$|F| = 180^{(N \times M)/t^2} + 200^{(N \times M)/t^2}.$$

Например, для изображения  $1000 \times 1000$  пикселей и окрестности  $10 \times 10$  мощность составит:

$$|F| \sim C_1 * 2^{C_2 * 10^4},$$

где  $C_1, C_2$  – константы.

Расчеты показывают, что мощность пространства векторных полей меньше мощности пространства изображений. Известно, что НС позволяет разделять элементы  $n$ -мерного пространства и, поэтому, эффективнее создать сеть, разделяющую элементы пространства с меньшей мощностью [3]. Так как направление линий, которое определяет векторное поле, является ведущей характеристикой ДИ [5], векторное поле градиента было выбрано для обучения НС.

## 3. РЕЗУЛЬТАТЫ

Разработана программа, позволяющая обучать и тестируировать работу трехслойной НС архитектуры 81-27-3. Для работы программы сформирован тестовый и обучающий массив ДИ. На обучающих ДИ вручную экспертом-криминалистом указаны частные признаки ДИ (координаты и направления). Пример такого ДИ показан на рис. 1. Частные признаки определяют координаты, указывающие малые окрестности. По этим окрестностям для НС построены обучающие векторы как часть векторного поля градиента по правилу: одна окрестность – вектор модуля и вектор аргумента градиента.



Рис. 1: Частные признаки изображения

Дополнительно к указанным обучающим векторам вручную добавлены векторы, соответствующие областям изображения, не содержащих частных признаков. Окрестности таких векторов показаны квадратиками на рис. 2. Эти векторы, каждый из которых представляет вектор модуля и вектор аргумента градиента окрестности, также применялись для обучения НС. Они образуют класс векторов, для которых не должны детектироваться частные признаки ДИ.

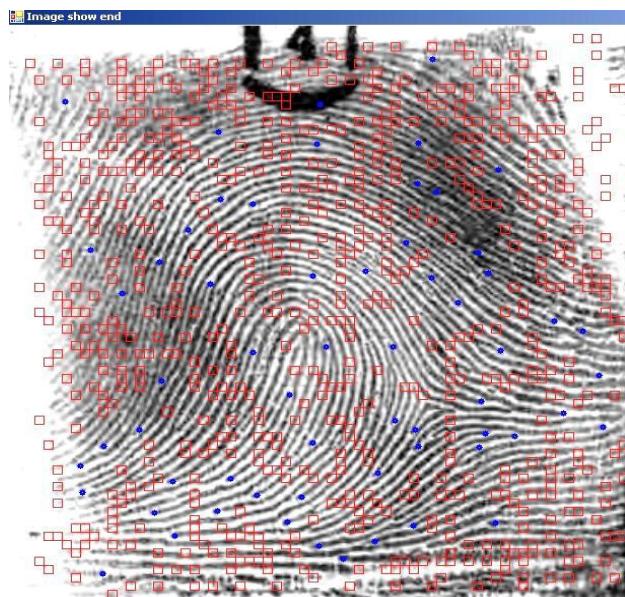


Рис. 2: Области с нулевым откликом

На выходе НС для каждой окрестности ДИ вычисляется вероятность того, что в выделенной окрестности расположен частный признак. Такие окрестности (всего 9x9 пикселей изображения) с масштабированной вероятностью показаны на рис. 3. Темным цветом отмечены области, в которых вероятность появления частного признака близка к нулю. Яркий цвет соответствует значительному отклику НС и в этой окрестности наиболее вероятно появление частного признака.

На этом же рис. 3. представлено детектированное поле направлений линий ДИ на основе векторного поля градиента, полученного на этапе предварительной обработки.

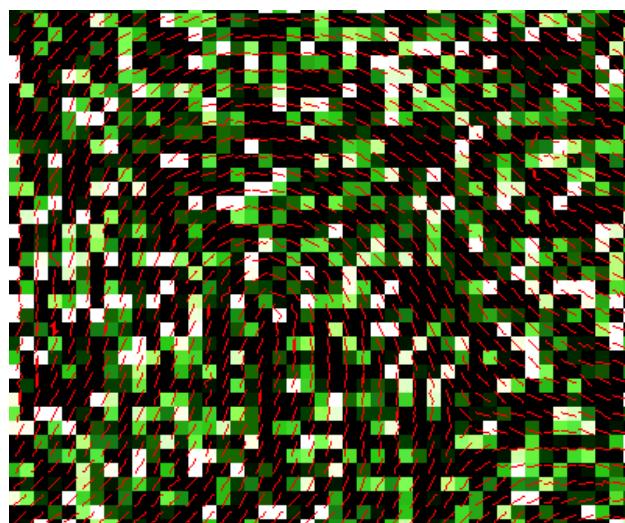


Рис. 3: Поле направлений и вероятности, рассчитанные на выходе НС

На рис. 4 крупным планом показаны результаты распознавания частных признаков ДИ с помощью НС. Области, в которых рассчитана высокая вероятность существования частного признака, выделены ярким квадратиком.

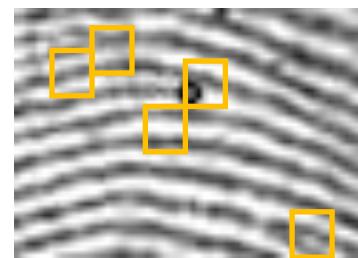


Рис. 4: Истинные и ложные области частных признаков

#### 4. ВЫВОДЫ

Эксперименты показали, что нейронная сеть обучается с ошибкой. На представленных рисунках видно, что отклик сети в "красных" областях близок к нулю. Также присутствует значимая вероятность правильного детектирования частных признаков. По мнению авторов полученных результатов для создания работающей дактилоскопической системы недостаточно, а общая задача распознавания частных признаков ДИ для НС является достаточно сложной.

Основные направления дальнейших исследований заключаются в увеличении мощности входного вектора НС до 400, исследовании различных активационных функций, в том числе гармонических, модификации алгоритма обучения с целью уменьшения ошибки обучения и сравнение различных способов предварительной обработки ДИ для формирования обучающих векторов НС.

#### 5. ССЫЛКИ

- [1] Галушкин, А.И. Теория нейронных сетей: учебное пособие для вузов: кн. 1 / А.И. Галушкин. – М.: Изд-во ИПРЖР, 2000. – 416 с.
- [2] Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс; пер. с англ.; под ред. П.А. Чочиа. – М.: Техносфера, 2006. – 1072 с.
- [3] Осовский, С. Нейронные сети для обработки информации / С. Осовский; пер. с польского И.Д. Рудинского. – М.: Финансы и статистика, 2002. – 344с.
- [4] Яне, Б. Цифровая обработка изображений / Б. Яне; пер. с англ. А.М. Измайлова. – М.: Техносфера, 2007. – 583 с.
- [5] Ratha, N. Automatic Fingerprint Recognition Systems / N. Ratha. – New York.: Business Media, 2004. – 458 с.

#### Об авторах

Гудков Владимир Юльевич – кандидат технических наук, доцент Челябинского государственного университета. Его адрес: [diana@sonda.ru](mailto:diana@sonda.ru).

Область научных интересов: обработка изображений, распознавание образов, дактилоскопические технологии.

Дорофеев Константин Андреевич – аспирант кафедры прикладной математики Челябинского государственного университета. Его адрес: [kostuan1989@mail.ru](mailto:kostuan1989@mail.ru).

Область научных интересов: биометрия, нейронные сети, обработка изображений.

# Параллельное Программирование Алгоритма Идентификации Дактилоскопических Изображений

Гудков Владимир Юльевич, Челябинский государственный университет, Миасс, Россия

diana@sonda.ru

Лепихова Дарья Николаевна, Южно-Уральский государственный университет, Челябинск, Россия

daria.lepikhova@yandex.ru

## Аннотация

В статье рассматривается алгоритм идентификации дактилоскопических изображений на основе модели топологических векторов. Сравнение выполняется путем оценки степени подобия векторов как отдельно по топологии и по геометрии, так и совместно. Предлагается вариант параллельной реализации алгоритма сравнения, основанный на технологии OpenMP. Приводятся результаты вычислительных экспериментов.

**Ключевые слова:** отпечаток пальца, сравнение шаблонов, параллельный алгоритм, OpenMP.

## 1. ВВЕДЕНИЕ

Идентификация личности по отпечаткам пальцев является наиболее распространенным методом биометрической идентификации [2]. В компьютеризированных системах такую идентификацию обычно выполняют по шаблонам, вычисляемым при обработке дактилоскопических изображений (ДИ) [3]. Современные автоматизированные дактилоскопические системы насчитывают до сотни миллионов изображений отпечатков пальцев и их шаблонов, а идентификация ДИ типа 1:m требует такого же числа сравнений различных пар шаблонов [2]. Поэтому фактор времени является решающим для обеспечения конкурентоспособности больших компьютеризированных систем [5]. В связи с этим задача сокращения времени работы алгоритма сравнения шаблонов ДИ актуальна.

Одним из способов повышения реактивности системы является параллельное программирование базы данных, при котором задача сравнения шаблонов остается однопоточной. Такой способ решения актуальной проблемы не требует внесения изменений в алгоритм сравнения шаблонов ДИ. При всей привлекательности, однако, такой подход не позволяет ускорить задачи верификации ДИ типа 1:1.

Задача заключается в выделении и адаптации функций и данных алгоритма сравнения шаблонов ДИ с целью его параллельного программирования.

## 2. СРАВНЕНИЕ ШАБЛОНОВ ДИ

Оценку степени сходства двух ДИ выполняют методом сравнения их шаблонов. Под шаблоном понимают математическую модель, которая содержит описание признаков изображения. В статье рассматривается представление ДИ на основе топологических векторов [4]. Тогда математическую модель представляют в виде

$$T = \{(m_i, V_i)\}, m_i = \{x_i, y_i, \theta_i, t_i\}, V_i = \{(e_l, n_l)\}, i \in I = 1..n, \quad (1)$$

где  $m_i$  – частный признак, а  $V_i$  – топологический вектор, соответствующий  $m_i$ ;  $(x_i, y_i)$ ,  $\theta_i$  и  $t_i$  – координаты, вектор направления как угол и тип частного признака,  $t_i \in \{0; 1\}$  (окончание и разветвление);  $e_l$  – событие, сформированное частным признаком номер  $n_l$  на связи номер  $l$  в векторе и

$l \in 0..w$ ;  $w$  – число связей в векторе (обычно до 35);  $n$  – число частных признаков изображения [3, 4].

При сравнении шаблонов  $T_1$  и  $T_2$  рассчитывают оценку подобия топологических векторов  $V_i \in T_1$  и  $V_j \in T_2$  и формируют упорядоченный набор этих оценок в виде

$$S_p(V_i, V_j) | V_i \in T_1, V_j \in T_2 \quad (2)$$

где  $p$  – номер элемента ряда, причем меньшим номерам соответствуют лучшие оценки [3].

В работе оценки подобия по (2) представляют композицией оценок векторов по топологии и по геометрии.

Рассмотрим два частных признака  $m_i$  и  $m_j$  и соответствующие им топологические векторы  $V_i$  и  $V_j$  с числом связей, равным  $w$ . Степень подобия векторов по топологии вычисляют на основе функции  $\lambda(e_l^i, e_l^j)$ , где  $e_l^i \in V_i$  и  $e_l^j \in V_j$  – события, располагаемые на одноименных связях в двух векторах, а  $l$  – номер связи, в виде

$$\Lambda(V_i, V_j) = \frac{\sum_{l=0}^w \lambda(e_l^i, e_l^j)}{w}. \quad (3)$$

Аналогично степень подобия векторов по геометрии вычисляют по формуле

$$\Psi(V_i, V_j) = \frac{\sum_{l=0}^w \psi(\omega(m_i, m_k), \omega(m_j, m_g))}{w}, \quad (4)$$

где  $w$  – число связей в векторе;  $\psi$  – функция расчета степени подобия двух пар частных признаков;  $\omega$  – функция расчета разностей геометрических характеристик для двух частных признаков;  $m_i, m_k \in T_1$  и  $m_j, m_g \in T_2$ , задаваемых их номерами  $i, k$  и  $j, g$  на одноименных связях, причем номера частных признаков  $k, g$  определяют по номеру связи  $l$ :  $k = n_l^i$  в векторе  $V_i$  и  $g = n_l^j$  в векторе  $V_j$ .

В целом оценку подобия двух топологических векторов в (2) оценивают на основе выражений (3) и (4) в виде полинома

$$S_p(V_i, V_j) = \sum_{l=1}^w (c_l \Lambda^{a_l}(V_i, V_j) \Psi^{b_l}(V_i, V_j)) \quad (5)$$

где  $c_l$  – коэффициенты;  $a_l$  и  $b_l$  – значения степеней,  $w$  – число связей. Здесь для простоты изложения векторы  $V_i$  и  $V_j$  предполагают равномощными. В частном случае оценка аналогична вероятности произведения или суммы событий. Собственно оценку подобия двух шаблонов  $T_1$  и  $T_2$  вычисляют в виде

$$S(T_1, T_2) = \frac{\sum_{p=1}^n S_p}{\min(n_1, n_2)} \quad (6)$$

где  $n$  – число частных признаков в шаблонах  $T_1$  и  $T_2$ , которое при изложении сути работы предполагают равным.

### 3. ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ

Алгоритм вычисления степени подобия шаблонов ДИ состоит из следующих основных этапов:

- выбор одноименных связей;
- оценка степени подобия векторов по геометрии;
- оценка степени подобия векторов по топологии;
- комплексная оценка степени подобия векторов;
- оценка степени подобия шаблонов.

На рис. 1 представлена общая схема оценки степени подобия векторов по геометрии, а на рис. 2 – по топологии.

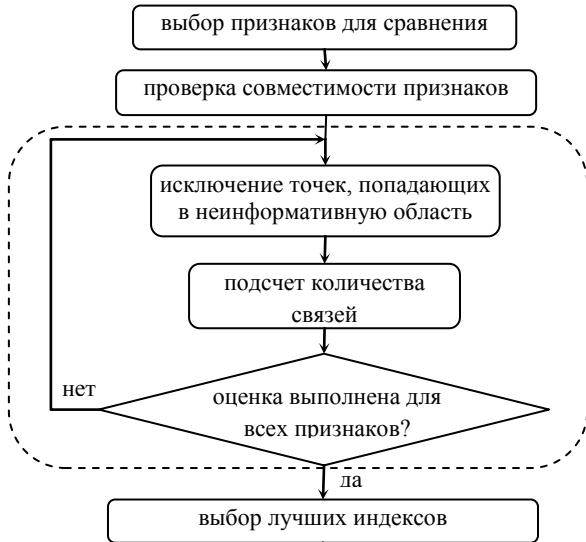


Рис. 1: Алгоритм оценки векторов по геометрии

На этапах, выделенных пунктиром, выполняется параллельная обработка данных, содержащих частные признаки изображения и связи между ними, причем каждый шаг вычислений выполняется независимо от других. Параллельные вычисления обеспечиваются технологией OpenMP [6].

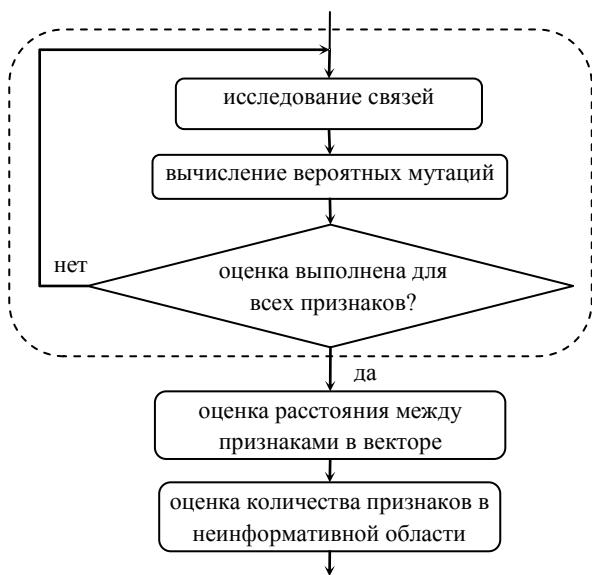


Рис. 2: Алгоритм оценки векторов по топологии

При оценке степени подобия по геометрии и по топологии применяется директива `omp for`. Каждый из параллельных процессов работает с собственным набором данных.

Проверка эффективности параллельных вычислений по серии вычислительных экспериментов показана на рис. 3.

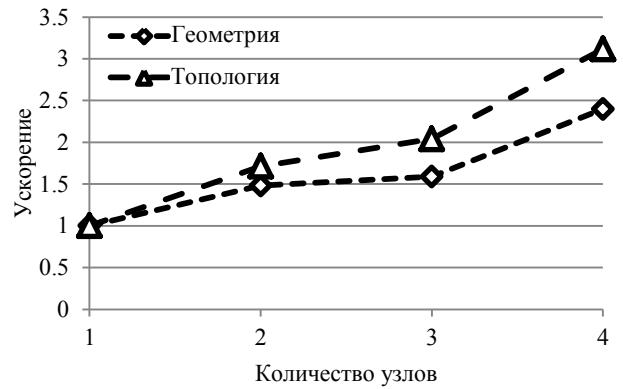


Рис. 3: Графики производительности алгоритмов

Общая схема параллельного алгоритма, который применялся в вычислительных экспериментах, представлена на рис. 4.

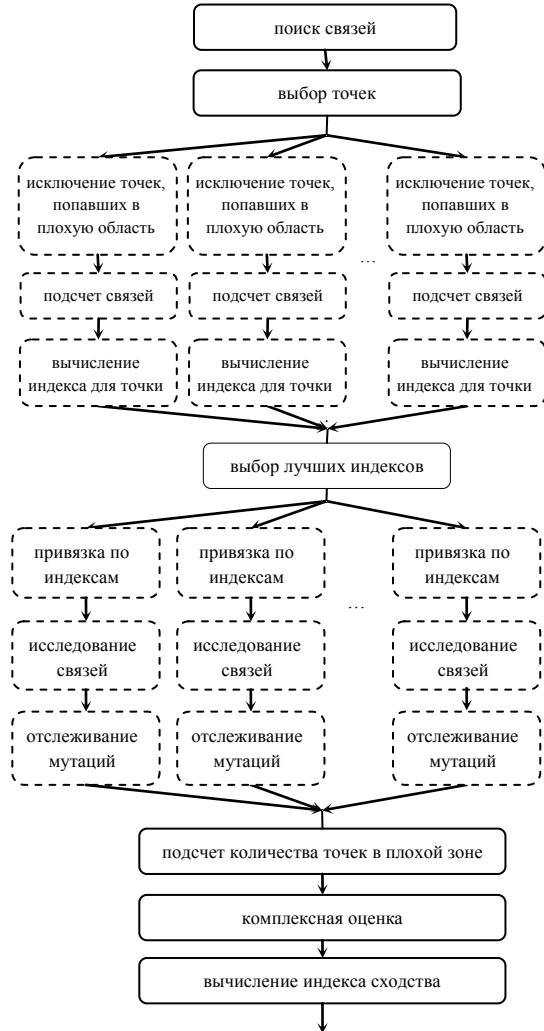


Рис. 4: Общая схема параллельного алгоритма

На графиках видно, что использование OpenMP позволяет сократить время оценки по геометрии в 1,8 раза, а по топологии – в 1,5 раза при организации 2 параллельных процессов. Здесь пунктиром отмечены параллельные секции, в которых возможно возникновение узких мест при оценке степени подобия. Универсальное решение для преодоления таких узких мест неизвестно [1]. Однако программирование позволяет выполнить следующую оптимизацию алгоритма.

## 4. МОДИФИКАЦИЯ АЛГОРИТМА

Общая оценка по топологии выполняется на основе вычисленных оценок подобия по топологии для каждой пары событий  $e_l^i$  и  $e_l^j$ . После оценки по топологии выполняется привязка по индексам точек группы. Чтобы избежать образования узких мест применяют:

- организацию параллельной обработки на узком участке при помощи технологии OpenMP;
- привязку по индексам совмещают с оценкой по топологии.

На рис. 5 показан псевдокод модифицированного алгоритма. Такая реализация алгоритма хотя и приводит к дублированию данных, однако значительно уменьшает время оценки.

```
#pragma omp parallel
{
    #pragma omp for
    for (i=0; i<n; i++)
        оценка_по_геометрии();
    #pragma omp single nowait
    выбор_индексов();
    #pragma omp for
    for (j=0; j<m; j++)
        #pragma omp single
        привязка_по_индексам();
        оценка_по_топологии();
}
```

Рис. 5: Алгоритм параллельных вычислений

Модифицированный алгоритм показывает лучшую производительность, но из-за неравномерного распределения циклов и нециклических разделов кода между потоками наблюдается и снижение производительности. Для преодоления указанных недостатков необходимо использовать основное преимущество OpenMP: единое адресное пространство для всех потоков, задействованных в одном процессе.

## 5. ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ

На рис. 6 приведен график зависимости среднего времени оценки подобия по топологии от числа параллельных процессов до преобразования алгоритма и после. Видно, что после преобразования алгоритм демонстрирует выигрыш по времени. Обе версии показывают некоторое снижение производительности при запуске 3 параллельных процессов, что объясняется необходимостью обмена приватными данными. Оптимальным является запуск алгоритма на  $2n$  процессах, где  $n$  – количество ядер процессора.

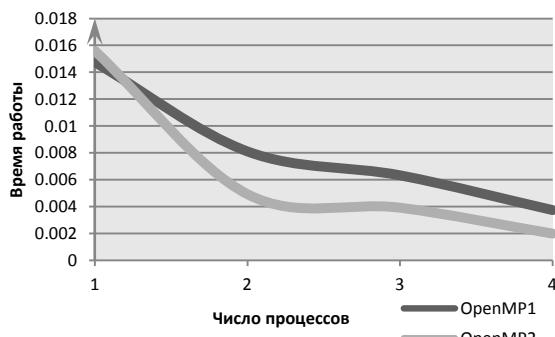


Рис. 6: Сравнение времени оценки по топологии

## 6. ЗАКЛЮЧЕНИЕ.

В работе рассматривается вариант ускорения алгоритма сравнения дактилоскопических изображений за счет организации параллельных вычислений на этапах выполнения оценок степени подобия векторов по топологии и по геометрии. Также предложен вариант оптимизации узких мест алгоритма, основанный на модификации параллельной версии этапа оценки по топологии. Рассмотренный вариант предполагает преобразование алгоритма таким образом, что привязка по индексам выполняется в том же цикле, что и оценка степени подобия векторов по топологии. При помощи директив OpenMP выполняется равномерное распределение нагрузки между параллельными потоками.

Предложенная реализация позволяет снизить время выполнения оценок по топологии и по геометрии в среднем в 1,8 раза при вычислении на двух потоках и в 3,5 раза на 4 потоках.

В данной работе технология OpenMP является базисом для построения параллельных вычислений. В качестве направления дальнейших исследований может быть выбрана реализация нового уровня параллелизма на основе графического процессора.

## 7. ЛИТЕРАТУРА

- [1] Андреев Н.Е. *Методы автоматизированного анализа производительности параллельных программ* // Вестник Новосибирского государственного университета. – 2009. – Т. 7, вып. 1. – С. 16-25.
- [2] Арутюнян А.Р. *Методы и алгоритмы анализа и синтеза деформаций дактилоскопических изображений: автореферат дис. ... канд. техн. наук* / М: Цифровичок, 2010. – 18 с.
- [3] Гордеева Д.Н., Гудков, В.Ю. *Распознавание дактилоскопических изображений* // Вестник МГТУ. Серия «Приборостроение». Специальный выпуск «Биотехнологии». – 2011. – С. 47–58.
- [4] А.С. Боков, В.Ю. Гудков. *Пат. 2185660 Российская Федерация, МПК G06K 9/52. Способ кодирования отпечатка папиллярного узора* – № 2000118065/09; заявл. 07.07.2000; опубл. 20.07.2002; Бюл. № 20. – 13 с.
- [5] Ушмаев О.С. *Проблемы распараллеливания биометрических вычислений в крупномасштабных информационных системах* // Информатика и ее применения, т.3, вып. 1. – 2009. – С. 8-18.
- [6] OpenMP Tutorial. URL: <https://computing.llnl.gov/tutorials/openMP/>

## Об авторах

Гудков Владимир Юльевич – доцент кафедры «Прикладной математики» Миасского филиала ЧелГУ  
e-mail: [diana@sonda.ru](mailto:diana@sonda.ru)

Область научных интересов: обработка изображений, распознавание образов, дактилоскопические технологии.

Лепихова Дарья Николаевна – инженер-программист ООО «Сонда-Технолоджи».  
e-mail: [daria.lepikhova@yandex.ru](mailto:daria.lepikhova@yandex.ru)

Область научных интересов: идентификация отпечатков пальцев, параллельные вычисления.

# Реализация методики улучшения и сегментации изображения темплета непрерывнолитой заготовки

Игорь Мацко, Оксана Логунова, Иван Посохов

Кафедра вычислительной техники и прикладной математики

Магнитогорский Государственный Технический Университет им. Г.И. Носова, Магнитогорск, Россия

matskoigor@gmail.com, logunova66@mail.ru, posohoff@bk.ru

## Аннотация

В работе приводится методика улучшения и сегментации изображения темплета непрерывнолитой заготовки и результаты ее программной реализации.

**Ключевые слова:** Непрерывнолитая заготовка, Темплет, Улучшение изображения, Сегментация изображения.

## 1. ВВЕДЕНИЕ

Современные производства используют компьютерное зрение с целью контроля технологического процесса, выявления дефектов, анализа поверхности и т.п. В черной металлургии компьютерное зрение широко применяют для анализа микроструктуры металлов и сплавов, при этом остается не затронутой область анализа макроструктуры. Существуют решения для автоматизированного выявления дефектов металла [1 – 3]. Авторами разрабатывается автоматизированная система определения макродефектов темплетов (образцов) непрерывнолитых заготовок (НЛЗ). Отличительные черты разрабатываемой системы от существующих:

- Рассматривается внутренняя структура непрерывнолитых заготовок.
- Ведется сегментация множественных не связных объектов нерегулярной формы, характеризующие дефекты нескольких видов.

## 2. ИСХОДНЫЕ ИЗОБРАЖЕНИЯ ТЕМПЛЕТОВ И ИХ ОЦЕНКА

После отбора темплета от НЛЗ, его отправляют на холодную механическую обработку, а затем на травление. После подготовки темплета его фотографируют (рис. 1) и снимают серный отпечаток (рис. 2). Дефекты оценивают согласно ОСТ 14-4-73 по оригиналу, фотографии и серному отпечатку.

Анализ множества фотографий и серных отпечатков темплетов выявил следующие задачи для исследования:

- Выделение области темплета из окружающего фона.
- Коррекция неоднородной освещенности темплета.
- Коррекция перспективы.
- Исправление резкости изображения.

При анализе изображений серных отпечатков были определены задачи исследования:

- Обработка низкоконтрастного изображения серного отпечатка.
- Устранение ложного выделения отметок оставленных сотрудниками с обратной стороны серного отпечатка.

Макродефекты НЛЗ оцениваются по ОСТ 14-4-73. Для дефектов выделяется четыре класса (рис. 3), которые оцениваются баллами от 0 до 4, с шагом в 0,5 балла.

Исследование структуры изображений было выявлено, что:

- Выделяемые объекты имеют нерегулярную форму и неопределенное место положение.
- Для изображений искажается прямоугольная система координат.
- Яркость изображения зависит от сортамента разливаемой и исследуемой стали.



Рис 1: Фотография темплета.

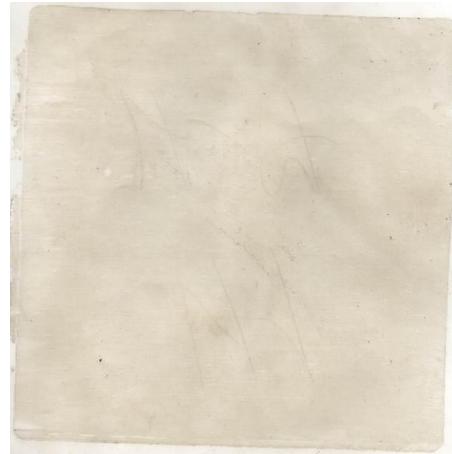


Рис 2: Серный отпечаток темплета по Бауману.

## 3. МЕТОДИКИ УЛУЧШЕНИЯ И СЕГМЕНТАЦИИ ИЗОБРАЖЕНИЯ ТЕМПЛЕТА НЕПРЕРЫВНОЛИТОЙ ЗАГОТОВКИ

Для реализации вышеуказанных задач и оценки макродефектов темплета была разработана методика улучшения и сегментации изображения (рис. 4). Методика построена на последовательном сочетании

морфологических операций [4] для улучшения изображения и принятия решений о его структуре.

На данный момент реализованы все этапы методики, за исключением блока 11.

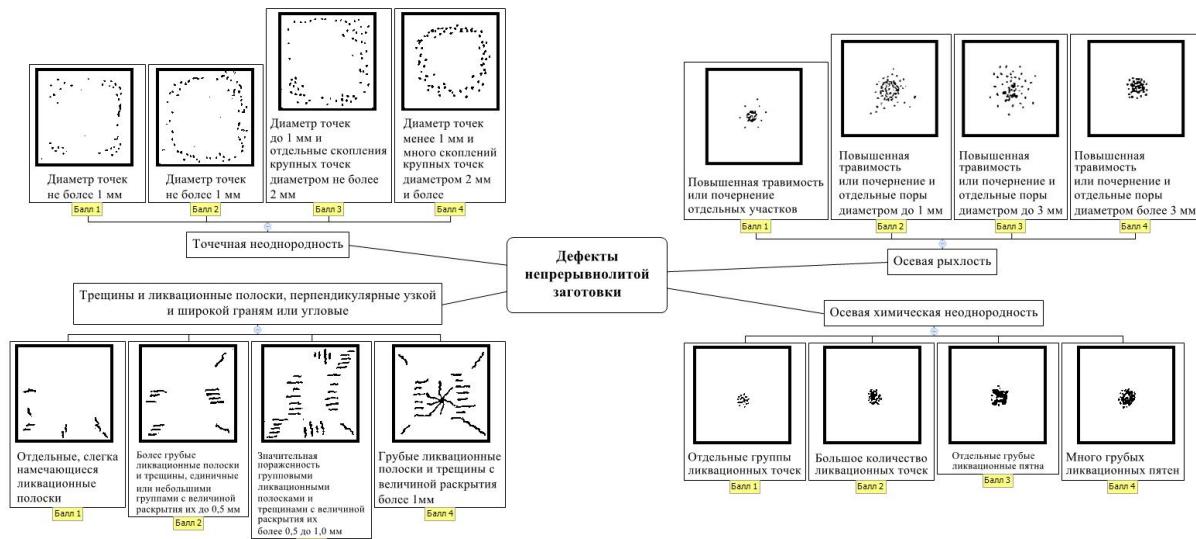


Рис 4: Классификация макродефектов непрерывнолитых заготовок и их бальная оценка

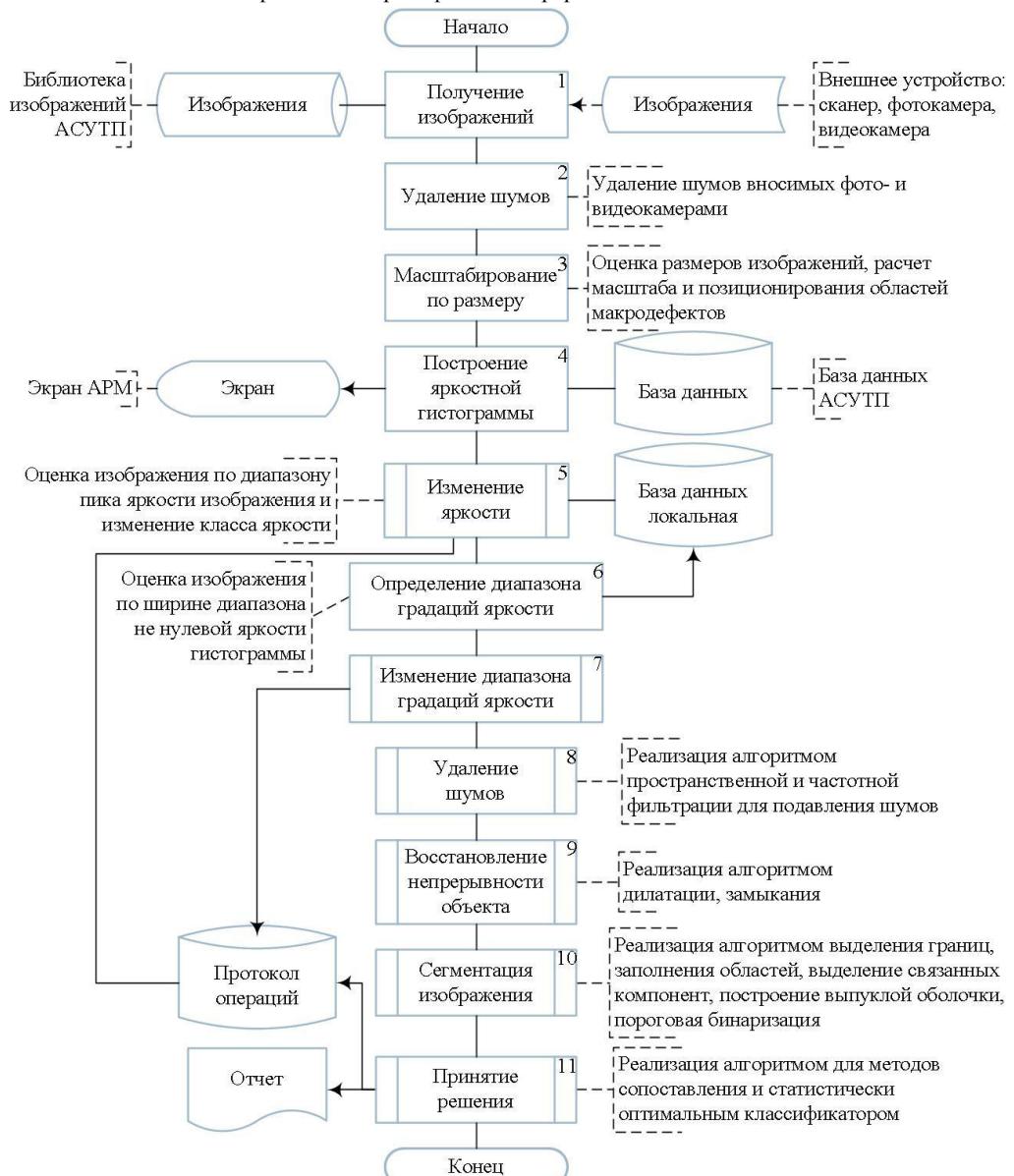


Рис 3: Методика улучшения и сегментации изображения темпleta непрерывнолитой заготовки.

Реализация построенной методики потребовала рассмотрение изображения в растровом виде и построения его математического описания в аффинной системе координат.

Для реализации построенной методики разработан программный продукт, который позволяет выполнить загрузку изображения, определение его яркостных характеристик и их отображение с помощью гистограммы, реализовать процедуры построенной методики по всем указанным алгоритмам, выполнить протоколирование выполненных действий и создать сценарий обработки для последующего применения к изображениям одного класса.

Отличительными особенностями программного продукта являются:

- Наличие широких функциональных возможностей для улучшения и сегментации изображения, включающий все алгоритмы перечисленные в методике на рис. 4.
- Возможность построения, сохранения и применения множества сценариев для обработки группы однотипных изображений. Каждый сценарий построен с условиями включения алгоритмов из каждой группы и исключения

дублирования операций, приводящих к одному результату.

- Наличие многооконного графического интерфейса, позволяющего организовать полнофункциональную работу с выбранным изображением.
- Возможность функционирования в качестве независимого приложения в рамках автоматизированного рабочего места технолога лаборатории.

Был проведен сбор информации в двух лабораториях ОАО «Магнитогорский металлургический комбинат». Эта информация представлена в графическом виде: фотографии темплетов и отсканированные изображения серных отпечатков. Была получена база, содержащая 384 фотографии темплетов и 71 изображение серных отпечатков. Опытная эксплуатация программного продукта позволила получить результаты обработки фотографий и серных отпечатков темплетов по разработанной методике. Вид основного рабочего окна программного продукта с результатами обработки изображений приведен на рис. 5. и рис. 6.

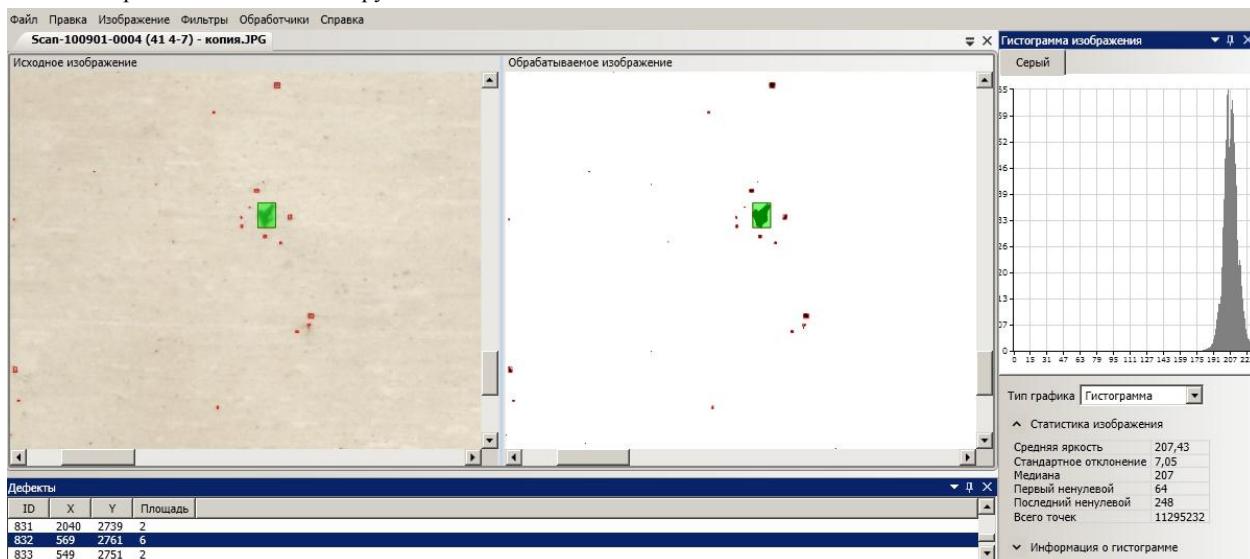


Рис 5: Результаты обработки серного отпечатка темплета с помощью разработанного программного продукта.

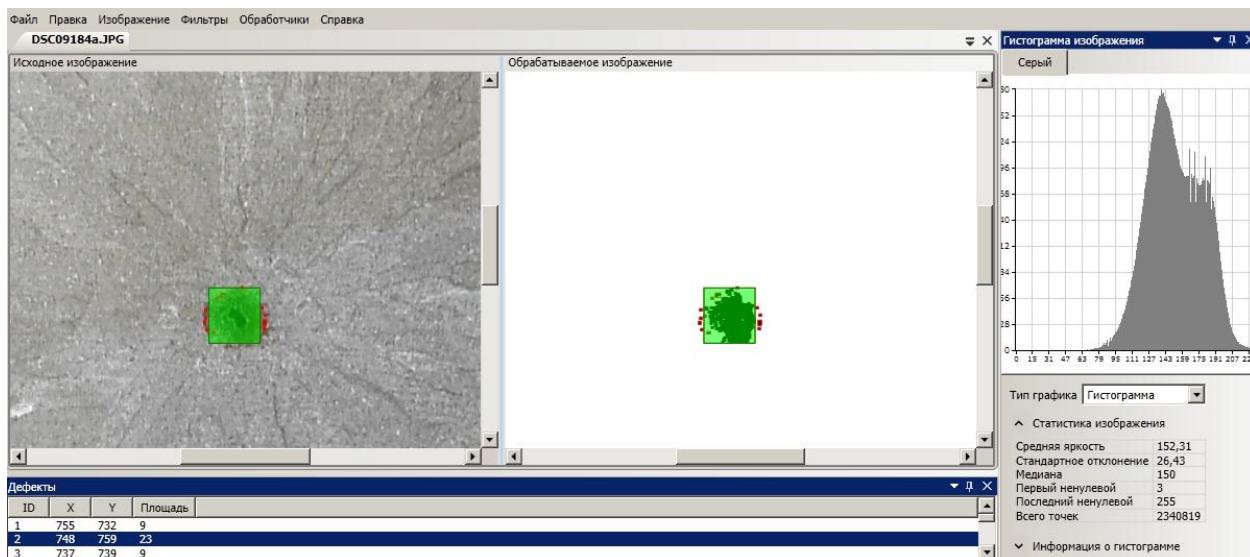
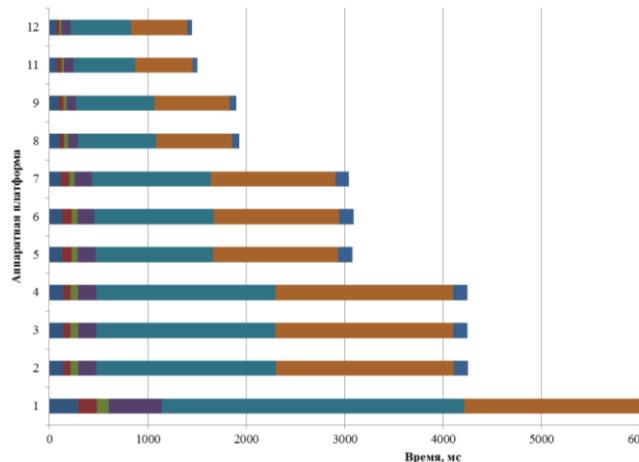


Рис 6: Результаты обработки фотографии темплета с помощью разработанного программного продукта.

В ходе исследования времени обработки изображений было выявлено, что при реализации методики, приведенной на рис. 3, затрачивается время в количестве от 1 до 7,5 с в зависимости от выбранной аппаратной платформы при условии использования в качестве основных ресурсов процессора и оперативной памяти. Результаты времени обработки изображения с высокой яркостью и низким контрастом приведены на рис. 7а. Как наиболее рациональная, для использования в производственных условиях, была определена конфигурация персонального



- Построение яркостной диаграммы
- Изменение яркости
- Определение диапазона градаций яркости
- Изменение диапазона градаций яркости
- Дилатация
- Эрозия
- Пороговая бинаризация

компьютера: процессор с тактовой частотой не менее 1,6 ГГц, оперативная память объемом не менее 1 ГБ. Объем носителя информации определяется использованием базового программного обеспечения и количеством обрабатываемых изображений. Объем одного изображения серного отпечатка составляет 30 – 40 Мб и одной фотографии – 2 – 3 Мб. Проводилось прогнозирование времени обработки изображений. Абсолютные погрешности данного прогнозирования представлены на рис. 7б.

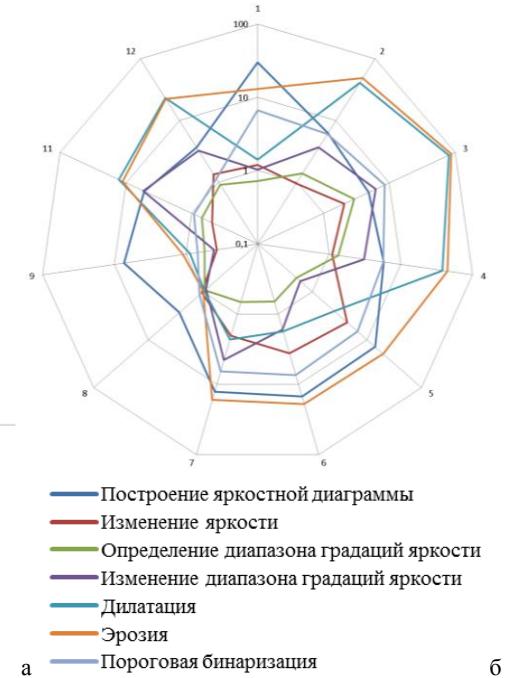


Рис 7: Результаты вычислительного эксперимента:

а – время работы алгоритмов обработки изображения;  
б – абсолютные погрешности прогнозирования времени обработки изображения.

## 4. ЗАКЛЮЧЕНИЕ

На основе анализа базы данных графической информации, полученной в ходе производственного процесса на одном из крупнейших металлургическом предприятии России, определены задачи для предварительной обработки и состав операций для улучшения и сегментации изображений сложной структуры. Предложена методика улучшения и сегментации таких изображения, и ее программная реализация, позволяющие повысить качество информации для ее использования в автоматизированной системе управления производством.

## 5. БЛАГОДАРНОСТЬ

Научные исследования выполняются при финансовой поддержке грантов Правительства Челябинской области и ФГБОУ ВПО «Магнитогорский государственный технический университет им. Г.И. Носова».

## 6. ССЫЛКИ

- [1] Guzaitis J., Verikas A. Image analysis and information fusion based defect detection in particleboards / J.Guzaitis, A. Verikas // Electronics and electrical engineering. – Kaunas: Technologija, 2006. – № 7. – С. 67-72.
- [2] Du-Ming Tsai, Tse-Yun Huang. Automated surface inspection for statistical textures / Du-Ming Tsai, Tse-Yun

- [3] Martins L.A.O. Automatic detection of surface defects on rolled steel using computer vision and artificial neural networks / L.A.O. Martins // IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society. – Glendale: 2010. – С. 1081-1086.
- [4] Matsko I.I., Snegirev Y.V., Logunova O.S. Data acquisition, preparation and processing methods by means of continuously-casted billets' quality analysis software / I.I. Matsko, Y.V. Snegirev, O.S. Logunova // International Journal of Applied Physics and Mathematics. – 2011. – Т. 1, № 2. – С. 106-111.
- [5] Гонсалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2005. – 1072 с.
- [6] Шапиро Л., Стокман Дж. Компьютерное зрение. – М.: БИНОМ. Лаборатория знаний, 2006. – 752 с.

## Об авторах

Игорь Мацко - аспирант каф. ВТ и ПМ ФГБОУ ВПО «МГТУ им. Г.И. Носова». Его адрес: matskoigor@gmail.com.  
Оксана Логунова – каф. ВТ и ПМ ФГБОУ ВПО «МГТУ им. Г.И. Носова». Ее адрес: logunova66@mail.ru.  
Иван Посохов - аспирант каф. ВТ и ПМ ФГБОУ ВПО «МГТУ им. Г.И. Носова». Его адрес: posohoff@bk.ru.

# Устойчивый Метод Выделения Границ Радужки в Видимом Диапазоне

Иван Малин

Факультет прикладной математики и физики

Московский Авиационный Институт, Москва, Россия

ivan.malin@gmail.com

## Аннотация

В данной статье предлагается новый алгоритм выделения границ радужной оболочки, устойчивый к ряду помех, типичных для изображений глаза.

Метод основан на анализе градиента изображения с учетом особенностей предметной области. Результаты тестирования на открытых базах изображений

глаза показали высокую эффективность и устойчивость метода.

**Ключевые слова:** Выделение радужки, Биометрия, Отслеживание направления взгляда

## 1. ВВЕДЕНИЕ

Методы выделения границ радужки на изображении глаза являются актуальным предметом исследования и находят свое применение в большом количестве приложений. В первую очередь это биометрические системы, которые в настоящее время активно развиваются и используются как составные часть систем безопасности. Выделение границ радужной оболочки требуется при получении рисунка радужки, индивидуального для каждого человека, определения цвета глаз и межглазного расстояния. Также положение радужки является важным признаком при построении систем отслеживания направления взгляда и бесконтактных интерфейсов, основанных на движениях глаз. При этом границы между радужкой и белком глаза остаются различимыми даже в условиях низкой освещенности, чего нельзя сказать о границах зрачка.

Таким образом, необходимо иметь эффективный метод выделения радужки на изображении глаза, устойчивый к условиям съемки и качеству полученного изображения. Существует ряд негативных факторов, затрудняющих успешное выделение радужки. Во-первых это общее низкое качество изображения: расфокусировка, шум, недостаточная контрастность. Во-вторых, это помехи, специфичные именно для данной предметной области: блики на роговице, частичное закрытие радужки веками и ресницами, тени от выступающих участков лица. Хороший метод должен успешно справляться с перечисленными помехами.

Следствием актуальности задачи является большое количество работ, посвященных данной проблеме. Так, в работе [6] был проведен обзор и апробация четырех известных методов. В качестве тестовых данных авторы использовали свою тестовую базу UBIRIS. UBIRIS - база данных изображений глаз, полученная в видимом диапазоне. Мотивацией ее авторов было создание тестовой базы изображений, подверженных описанным выше помехам, в отличие от существующих баз CASIA[1] и UPOL. В результате тестирования наилучшее детектирование показал метод Даугмана[2], затем, в порядке убывания процента правильных распознаваний, методы Wildes[7], Masek[4], Liam, Chekila[3]. В последних

трех подходах используется бинаризация контуров[7,4] или самого изображения[3]. Для ее осуществления необходимо задать один или несколько порогов бинаризации, в результате чего качество работы этих методов существенно зависит от правильного подбора параметров. Метод Даугмана является беспараметрическим и показывает хороший результат (93,5% корректных распознаваний на базе UBIRIS v1), однако и он обладает рядом проблем, которые были учтены при разработке предлагаемого метода.

## 2. ПРЕДЛАГАЕМЫЙ МЕТОД

Общая идея метода заключается в поиске таких параметров окружности (радиуса и координат центра), что направления векторов градиента изображения в каждой точке этой окружности будут максимально близки к направлению радиуса, проведенного к данной точке. То есть, будем искать координаты центра  $(i_0, j_0)$  и радиус  $r_0$ , при которых достигается максимум значений оператора

$$\oint_{O(i_0, j_0, r_0)} \langle N \nabla I(i, j), f(i_0, j_0, i, j) \rangle \cdot w(i_0, j_0, i, j) ds \quad (1)$$

Здесь  $I(i, j)$  – исходное изображение в градациях серого, а  $\nabla I(i, j)$  – его градиент.

$N$  – оператор локальной нормализации, применяемый к градиенту изображения. Мотивацией к его использованию послужил тот факт, что на изображении присутствуют контура, не являющиеся границами радужки, но при этом имеющие более резкий и амплитудный перепад яркости, например, ресницы, брови, блики, края век. Такие контуры вносят большой вклад в значение оператора (1) и смещают результирующую окружность, а нормализация контраста позволяет уравновесить их с вкладами, вносимыми пикселями границ радужки.

Для увеличения быстродействия метода использовалась интегральная гистограмма[5]. Данный объект позволяет получить гистограмму любой прямоугольной области изображения за малое константное время. Следует отметить, что операция локальной нормализации контраста градиента изображения не эквивалентна вычислению градиента изображения с локально нормализованным контрастом.

В операторе (1)

$$f(i_0, j_0, i, j) = \left( \frac{i - i_0}{\sqrt{(i - i_0)^2 + (j - j_0)^2}}, \frac{j - j_0}{\sqrt{(i - i_0)^2 + (j - j_0)^2}} \right)$$

является вектор-функцией, значение которой представляет собой вектор единичной длины, направленный из точки  $(i_0, j_0)$  в точку  $(i, j)$ . Иными словами, функция возвращает направление градиента изображения, которое должно быть в данной точке  $(i, j)$ , если через нее проходит окружность с центром в  $(i_0, j_0)$ . В общем случае данной функцией можно задать любую форму искомой фигуры.

Функция  $w$  в операторе (1) является весовой функцией, подключающей априорные данные о модели предметной области. В частности, при оценке положения радужки следует опираться скорее на участки ее границ, располагающиеся слева и справа от центра, нежели на верхний и нижний участки. Во-первых, нижний и верхний края радужки часто бывают закрыты веками и ресницами. Во-вторых, веки и ресницы сами по себе создают значимые контуры, которые не следует учитывать. Также при тестировании описываемого метода был увеличен вес больших окружностей по сравнению с маленькими, чтобы детектировать границы радужки, а не зрачка. Таким образом, весовая функция  $w$  имеет вид

$$w(r_0, i_0, j_0, i, j) = \frac{r_0}{\sqrt{1 + \frac{|i - i_0|^2}{|j - j_0|^2}}}$$

В результате, предлагаемый оператор (1) учитывает ряд особенностей предметной области, оставленных без внимания в других методах, что позволяет улучшить результаты детектирования.

### 3. РЕЗУЛЬТАТЫ РАБОТЫ МЕТОДА

В качестве тестовых данных были использованы вышеупомянутые базы UBIRISv1 и CASIA-Irisv1 [1]. База UBIRISv1 содержит 1871 изображение глаз, полученное в видимом диапазоне спектра, CASIAv1 – 767 изображений, полученных в диапазоне спектра, близком к инфракрасному. Тестовые базы не содержат эталонной разметки радужки, поэтому решение о корректности локализации границ выносилось на основе экспертной оценки трех специалистов.

В таблице 1 приведено сравнение результатов работы методов. Результаты использования известных методов взяты из работы [6].

Тестовая база	UBIRIS	CASIA
Предлагаемый метод	99.3 %	98.2 %
Daugman	93.5 %	54.4 %
Wildes	89.1 %	86.5 %
Masek	87.1 %	83.9 %
Liam, Chekima	47.9 %	64.4 %

Таб. 1: Сравнение результатов работы методов нахождения границ радужки.

На рисунке 1 приведены примеры корректной работы метода на тестовых изображениях. На рисунке 2 приведены примеры ошибок метода. Причиной данных ошибок стали высокая степень перекрытия радужки веками и экстремально низкий контраст изображения.

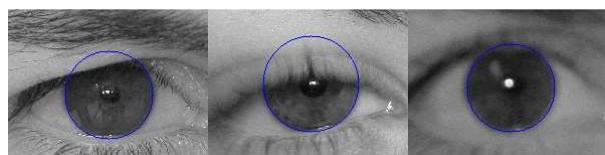


Рис. 1: Примеры корректного детектирования радужки на тестовых изображениях

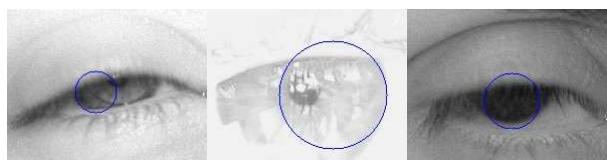


Рис. 2: Примеры ошибок метода

### 4. ЗАКЛЮЧЕНИЕ

Новый метод позволяет с большой надежностью получить корректный результат. Метод устойчив к характерным помехам и работает как на изображениях в видимом диапазоне спектра, так и на полученных в близком к инфракрасному диапазону. Важными отличительными особенностями метода являются локальная нормализация градиента изображения и использование весовой функции, учитывающей модель предметной области.

### 5. ССЫЛКИ

- [1] CASIA-IrisV1, <http://biometrics.idealtest.org/>
- [2] John G. Daugman. How Iris Recognition Works, 2004
- [3] Lye Liam, Ali Chekima, Liua Fan, and Jamal Dargham. Iris recognition using self-organizing neural network, 2002
- [4] Libor Masek. Recognition of Human Iris Patterns for Biometric Identification, 2003.
- [5] FatihPorikli. Integral Histogram: A Fast Way to Extract Histograms in Cartesian Spaces, 2005
- [6] HugoProen , Lu sA. Alex. UBIRIS: A Noisy Iris Image Databas, 2005
- [7] Richard P. Wildes. Iris recognition: an emerging biometric technology, 1997

# Классификация автодорожных знаков на основе свёрточной нейросети, обученной на синтетических данных

Моисеев Борис, Чигорин Александр

Факультет вычислительной математики и кибернетики

Московский Государственный Университет имени М.В.Ломоносова, Москва, Россия

mbd\_mezhgorie@mail.ru

## Аннотация

В данной статье представляется метод генерации синтетических изображений автодорожных знаков для машинного обучения свёрточной искусственной нейронной сети. За счёт этого метода можно получить большую тренировочную базу, не прилагая усилий для фотосъёмки и разметки полученных фотографий. Проанализированы различные приёмы, позволяющие получить наиболее реалистичные изображения. Также в данной работе рассматривается устройство свёрточной нейросети и анализируется эффективность её применения к задаче классификации дорожных знаков.

**Ключевые слова:** синтетические данные, свёрточная нейросеть, машинное обучение.

## 1. ВВЕДЕНИЕ

Системы распознавания автодорожных знаков имеют множество различных применений, таких как системы помощи водителю (ADAS - Advanced Driver Assistance Systems), управление автодорожной инфраструктурой и создание навигационных карт. На данный момент системы распознавания знаков включены в комплектацию автомобилей различных марок, но большинство из них имеют ряд существенных ограничений, сужающих спектр их применения; например, большинство систем способно распознавать лишь знаки ограничения скорости.

Одними из основных трудностей, с которыми сталкивается разработчик детектора и классификатора автодорожных знаков, являются жесткие ограничения на ресурсы, потребляемые программой. Бортовой автомобильный компьютер не может обладать большой вычислительной мощностью, равно как и большим объёмом оперативной памяти. Обе эти проблемы успешно решаются с помощью свёрточных нейросетей. Благодаря принципу разделяемых весов классификаторы на их основе не требуют большого объёма оперативной памяти, равно как и существенных вычислительных мощностей. Кроме того, как обучение, так и использование нейросети легко поддаётся распараллеливанию. Свёрточная нейросеть также может применяться и как детектор на основе скользящего окна, это применение оставим за рамками данной работы.

Однако при использовании нейросети разработчик может столкнуться с другой проблемой: качество классификации существенно зависит от объёма и презентативности тренировочной выборки. Для обучения эффективного классификатора необходимо иметь базу фотографий, содержащую знаки всех рассматриваемых классов в различных условиях освещения, снятых под различными углами и с различным зашумлением, возникающим при съёмке на камеру низкого разрешения на высокой скорости. Получение такой базы стоит существенных затрат времени и ресурсов.

Для устранения этой трудности предлагается метод генерации синтетических изображений. Используя пиктограммы знаков и фотографии природного и

городского пейзажа, с его помощью можно создать набор изображений, максимально приближенных к реальным фотографиям автодорожных знаков.

## 2. ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ

В работе [6] приводится сравнение результатов классификации при обучении на синтетических и реальных данных. Из графиков видно, что результаты классификатора, обученного на синтетических данных, существенно уступают результатам классификатора, обученного на реальных данных. Целью же этой работы является разработка алгоритма генерации синтетических изображений, таких, что обученный на них классификатор, пусть и на более массивной выборке, окажется не менее эффективным, чем классификатор, обученный на реальных данных.

## 3. СВЁРТОЧНАЯ НЕЙРОСЕТЬ

В качестве классификатора, с помощью которого будет проводиться оценка данного метода была выбрана реализация свёрточной нейросети [4]. Её преимуществами является высокая скорость работы за счёт распараллеливания на графическом процессоре с архитектурой CUDA (Compute Unified Device Architecture), а также возможность легко адаптировать её к поставленной задаче.

Идея свёрточной нейросети основывается на трёх основных принципах:

- Локальные поля восприятия
- Разделяемые веса
- Пространственная субдискретизация

Эти принципы реализуются следующим образом. Нейросеть состоит из двух основных типов слоёв: свёрточных и субдискретизирующих.

Каждый нейрон свёрточного слоя соединён лишь с некоторой областью входной карты признаков (либо входного изображения, если этот слой — первый). При этом все нейроны этого слоя имеют одни и те же веса (так реализуется принцип разделяемых весов). Таким образом, карта признаков, полученная на выходе из свёрточного слоя, является, по сути, свёрткой входной карты с некоторым фильтром, коэффициенты которого и подвергаются обучению. Слой может иметь несколько таких фильтров, соответственно на выходе будет получено несколько карт признаков. Обучение может осуществляться методом обратного распространения ошибки или его модификациями, подробнее об этом можно прочитать в работе [5].

Каждый нейрон субдискретизирующего слоя также соединён лишь с некоторой областью входной карты признаков, но эти области обычно не пересекаются и нейроны не имеют весов. Вместо этого в нейроне вычисляется некая функция значения входов, часто используется максимум среди входных значений. В результате получается карта признаков меньшей

размерности, т.к. некоторая область входной карты заменяется одним значением.

Свёрточные и субдискретизирующие слои в свёрточной нейросети чередуются, пока не будет получен набор карт признаков достаточно маленькой размерности, после чего все значения подаются на вход полно связной одно- или двухслойной нейросети.

В результате нейросеть имеет сравнительно мало весов (свёрточный слой имеет число весов, пропорциональное числу обучаемых фильтров, субдискретизирующий слой не имеет весов), что уменьшает занимаемый объём памяти. Благодаря принципу локальных полей восприятия каждый нейрон соединён лишь с некоторой областью входной карты признаков, благодаря чему существенно повышается скорость обучения. Пространственная субдискретизация позволяет уменьшать размерность карт признаков без потери важной информации, а также обеспечивать инвариантность к масштабу классифицируемых изображений.

На вход классификатору подавались цветные изображения размером 32x32 пикселя в формате RGB, нормализованные с помощью алгоритма CLAHE (Contrast Limited Adaptive Histogram Equalization).

## 4. ГЕНЕРАЦИЯ СИНТЕТИЧЕСКИХ ИЗОБРАЖЕНИЙ

### 4.1 Экземпляры фона



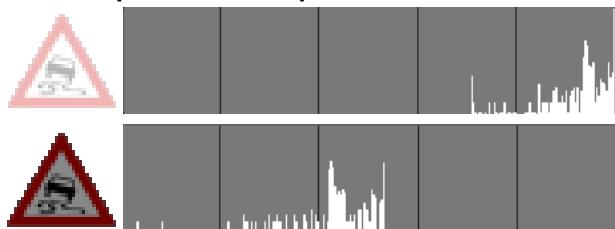
Для получения примеров фона были использованы фотографии разрешения около 1000x1000 пикселей, из которых в случайных местах вырезались образцы 32x32. Фотографии мы брали разнообразные, сделанные в разное время года, в разное время суток и при различных погодных условиях, пейзажные и фотографии города.

### 4.2 Пиктограммы знаков



Пиктограммы знаков были обработаны таким образом, чтобы вся часть изображения вне знака была прозрачной, что было реализовано с помощью альфа-канала на изображениях в формате RGBA. Впоследствии значение альфа-канала используется в качестве маски для наложения на фон и блендинга. Кроме того, были добавлены некоторые пиктограммы, не соответствующие стандарту, но встречающиеся в реальных условиях.

### 4.3 Яркость и контраст



Яркость и контраст изображений меняется за счёт сужения гистограммы яркости пикселей. При этом смещается либо только левая граница («засвеченный» знак), либо только правая (затемнённый) знак). Меняется только яркость знака, т.к. не всегда яркость знака и фона совпадают. Тем самым, мы одновременно и снижаем контраст, и изменяем

яркость изображений. Степень сжатия гистограммы принимает произвольное значение от 0% до 90%.

### 4.4 Изменение размера



Размер пиктограммы изменялся в пределах от 90% до 110%, что приблизительно соответствует точности выделения описывающего квадрата для знаков в базе GTSRB. Но кроме этого, в базе есть изображения, на которых границы выделены не совсем точно, т.е. знаки, которые сильно меньше, чем описанный вокруг них квадрат. Для таких знаков были добавлены примеры (не более 10% от общего числа синтетических изображений) с пиктограммой, уменьшенной в пределах от 60% до 90%.

### 4.5 Поворот



Поворот пиктограммы осуществляется на случайный угол, не превосходящий 30 градусов по часовой или против часовой стрелки. Ограничение на допустимый угол поворота накладывается тем фактом, что некоторые знаки при повороте выглядят одинаково (например, знаки со стрелками).

### 4.6 Положение пиктограммы на изображении

Положение пиктограммы выбирается случайным образом, но ввиду того, что в большинстве случаев знак расположен в центре описанного квадрата, из двух случайно сгенерированных положений пиктограммы берётся среднее значение. Таким образом обеспечивается распределение, приближающееся к нормальному, при сохранении высокой производительности вычислений.

### 4.7 Проективные преобразования



Так как автодорожный знак со стороны проезжающего автомобиля виден обычно под некоторым углом, к генерируемым изображениям необходимо применить проективные преобразования. Ввиду различных условий установки знака и различных его положений относительно дороги и автомобиля, проезжающего по ней, бесполезно пытаться точно посчитать конкретные проективные преобразования, более эффективно попытаться покрыть весь класс таких преобразований (разумеется, не слишком искажающих изображение). Искомое проективное преобразование рассчитывалось как преобразование, ставящее в соответствие вершины двух четырёхугольников. В качестве первого четырёхугольника использовался квадрат, соответствующий знаку, т.е. со сторонами 32x32. Второй четырёхугольник получался из первого смещением вершин на случайное значение, не превосходящее 10% от размеров изображения.

### 4.8 Интерполяция

Важным моментом при проведении различного рода геометрических преобразований является интерполяция. При неаккуратной обработке на границах пиктограммы могут возникать артефакты, вызванные смешиванием

пиктограммы с окружающими её пустыми пикселями. Чтобы этого избежать, ещё перед всеми преобразованиями пиктограмма накладывается на фон. В результате на границах пиктограммы получается своего рода блендинг за счёт смешивания её с фоном. Интерполяция во всех описанных выше преобразованиях использовалась билинейная, как достаточно быстрая. Кроме того, ввиду низкого разрешения изображения кубическая интерполяция не сможет обеспечить существенного прироста в качестве. Мaska прозрачности (альфа-канал) была приведена к формату черно-белого изображения и подвергалась таким же преобразованиям, как и пиктограмма, с использованием также билинейной интерполяции. Такой подход сразу обеспечивает необходимое размытие маски для этапа блендинга.

#### 4.9 Блендинг

Наложение пиктограммы на фон осуществляется следующим образом:

- Мaska переводится в представление с плавающей точкой;
- Мaska размывается гауссовым фильтром (использовался фильтр размера 5x5 с параметром сигма, равным 0.5)
- Матрица пикселей пиктограммы умножается поэлементно на матрицу маски, а матрица фона — на матрицу из единиц минус маска.
- Полученные матрицы складываются

#### 4.10 Точечный блик

В качестве одного из факторов, способных усложнить классификацию знака, было рассмотрено неравномерное освещение. В частности, на некоторых фотографиях наблюдается блик, из-за которого яркость различных частей фотографии существенно меняется и нормализация не может быть применена достаточно эффективно. Чтобы обучить классификатор распознавать и такие примеры входных данных, к части генерируемых синтетических изображений (20%) добавляется блик. Реализуется он следующим образом:



- Изображение переводится в формат YUV
- Выбирается произвольная точка
- Компонента изображения, соответствующая яркости (Y) домножается на гауссиану относительно расстояния пикселя до выбранной точки со случайно выбранным параметром сигма.

#### 4.11 Шум

После всех действий к сгенерированному примеру применяются различные виды искажений, способные приблизить его к примерам реальных данных. Каждое из искажений применяется лишь к 20% от всех примеров.

##### 4.11.1 «Смазанные» фотографии



Так как в большинстве случаев съёмка ведётся из движущегося автомобиля на обычную камеру, не обладающую способностью к высокоскоростной съёмке, то довольно большое количество изображений получаются смазанными. Для моделирования этого явления применяется фильтр motion, заключающийся в усреднении значений нескольких пикселей в одном направлении. Количество усредняемых пикселей не превосходит 25% от

размера знака, направление может принимать значение в пределах до 45% по часовой или против часовой стрелки

##### 4.11.2 Фильтр Гаусса

Применяется фильтр Гаусса размером до 15% от размера изображения.



##### 4.11.3 Морфологическое открытие/закрытие

Осуществляется над компонентой яркости для изображения в формате YUV, моделирует эффект мокрого объектива. В качестве фильтра используется дисковый фильтр радиуса 1 (т.е. 3x3).



##### 4.11.4 Пуассоновский и соль-перец шум

Применяется для повышения устойчивости обученного классификатора к мелким дефектам изображения.



##### 4.11.5 Уменьшение и увеличение изображения



Если при съёмке знак был достаточно удалён, то на фотографии он может занимать область, меньше чем 32x32, но несмотря на такое низкое разрешение, всё равно читаться (если не было других помех). Чтобы научить классификатор распознавать такие знаки, часть примеров уменьшаются до размера 16x16, а затем увеличиваются вновь.

##### 4.11.6 «Битый пиксель»

Черный или белый квадрат накладывается в случайном месте изображения. Позволяет промоделировать различные помехи на знаке, а также избежать чрезмерной важности того или иного локального участка для классификации примера в целом.



Кроме всего перечисленного, на все примеры накладывается слабый гауссовский шум (сигма = 0.0002), иначе на части примеров могут возникнуть области в точности одного цвета.

### 5. ОЦЕНКА ЭФФЕКТИВНОСТИ МЕТОДА И ЕГО ЧАСТЕЙ

#### 5.1 Конфигурация сети

Используемая сеть состояла из 7 слоёв:

- Свёрточный с 64 фильтрами размера 5x5;
- Субдискретизирующий типа max-pooling (выбор максимального значения среди входов) с фильтром размера 3x3 и шагом 2;
- Свёрточный с 1 фильтром 5x5 на каждый канал;
- Субдискретизирующий типа max-pooling с фильтром размера 3x3 и шагом 2;
- Локальный слой с фильтрами 3x3;
- Локальный слой с фильтрами 3x3;
- Полносвязный слой из 43 нейронов (по числу классов).

Ранее не был рассмотрен локальный слой — слой нейросети, в котором каждый нейрон соединён с некоторой областью (в данном случае — квадрат 3x3)

входной карты признаков. Главное отличие от свёрточного слоя заключается в том, что веса эти связей — не разделяемые, и у каждого нейрона они свои.

Как уже указывалось выше, на вход нейросети подавались RGB изображения 32x32, нормализованные с помощью алгоритма CLAHE, на выходе сеть определяла его принадлежность к одному из классов.

Время обучения сети на выборке размером 240 000 изображений составляет 15-20 минут, время классификации — 4 000 примеров в секунду. Такая производительность может позволить не только проводить классификацию в режиме онлайн, но и использовать сеть в качестве детектора знаков.

## 5.2 Эксперименты с синтетическими данными

За счёт применения описанной выше конфигурации нейросети при обучении и тестировании данных из базы GTSRB [2] была достигнута точность 96.8%. Это не очень хороший результат по сравнению с результатами победителей соревнования GTSRB Challenge [3], но преимуществами данного подхода является сравнительная простота реализации и высокая скорость работы.

При обучении нейросети на выборке из 240 000 синтетических изображений, сгенерированных по описанному выше алгоритму, была достигнута точность классификации 97.1%. Тестирование проводилось на тестовой выборке базы GTSRB. Генерация такой выборки заняла 3.5 часа.

Генерация синтетики без использования нестандартных знаков (см. п. 4.2) привела к снижению точности на 0.2%;

Отсутствие изменения яркости изображений (см п.4.3) снизило точность на 10%;

Отсутствие колебаний размера снизило точность на 3%;

Отсутствие поворота пиктограмм — на 5.2%;

Отсутствие перспективных преобразований — на 1.8%

Без добавления точечного блика (п.4.10) сеть работает на 0,2% хуже.

Отсутствие шумов также снижает производительность, на 2.3%.

Таким образом, применение всех преобразований оправдано и приводит к повышению качества классификации.

## 6. ЗАКЛЮЧЕНИЕ

В результате работы была продемонстрирована как эффективность применения свёрточных нейросетей для классификации автодорожных знаков, так и продемонстрированы основные идеи генерации синтетических изображений для обучения классификатора.

В планах:

- Протестировать алгоритм на других базах данных;
- Применить свёрточную нейросеть не только для классификации, но и для детекции знаков.
- Создать более сложную структуру из свёрточных нейросетей, как, например, в работе [1]

## 7. ССЫЛКИ

[1] *Dan Ciresan, Ueli Meier, Jonathan Masci and Jürgen Schmidhuber , “A Committee of Neural Networks for Traffic Sign Classification”, IJCNN, 2011.*

[2] *Institut für Neuroinformatik, Dataset for The German Traffic Sign Recognition Benchmark. Available: <http://benchmark.ini.rub.de/>*

[3] *Institut für Neuroinformatik, The German Traffic Sign Recognition Benchmark Challenge. Available: <http://benchmark.ini.rub.de/>*

[4] *Alex Krizhevsky, “Convolutional Deep Belief Networks on CIFAR-10”. Available: <http://code.google.com/p/cuda-convnet/>*

[5] *Y. LeCun, L. Bottou, Y. Bengio, P. Haffner. “Gradient-Based Learning Applied to Document Recognition.” Proceedings of the IEEE, 86(11):2278-2324, November 1998.*

[6] *Paolo Medici, Claudio Caraffi, Elena Cardarelli, Pier Paolo Porta, Guido Ghisio. “Real Time Road Signs Classification.” Proceedings of the 2008 IEEE International Conference on Vehicular Electronics and Safety. Columbus, OH, USA. September 22-24, 2008. pp. 253-258.*

## Об авторах

Моисеев Борис — студент ВМК МГУ им М.В.Ломоносова.

Чигорин Александр — аспирант ВМК МГУ им М.В.Ломоносова.

# Многокадровый оптический поток на основе траекторий

Михаил Синдеев<sup>1,2</sup>, Антон Конушин<sup>2</sup>, Карстен Ротер<sup>3</sup>

<sup>1</sup> Институт прикладной математики им. М.В. Келдыша РАН, Москва, Россия

<sup>2</sup> Факультет вычислительной математики и кибернетики, МГУ имени М.В. Ломоносова, Москва, Россия

<sup>3</sup>Microsoft Research Cambridge

{msindeev, ktosh}@graphics.cs.msu.ru, carrot@microsoft.com

## Аннотация

В этой статье предлагается новый алгоритм вычисления оптического потока, использующего соседние кадры в качестве вспомогательных для точного анализа перекрытий. Это позволяет оценить движение в областях, которые видимы лишь в одном из двух последовательных кадров. Кроме того, может быть смоделировано нелинейное движение пикселов между двумя кадрами для более точной генерации промежуточных кадров в задаче интерполяции.

**Ключевые слова:** оптический поток, траектории, перекрытия, карта видимости.

## 1. ВВЕДЕНИЕ

Пусть требуется построить карту соответствия между двумя изображениями, являющимися соседними кадрами видеопоследовательности  $I_0$  и  $I_1$ . С учетом погрешностей регистрации изображений (шум CCD-матрицы, перекрытия, артефакты дискретизации), соответствие между изображениями можно записать как

$$I_0(x, y) \approx I_1(x + V_x(x, y), y + V_y(x, y)), \quad (1)$$

где  $V_x$ ,  $V_y$  – вертикальная и горизонтальная компоненты видимого движения пикселов между кадрами, называемого оптическим потоком  $V$ . Для нецелых значений сдвига используется билинейная интерполяция. Поскольку для каждого пикселя  $(x, y)$  изображения  $I_0$  можно найти много пикселов в изображении  $I_1$  с совпадающим или близким цветом, решение для  $V$  не единственno. Поэтому задачу нахождения оптического потока обычно формулируют как минимизацию функционала, состоящего из члена данных  $E_D$  и члена гладкости  $E_S$ , умноженного на коэффициент регуляризации  $\lambda > 0$  (см. [4]).

$$E(V) = E_D(V) + \lambda E_S(V), \quad (2)$$

$$E_D(V) = \iint (I_0(x, y) - I_1(x + V_x, y + V_y))^2 dx dy \quad (3)$$

$$E_S(V) = \iint \|\nabla V_x\|^2 + \|\nabla V_y\|^2 dx dy, \quad (4)$$

$$V = \arg \min_V E(V) \quad (5)$$

Возможны и другие способы регуляризации, использующие неквадратичные нормы для данных слагаемых [10], например:

$$E_D(V) = \iint \psi(I_0(x, y) - I_1(x + V_x, y + V_y)) dx dy \quad (6)$$

$$E_S(V) = \iint \psi(\nabla V) dx dy, \quad (7)$$

$$\psi(z) = \sqrt{z^2 + \sigma^2} \text{ для малого } \sigma. \quad (8)$$

Основная цель такой регуляризации – учесть перекрытия и разрывы в потоке.

Однако робастные нормы лишь примерно интерполируют поток в области перекрытий, используя значения из соседних пикселов. Выбор подходящей нормы в слагаемом гладкости позволяет добиться правдоподобного профиля интерполяции для определенных классов объектов (например, для объектов с гладким контуром или объектов, выделяющихся по цвету – в этом случае норма также зависит от изображения  $I_0$ ).

Некоторые методы поиска перекрытий помимо прямого потока вычисляют обратный поток  $V'$  от  $I_1$  к  $I_0$  и сравнивают их, помечая пиксели  $|V + V'| > \delta$  как перекрытия для некоторого фиксированного порога  $\delta$  (обратный поток берется со знаком «+», т.к. он в сумме с прямым потоком в идеале должен давать 0), либо используют фиксированный штраф для перекрытий пикселов [13].

Иногда принимается во внимание пространственная разреженность карты перекрытий, например, путем фильтрации полученной бинарной карты [13].

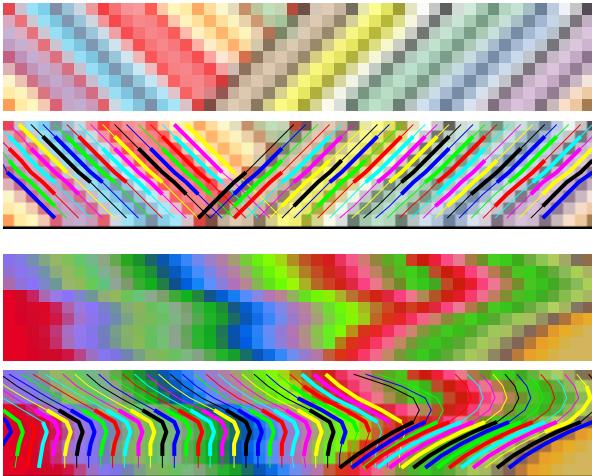
Однако в силу неоднозначности потока возможны случаи, когда при похожих цветах объектов в изображениях слагаемое данных получается слабым и поток  $V$ ,  $V'$  различаются лишь из-за регуляризации, которая оптимизируется независимо для прямого и обратного потока. Возможна формулировка совместной минимизации с общей регуляризацией, но она не улучшает ситуацию, т.к. результирующий функционал имеет много локальных минимумов.

Другая проблема – выбор порога отсечения перекрытий, т.к. количество перекрытий пикселов может существенно различаться в разных кадрах. Одним из решений является добавление в задачу неизвестной бинарной маски перекрытий и попиксельное умножение слагаемого данных  $E_D$  на эту маску. Такой подход приводит к тому, что глобальный минимум  $E$  достигается при отметке всех пикселов как перекрытий. Попытка выбора оптимального штрафа за избыток перекрытий равносильна проблеме поиска порога  $\delta$ , т.е. не может быть решена сразу для всех кадров.

В данной статье рассматривается возможность уточнения слагаемого данных за счет использования соседних кадров с явным вычислением карты видимости.<sup>3</sup> Наш подход основан на представлении многокадрового потока в виде траекторий [12] и идее моделирования перекрытий с помощью сортирующего суммирования, применяемой в алгоритмах многоракурсного стереосопоставления [9], [5].

## 2. ПРЕДЛАГАЕМЫЙ ПОДХОД

Вместо пространственной разреженности перекрытий, применяемой в двухкадровых алгоритмах оптического потока [13], сосредоточимся на их временной разреженности. Временная разреженность означает, что каждая точка сцены попадает в перекрытие не слишком часто. Это условие является более естественным, чем пространственная разреженность в отдельно взятом кадре, т.к. нельзя заранее сказать, сколько точек в кадре перекрыто, в то время как частоту перекрытий во времени можно примерно оценить снизу.



**Рисунок 1.** Искусственные примеры. Исходные одномерные видеопоследовательности ( $T = 9$  кадров, по вертикали – ось времени) и найденные траектории. Толстые части траекторий – диапазоны видимости.

Предположим, что каждая точка в изображении  $I_0$  видима хотя бы на протяжении 5 кадров. Тогда, рассматривая 9 последовательных изображений  $I_{-4}, \dots, I_4$  можно ввести переменную, отвечающую за перекрытие  $\rho \in \{0, 1, 2, 3, 4\}$ .

При этом  $\rho = 0$  означает, что данный пиксель является видимым в кадрах  $I_{-4}, \dots, I_0$ ,  $\rho = 1$  – в кадрах  $I_{-3}, \dots, I_1$  и т.д.,  $\rho = 4$  – в кадрах  $I_0, \dots, I_4$ . Обратим внимание, что в такой формулировке пиксель всегда виден в центральном кадре  $I_0$ . Карта видимости  $\rho$  по координатам  $(x, y)$  соответствует именно этому кадру (в то время как координаты пикселя в других кадрах зависят от потока между  $I_0$  и данным кадром).

В общем случае мы рассматриваем произвольное нечетное число кадров  $T \geq 3$ . Для обозначения диапазона индексов введем значение  $K = \frac{T-1}{2}$ . Пиксель является видимым по

меньшей мере в  $K+1$  изображении из  $T$  изображений  $I_{-K}, \dots, I_K$ . Карта видимости:  $\rho \in \{0, \dots, K\}$ , пиксель является видимым в кадрах  $\rho - K, \dots, K$ . Пример траекторного потока приведен на рис. 1.

Заметим, что в такой формулировке не ставится вопрос о наличии или отсутствии перекрытия в каком-либо пикселе какого-либо кадра, поэтому не придется выбирать функцию штрафа за чрезмерные перекрытия. Мы лишь выбираем набор из  $K+1$  последовательных кадров, в которых пиксель точно является видимым (видимость в остальных  $K$  кадрах может быть произвольной).

Для работы с пикселями в такой формулировке нам потребуется  $T - 1$  (или, что то же самое,  $2K$ ) двухмерных оптических потоков – из кадра  $I_0$  в каждый из остальных кадров. Обозначим эти потоки как  $V_{-K}, \dots, V_{-1}, V_1, \dots, V_K$ . Если бы они были последовательными ( $I_1 \rightarrow I_2, I_2 \rightarrow I_3$  и т.д.), для вычисления позиции пикселя требовалось бы интегрирование нескольких потоков, что сильно усложняет формулировку и снижает точность из-за погрешностей интерполяции. Поэтому поток  $V_k$  отвечает за трансформацию между кадрами  $I_0 \rightarrow I_k, k = -K, \dots, -1, 1, \dots, K$ .

Выходным результатом алгоритма считается поток  $V_1$ , и, если нужно, обратный поток в предыдущий кадр  $V_{-1}$ .

Остальные потоки являются лишь вспомогательными для уточнения перекрытий. Также для выходных потоков можно вычислить бинарную маску перекрытий как  $\{\rho = 0\}$  и  $\{\rho = K\}$  соответственно (т.е. случаи, когда не виден кадр  $I_1$  и  $I_{-1}$  соответственно). Квадратными скобками обозначена индикаторная функция.

Т.к. потоки вычисляются относительно кадра  $I_0$ , их можно считать одним «траекторным» потоком, где каждому пикселу сопоставлен  $4K$ -мерный (или, что то же самое,  $2(T - 1)$ -мерный) вектор потока

$$\vec{V} = (V_{-K,x}, V_{-K,y}, \dots, V_{-1,x}, V_{-1,y}, V_{1,x}, V_{1,y}, \dots, V_{K,x}, V_{K,y}) \quad (9)$$

Такое векторное представление позволяет использовать любые известные функционалы гладкости, определенные для двухмерного потока, которые могут быть сформулированы в терминах векторной алгебры, а также многие алгоритмы минимизации данных функционалов. Исключением будут алгоритмы, явно использующие факт двухмерности, а также алгоритмы, несовместимые с членом данных  $E_D$ , который будет сформулирован в следующем разделе (однако их часто можно адаптировать для данной задачи, либо использовать методы раздельной оптимизации, используя другой алгоритм для члена данных). Кроме того, стохастические и переборные алгоритмы могут потерять свою эффективность из-за увеличения пространства поиска.

## 2.1 Функционал энергии

Задача поиска траекторного потока может быть сформулирована как задача минимизации функционала энергии

$$E(V, \rho) = E_D(V, \rho) + \lambda E_S(V) + \mu E_T(V) + \nu E_\rho(\rho) \quad (10)$$

где  $E_D$  – член данных,  $E_S$  – член пространственной гладкости потока,  $E_T$  – член временной гладкости потока,  $E_\rho$  – член гладкости карты видимости,  $\lambda, \mu, \nu$  – веса слагаемых.

$$E_D(V, \rho) = \frac{1}{K} \sum_x \sum_y \sum_{\substack{t=0 \\ t=\rho(x,y)-K}}^{\rho(x,y)} (I_0(x, y) - I_t(x + V_{t,x}, y + V_{t,y}))^2 \quad (11)$$

В этом члене не используется робастная норма (8) или аналогичная, как в формуле (6), т.к. за перекрытия явно определяются значением  $\rho$ , и цвета не перекрытых пикселов должны совпадать с высокой точностью.

Член пространственной гладкости соответствует случаю двухкадрового потока (7) с некоторыми модификациями:

$$E_S(V) = \frac{1}{T-1} \sum_x \sum_y \left[ \sum_{\substack{t=-K, \\ t \neq 0}}^K \left( \frac{\nabla V}{|t|} \right)^2 \right]^\gamma \quad (12)$$

В качестве робастной нормы взята функция, инвариантная к повороту. Степень  $\gamma = 0,5$  соответствует робастной норме  $L_1$ , инвариантной к повороту, которая определяется как

$$\psi(V) = \sqrt{\left( \frac{\partial V_x}{\partial x} \right)^2 + \left( \frac{\partial V_x}{\partial y} \right)^2 + \left( \frac{\partial V_y}{\partial x} \right)^2 + \left( \frac{\partial V_y}{\partial y} \right)^2}, \quad (13)$$

причем в нашем случае суммирование по  $t$  также производится под знаком радикала. Это означает, что в случае разрывности решения разрыв потока в некоторой точке происходит одновременно в горизонтальной и вертикальной

компонентах потока и сразу во всех кадрах. Таким образом, траектории в двух соседних пикселях не могут разойтись, а через несколько кадров слиться в единый объект.

На практике мы используем значение  $\gamma = 0.45$ , как рекомендовано в статье [10], порождающее невыпуклую норму, которая «поощряет» более четкие края в случае разрывного потока.

Деление на  $|t|$  в формуле (12) уравнивает разброс значений по разным кадрам, т.к. он возрастает при удалении от центрального кадра.

Энергия временной гладкости использует вторую производную потока по времени, т.е. поощряет траектории, близкие к линейным:

$$E_T(V) = \frac{1}{T-2} \sum_x \sum_y \sum_{t=-K+1}^{K-1} (V_{t-1} - 2V_t + V_{t+1})^2, \quad (14)$$

при этом полагается  $V_0 = 0$ .

Коэффициенты  $1/K, 1/(T-1), 1/(T-2)$  в формулах (11), (12), (14) соответствуют количеству слагаемых на пикселях, зависящему от числа рассматриваемых кадров  $T$ , что упрощает подгонку коэффициентов  $\lambda, \mu, \nu$  при изменении числа кадров.

Функционал гладкости карты видимости также является квадратичным:

$$E_\rho(\rho) = \sum_x \sum_y (\nabla \rho(x, y))^2 \quad (15)$$

При этом градиент означает конечную разность ввиду дискретности значений  $\rho$ . Выбор квадратичной функции штрафа обусловлен структурой перекрытий в видео: движущийся объект оставляет за собой «след» из перекрытий, каждое из которых сдвинуто на 1 кадр относительно предыдущего. Скачки в карте видимости, превышающие 1 кадр, чаще всего соответствуют неправильно найденным перекрытиям. Надежными перекрытиями являются такие, которые последовательно возникают по пути движения объекта, поэтому имеет смысл искать перекрытия именно с такой структурой.

Также можно ввести два дополнительных члена, связанных с картой видимости.

Первый член запрещает выход видимой части траектории за края изображения, т.к. обращает энергию в бесконечность. Таким образом, данная ситуация рассматривается как обычное перекрытие, в отличие от алгоритмов двухкадрового потока, где такие случаи приходится обрабатывать отдельно, чаще всего обнулением производных для векторов потока, выходящих за края изображения.

Второй дополнительный член поощряет, с очень маленьким весом, близость  $\rho$  к значению  $K/2$ , т.е. симметрию диапазона видимости. Такая регуляризация улучшает маску видимости для выходного потока  $V_I$ , т.к. иначе в простых случаях при отсутствии перекрытий может быть выбран диапазон видимости  $-K, \dots, 0$  и выходной поток будет менее информативным.

### 3. МИНИМИЗАЦИЯ

В данном разделе приводится предлагаемый алгоритм минимизации полного функционала энергии (10). Важным аспектом является совместная минимизация по потоку и карте видимости. При поочередной минимизации возможны локальные минимумы [11].

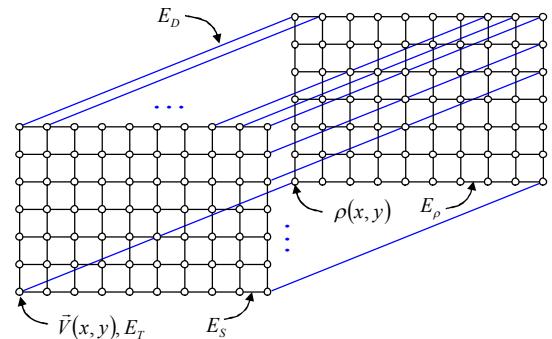


Рисунок 2. Структура графа, позволяющая описать функционал энергии (10) в виде суммы унарных и парных потенциалов на двух двухмерных сетках (для  $\bar{V}$  и  $\rho$ ).

Идея алгоритма заключается в использовании метода QPBO [8], минимизирующего энергию для бинарных задач, и алгоритма слияния [6], использующего QPBO для последовательной склейки решений-кандидатов для небинарных задач. При этом сама задача склейки является бинарной.

Будем осуществлять минимизацию на двухмерной сетке, соответствующей изображению  $I_0$ . Каждому узлу сетки соответствует вектор потока  $\bar{V}$  (см. обозначение (9)) и значение  $\rho$ . Можно строить решения-кандидаты, состоящие из пар  $(\bar{V}, \rho)$ , но можно сделать их независимыми, разместив их на двух сетках (рис. 2).

Первая сетка содержит пространственные ребра, отвечающие за слагаемое  $E_S$ . Узлы содержат унарный потенциал  $E_T$ . Ребра второй сетки отвечают за слагаемое  $E_\rho$ . Ребра, соединяющие соответственные узлы первой и второй сеток, отвечают за слагаемое  $E_D$ .

Инвариантные к повороту нормы не могут быть заданы с помощью ребер двухмерной сетки, поэтому используется аппроксимация – каждая вершина соединяется не с 4, с 16 ближайшими [3].

#### 3.1 Начальное приближение

В качестве начального приближения вычислим двухкадровые потоки между последовательными кадрами в прямом и обратном направлении от центрального кадра:  $I_0 \rightarrow I_1, I_1 \rightarrow I_2, \dots; I_0 \rightarrow I_{-1}, I_{-1} \rightarrow I_{-2}, \dots$ . Затем выполним рекурсивную трансформацию этих потоков, чтобы получить потоки между центральным кадром и остальными кадрами. Для вычисления двухкадровых потоков использовалась библиотека texOpticalFlow [7].

#### 3.2 Решения-кандидаты

Будем генерировать решения-кандидаты, используя несколько различных эвристик, и сливать их с текущим решением методом QPBO. Алгоритма слияния [6] гарантирует невозрастание энергии, что снижает требования к решениям-кандидатам.

Решения-кандидаты для траекторного потока  $\bar{V}$ :

1. Всевозможные сдвиги всей сетки на  $\pm 1$  пиксель по каждой оси (т.е. значение потока  $\bar{V}$  копируется из соседнего пикселя, компоненты вектора при этом не меняются)
2. Выбор траектории  $\bar{V}$  в случайном пикселе  $(x, y)$  и использование его в качестве константного кандидата. Т.к. сцены часто состоят из нескольких объектов,

вместо равномерного распределения для  $(x, y)$  сначала осуществляется сегментация потока методом «*средних*», затем случайно выбирается сегмент, а в нем – случайный пиксель. Это нужно, чтобы объекты большей площади не имели преимущества.

3. Двухмерное гауссово размытие потока  $\vec{V}$
4. Один шаг градиентного спуска для слагаемых, отвечающих за данные на двухмерной сетке (т.е. без учета пространственной гладкости):  $E_D(V, \rho) + \mu E_T(V)$
5. Попиксельная минимизация  $E_D(V, \rho) + \mu E_T(V)$  по  $\vec{V}$  методом динамического программирования. Используется ускоренный приближенный вариант, основанный на идее алгоритма PatchMatch [2].

Как видно, часть решений-кандидатов соответствует копированию «удачных» траекторий  $\vec{V}$  из одних пикселей в другие, другая часть генерирует новые решения в окрестности текущего решения.

Кандидаты для карты видимости являются константами  $\rho = 0, \dots, K$  (также были опробованы сдвиги и увеличение/уменьшение  $\rho$  на 1, но эти варианты не дают существенного улучшения по сравнению с использованием констант).

Кандидаты для всех переменных  $(\vec{V}, \rho)$  строятся в виде декартина произведения описанных кандидатов (для каждого кандидата  $\vec{V}$  пробуем каждый кандидат  $\rho$ ). Т.к. кандидаты зависят от текущего решения, цикл оптимизации надо повторять до тех пор, пока полная энергия не перестанет уменьшаться.

### 3.3 Иерархический подход

Аналогично двухмерным методам оптического потока, возможно применение иерархического подхода с уменьшением размера всех кадров до небольшого размера (например, 40x30 пикселей), а затем поэтапным увеличением до исходного размера ([10], [7]). При этом, после каждого уровня иерархии можно выполнить дополнительное слияние: решение с предыдущего уровня плюс двухмерный поток для данного разрешения, преобразованный в траектории (как описано в разделе 3.1).

## 4. РЕЗУЛЬТАТЫ

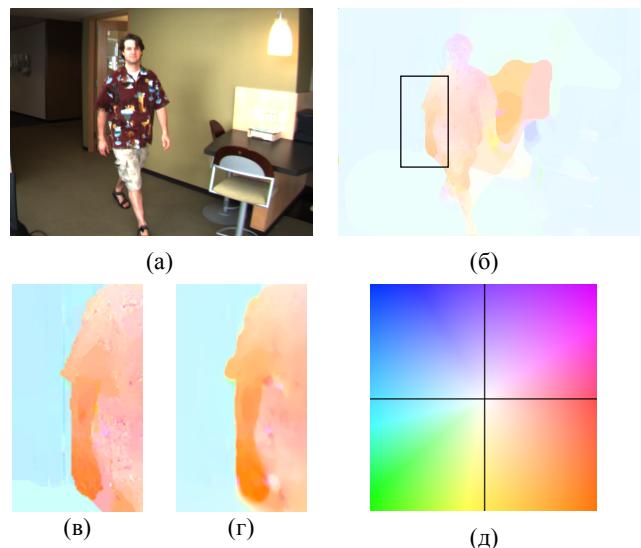
Пример результата работы алгоритма приведен на рис. 3. По сравнению с результатом алгоритма [7] контур объекта более точный и не содержит гладких переходов между объектом и фоном, которые видны в виде белой окантовки на рис. 3 (г). Это достигается за счет построения карты видимости и совместной оптимизации видимости и потока.

## 5. ЗАКЛЮЧЕНИЕ

В статье предложен алгоритм вычисления оптического потока на основе траекторий, дающий стабильный результат при наличии перекрытий за счет использования нескольких последовательных кадров. В дальнейшем планируется провести подробное экспериментальное сравнение метода с аналогами и получить численные оценки

## 6. БЛАГОДАРНОСТИ

Работа выполнена при поддержке проекта МРЛ-2010-050 с компанией Microsoft Research.



**Рисунок 3.** Пример результата. (а) изображение  $I_0$  из тестовой базы [1],  $T = 7$  кадров, (б) поток  $V_1$ , (в) увеличенный фрагмент, выделенный рамкой в (б), (г) результат алгоритма [7], (д) цветовая карта обозначения потока, предложенная в статье [1].

## 7. ССЫЛКИ

- [1] Baker S., Scharstein D., Lewis J., Roth S., Black M., Szeliski R., *A database and evaluation methodology for optical flow*, IJCV, 92(1):1–31, 2011.
- [2] Barnes C., Shechtman E., Finkelstein A., Goldman D., *PatchMatch: a randomized correspondence algorithm for structural image editing*, SIGGRAPH 2009
- [3] Boykov Y., Kolmogorov V., *Computing geodesics and minimal surfaces via graph cuts*, ICCV 2003
- [4] Horn B., Schunck B., *Determining optical flow*, Artificial Intelligence, vol 17, pp 185–203, 1981
- [5] Kang S., Szeliski R., *Extracting view-dependent depth maps from a collection of images*, International Journal of Computer Vision (IJCV), vol. 58, number 2, pp. 139–163, 2004
- [6] Lempitsky V., Roth S., Rother C., *FusionFlow: discrete-continuous optimization for optical flow estimation*, CVPR 2008
- [7] Liu C., *Beyond pixels: exploring new representations and applications for motion analysis*, Doctoral Thesis. Massachusetts Institute of Technology, 2009
- [8] Rother C., Kolmogorov V., Lempitsky V., Szummer M., *Optimizing binary MRFs via extended roof duality*, CVPR 2007
- [9] Satoh K., Ohta Y., *Occlusion detectable stereo – systematic comparison of detection algorithms*, ICPR 1996
- [10] Sun D., Roth S., Black M., *Secrets of optical flow estimation and their principles*, CVPR 2010
- [11] Sun D., Sudderth E., Black M., *Layered segmentation and optical flow estimation over time*, to appear: CVPR 2012
- [12] Volz S., Bruhn A., Valgaerts L., Zimmer H., *Modeling temporal coherence for optical flow*, ICCV 2011
- [13] Xiao J., Cheng H., Sawhney H., Rao C., Isnardi M., *Bilateral filtering-based optical flow estimation with occlusion detection*, ECCV 2006

# Реконструкция траектории движения и формы объекта с использованием инфракрасных маркеров и стереопары

Рамиз Зейналов, Антон Конушин

Факультет вычислительной математики и кибернетики

Московский государственный университет имени М.В.Ломоносова, Москва, Россия

{rzejnalov, ktosh}@graphics.cs.msu.ru

## Аннотация

В предыдущей работе был описан метод отслеживания объекта в пространстве с использованием инфракрасных маркеров. В этой работе описываются улучшения, проделанные в алгоритмах отслеживания объекта, а также описываются алгоритмы реконструкции формы движущегося объекта. Кроме того, в предыдущей работе было отмечено, что существует проблема наложения маркеров (“marker swapping”). В этой работе приводится обоснование отсутствия этой проблемы в рамках решаемой задачи.

**Keywords:** отслеживание, сопоставление проекций, реконструкция геометрии, инфракрасные маркеры, эпиполярные ограничения.

## 1. ВВЕДЕНИЕ

Иногда возникают практические задачи, связанные с необходимостью отслеживания перемещения объекта в пространстве. При этом не всегда имеется возможность ориентироваться на текстуру объекта – например, объект может иметь однородную окраску или его материал может быть глянцевым, что приводит к тому, что методы, основанные на текстуре объекта (например, методы оптического потока [5, 7, 9] и использование дескрипторов особых точек [1]), становятся неприменимыми. В таких случаях нередко используют контрастные маркеры, которые устанавливаются на объект исследований. В данной работе речь идёт об использовании инфракрасных маркеров (диодов, которые излучают в инфракрасном диапазоне) совместно с использованием стереопары, которая снимает в инфракрасном диапазоне. Таким образом, в качестве исходных данных (после распознавания маркеров, которое тривиально) выступают наборы проекций точек для каждой камеры во все моменты времени.

Для построения траекторий отдельных точек необходимо знать межкадровые соответствия проекций точек. Изначально этой информации нет, т.к. в исходных данных на каждом кадре имеется множество проекций маркеров без уникальных идентификаторов, которые позволили бы различать проекции точек между собой. Подробнее о решении задач, связанных с отслеживанием проекций точек, написано в секции 4.

Проекции точек, по сути, задают точечные особенности объекта. Однако здесь имеется принципиальная проблема – так как маркеры между собой не различимы, не различимы и их проекции – их изображения, которые снимаются камерами. А значит, неизвестны и межвидовые соответствия маркеров. Поэтому перед выполнением триангуляции [2] эти соответствия нужно каким-либо образом восстановить. В этой работе в секции 5

рассказывается об алгоритмах, которые позволяют восстановить эти соответствия.

Когда уже получены межвидовые соответствия точек, можно выполнить триангуляцию. Это позволит восстановить взаимное расположение точек объекта во все моменты времени. Однако ни в один момент времени обе камеры не наблюдают сразу все маркеры объекта. Поэтому на каждом кадре можно восстановить лишь отдельные фрагменты всего объекта. В этой работе рассказывается об алгоритмах, которые позволяют восстановить геометрию всего объекта. Этому посвящена секция 6.

Для апробации построенных алгоритмов был произведён ряд измерительных экспериментов с целью доказательства состоятельности предложенных алгоритмов, а также оценки точности производимых измерений. Проводились как статические эксперименты, основанные на измерении расстояний, так и динамические, основанные на измерении динамических характеристик движения объектов. Описания экспериментов и их результатов представлены в секции 7.

Заключение и будущие планы развития алгоритмов описаны в секции 8.

## 2. ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ

Для отслеживания движущихся объектов на практике часто используются дескрипторы особых точек, такие как SIFT и SURF [1], которые позволяют найти особые точки на объекте [2, 9]. Для отслеживания небольшого количества объектов на статической сцене иногда используется оптический поток для поиска движущихся объектов [5, 7, 9]. Для этого берутся соседние кадры и вычисляется их разность. Те регионы, в которых эта разность больше определённого порога, считаются регионами движения, то есть регионами, в которых движется объект интереса. Просматривая попарно все кадры последовательности, можно получить проекции траекторий движения объекта на камеры. В нашей задаче оптический поток бесполезен, так как при его использовании не будут найдены межвидовые соответствия маркеров.

В некоторых задачах трекинга требуется отслеживать перемещение объекта заранее известной формы [6]. В таких случаях нередко этот объект помечен маркерами, что позволяет эффективнее отслеживать его на изображениях. Когда форма отслеживаемого объекта известна, можно использовать свойства, связанные с расположением маркеров, и легче отличать один маркер от другого. Это т.н. техника Motion Capture [8, 10].

Такие подходы не пригодны для решения поставленной задачи, потому что форма отслеживаемого объекта заранее неизвестна и должна быть получена в процессе решения задачи.

### 3. ПРЕДЛОЖЕННОЕ РЕШЕНИЕ

Предложенное решение задачи состоит из набора алгоритмов, каждый из которых решает отдельную подзадачу. Для получения результата рассматриваемые алгоритмы последовательно применяются к исходным данным.

Рассматриваемая система состоит из стереопары (двух синхронизированных камер, которые снимают в инфракрасном диапазоне) и отслеживаемого объекта, помеченного инфракрасными маркерами (рис. 1).

Итак, в качестве исходных данных всей задачи выступают:

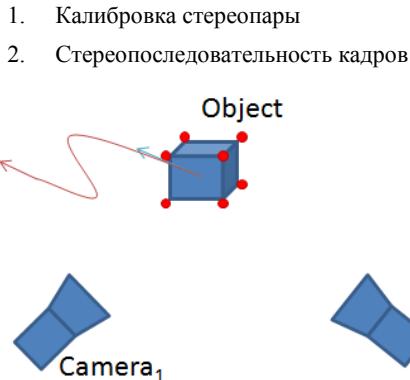


Рисунок 1: Объект и стереопара.

Под калибровкой стереопары понимаются внутренние параметры камер (матрицы внутренних калибровок и наборы коэффициентов дисторсий линз), положения и ориентации в пространстве в некоторой зафиксированной системе координат. Следует отметить, что камеры снимают кадры синхронно, то есть моменты времени съёмки очередного кадра  $t_i$  можно считать одинаковыми, что обеспечивается оборудованием, управляющим камерами.

Под стереопоследовательностью кадров понимаются две синхронизированные последовательности кадров с камер. Каждый кадр содержит набор изображений маркеров и, возможно, шум. Так как маркеры представляют собой инфракрасные диоды, а камеры снимают в инфракрасном диапазоне, кадр, как правило, представляет собой изображение с белыми пятнами на чёрном фоне (рис. 2).



Рисунок 2: Пример кадра.

Сформулируем решаемую задачу. Итак, дано:

- Калибровка стереопары (внутренние параметры камер  $K_c$ , дисторсии линз  $D_c$ , их положения  $T_c$  и ориентации  $R_c$ ):

$$C_c = \langle K_c; D_c; R_c; T_c \rangle, \quad c = 1, 2 \quad (1)$$

- Две последовательности кадров (по последовательности на камеру), состоящих из множества проекций точек ( $c$  – номер камеры,  $n_i^c$  – количество проекций  $q^{c,i}_j$  на кадре  $i$  камеры  $c$ ,  $l_{ij}$  – номер  $j$ -ой точки на  $i$ -ом кадре,  $F_i^c$  – кадр (2),  $S^c$  – последовательность кадров (3),  $N$  – количество кадров):

$$F_i^c = \{p_{l_{ij}}^{c,i}, j = 1..n_i^c\} \quad (2)$$

$$S^c = \{F_i^c\}, \quad i = 1..N, \quad c = 1..2 \quad (3)$$

Требуется:

- Определить межкадровые соответствия точек (4) ( $M^c$  – количество проекций точек, соответствующих разным точкам в пространстве,  $p^{c,i}_j$  – проекции точек, упорядоченные согласно межкадровым соответствиям,  $l(i,j,c)$  – номер проекции  $q$ , соответствующей точке  $j$  на кадре  $i$  для камеры  $c$ )

$$C_F^c : q_{l(i,j,c)}^{c,i} \rightarrow p_j^{c,i} \quad (4)$$

- Определить межвидовые соответствия проекций точек (5) (множество пар индексов  $j_1$  и  $j_2$ , для которых отношение эквивалентности, определяющее соответствие всех проекций одной камеры для одной точки всем проекциям одной точки на другой камере):

$$C_V^{1,2} : p_{j_1}^{1,i} \times p_{j_2}^{2,i} \rightarrow \{True, False\} \quad (5)$$

Задача поиска и распознавания маркеров на подобных изображениях достаточно тривиальна, поэтому не будем на ней подробно останавливаться.

### 4. ОТСЛЕЖИВАНИЕ ПРОЕКЦИЙ

#### 4.1 Модель маркера

Прежде чем описывать алгоритм получения межкадровых соответствий для проекций инфракрасных маркеров рассмотрим модель маркера (рис. 3).

Маркер представляет собой диод шириной  $d$ , высотой  $h$  и с углом рассеивания  $\alpha$  (угол между противоположными боковыми гранями пирамиды, описывающей поток излучаемой диодом энергии).

Проекции маркеров, расположенных на отслеживаемом объекте, теоретически могут накладываться друг на друга на изображениях камер, что может привести к проблемам при отслеживании отдельных маркеров и к ошибкам на этапе поиска межкадровых соответствий маркеров (т.н. проблема marker swapping). На самом деле, можно убедиться в том, что такие ситуации практически невозможны при соблюдении некоторых условий. Для этого необходимо рассмотреть модель маркера (рис. 3), которая адекватно описывает используемые инфракрасные маркеры – светоизлучающие диоды, которые излучают значительную часть энергии в инфракрасную область электромагнитного спектра. Итак, диод излучает свет не во всех направлениях, а на ограниченный угол. Кроме того, если располагать маркеры на значительном расстоянии друг от друга (намного большем, чем размер диода), тогда, в силу ограниченности угла излучения, наложение пятен от маркеров будет невозможно.

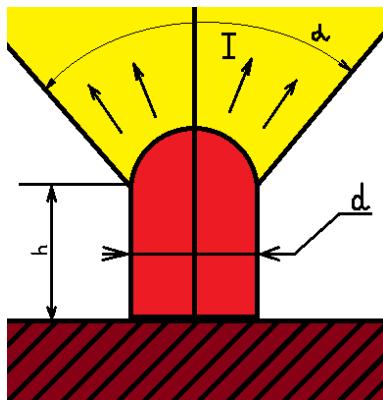


Рисунок 3: Модель маркера.

## 4.2 Алгоритм

На входе дана последовательность кадров (3), которые содержат наборы проекций точек – координат на изображении. Для одной и той же точки в пространстве нет соответствия между её проекциями на разных кадрах, в то время как для оценки траектории эти соответствия необходимы. Таким образом, требуется восстановить эти соответствия.

Проведённые эксперименты показывают, что для получения межкадровых соответствий достаточно использования простого алгоритма, основанного на межкадровом смещении. Ситуации, в которых алгоритм может не сработать, на практике встречаются крайне редко, особенно при малом количестве маркеров.

Кроме того, в силу ограниченности угла излучения маркера и большого расстояния между ними ситуация, когда один маркер накладывается на другой, практически невозможна. В противном случае простой алгоритм, основанный на межкадровом смещении, мог бы давать неверные результаты и нуждался бы в некотором усложнении.

Введём порог, характеризующий величину допустимого смещения проекции точки на изображении между соседними кадрами. Рассмотрим первый кадр последовательности. Каждой проекции, встречающейся на этом кадре, ставим в соответствие уникальный идентификатор – натуральное число (рис. 4а, 4в). Точки  $P_1$ ,  $P_2$  и  $P_3$  получают идентификаторы 1, 2 и 3 соответственно. Затем обрабатываем каждый следующий кадр одним и тем же способом: для каждой проекции на предыдущем кадре ищем ближайшую непомеченную проекцию на текущем кадре. Если расстояние между ними (в плоскости изображения) меньше порога – помечаем проекцию на новом кадре тем же идентификатором, что и на предыдущем кадре.

Каждую непомеченную проекцию на новом кадре помечаем новым уникальным идентификатором (рис. 4б). Если в какой-то момент для проекции на предыдущем кадре не нашлось проекции на текущем кадре, значит, эта точка только что появилась (рис. 4б) – точка  $P_4$ . Если для какой-то точки на предыдущем кадре не удалось найти точку на текущем кадре, значит, эта точка исчезла (рис. 4в, 4г) – точка  $P_2$ . Как было отмечено выше, такой алгоритм работает очень быстро, а ситуации, в которых результат работы может быть неверным, крайне маловероятны.

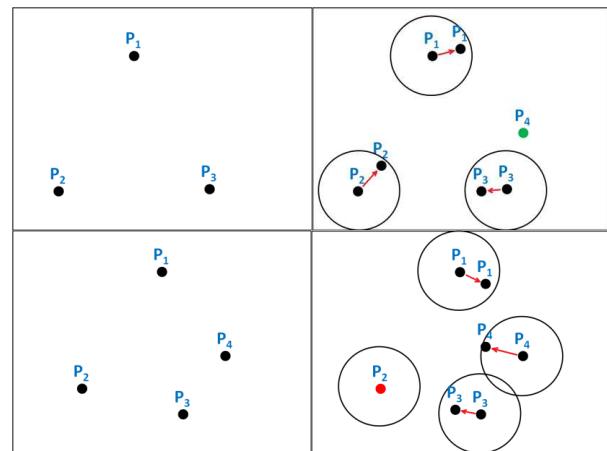


Рисунок 4: Межкадровое сопоставление (а, б, в, г).

## 5. МЕЖВИДОВОЕ СОПОСТАВЛЕНИЕ

Алгоритм делится на два этапа – вычисление возможных межвидовых соответствий для одной пары кадров и вычисление межвидовых соответствий для всей последовательности пар кадров.

### 5.1 Локальное сопоставление

Возможные межвидовые соответствия для кадра вычисляются следующим образом: для каждой точки  $p_j$  на каждой камере выбираются все точки  $\{q_k\}$  на другой камере в окрестности соответствующей эпиполярной линии. Все пары вида  $\langle p_j, q_k \rangle$  добавляются в список возможных соответствий для данной пары кадров. Указанные действия выполняются для всех пар кадров.

$$q^T F p = 0 \quad (6)$$

$$\begin{cases} l_q = Fp \\ l_p = q^T F \end{cases} \quad (7)$$

Для вычисления межвидовых соответствий используются эпиполярные ограничения (6) [3], которые заключаются в том, что для точки в пространстве её проекция  $p$  на одной камере ограничивает положение проекции  $q$  той же самой точки на другой камере эпиполярной линией  $l_q$ , которая задаётся фундаментальной матрицей  $F$ , и наоборот (7).

### 5.2 Глобальное сопоставление

Из-за ошибок на отдельных кадрах могут получиться противоречие между собой множества возможных соответствий. Для того чтобы их избежать, используется схема голосования. Пусть всего разных точек в последовательности кадров одной камеры  $M_1$ , второй камеры –  $M_2$ . Введём матрицу  $MC$ , заполненную нулями. Для каждого соответствия  $\langle p_j^1, p_l^2 \rangle$  каждой пары кадров увеличим на единицу элемент  $MC_{j,l}$  матрицы  $MC$ . В результате в каждом элементе матрицы будет количество голосов за соответствие  $\langle p_j^1, p_l^2 \rangle$ . Чтобы получить множество соответствий для всей последовательности пар кадров будем искать такой набор элементов матрицы  $MC$ , чтобы в каждом столбце и каждой строке было выбрано не более одного элемента, причём сумма этих элементов должна быть максимальной. При этом элементы, значения которых меньше некоторого порога, будем отсеять. Индексы выбранных элементов будут задавать искомые соответствия, сумма элементов – суммарный голос. Для поиска таких индексов воспользуемся алгоритмом

Хопкрофта-Карпа для поиска максимального назначения (паросочетания в двудольном графе [4]).

## 6. РЕКОНСТРУКЦИЯ ГЕОМЕТРИИ

На данном этапе требуется по трёхмерным координатам отдельных точек объекта в разные моменты времени восстановить геометрию объекта и траекторию его движения. Формально: в моменты времени  $\{i\} = \{1..N\}$  камера  $c$  наблюдает точки  $P_i^c = \{p_{i,(k,c)}^c\}$ . Требуется восстановить геометрию объекта  $O = \{P_k\}$ ,  $k = 1..K$  в некоторой системе координат, связанной с объектом.

Идея алгоритма решения этой задачи заключается в том, что множества наблюдаемых точек  $P_i^c$ , для соседних кадров часто бывают одинаковыми – точки появляются и пропадают из видимости камер не на каждом кадре. Это позволяет разбить всё множество моментов времени на непрерывные последовательные промежутки  $R_i = (t_i^l..t_i^u)$ , на которых множества точек  $Q_i$  одинаковы. Такие промежутки будем называть *диапазонами*. Для диапазонов  $R_x$  и  $R_y$ , для которых  $\text{card}(Q_x \cap Q_y) \geq 3$ , можно пересчитать координаты точек из  $Q_x \cup Q_y$ , которых нет в одном из диапазонов. Таким образом диапазоны  $R_x$  и  $R_y$  объединяются в диапазон  $R_z = \text{join}(R_x, R_y)$ . При этом можно объединить и геометрии фрагментов объектов, которые были реконструированы для этих диапазонов. Продолжая описанные действия для всех диапазонов, геометрия всего объекта может быть восстановлена.

## 7. ТЕСТИРОВАНИЕ

Тестирование производилось на реальных и синтетических данных. Однако эксперименты с реальными данными производить проблематично, поэтому такие эксперименты были проделаны в ограниченном объёме.

Для оценки алгоритмов поиска соответствий и реконструкции использовались метрики – ошибка репроекции и соотношение количества проекций точек, для которых удалось найти соответствия к общему числу проекций. В качестве тестовых данных были выбраны реальные данные, чистые синтетические данные и синтетические данные с шумом. Все данные содержали движение объекта с маркерами в пространстве, объединённое с его вращением. Тестирование показало, что ошибка репроекции не превышает 1 пикселя при наличии шума, а на чистых синтетических данных не превышает 0.2 пикселя. Процент использованных репроекций точек для реконструкции, в зависимости от последовательности и шума, находится в диапазоне примерно 60-70 % для данных с шумом и выше 80 % для чистых синтетических данных. Использовать все найденные проекции не получается потому, что очень часто маркеры видны на одной камере, в то время как на другой камере их не видно.

Для оценки статической погрешности был измерен объект с размером 20 см на расстоянии порядка 5 метров от обеих камер (калибровка системы осуществлялась с шаблоном на том же расстоянии от камер), вычисленное значение оказалось равно 20.09 см, что соответствует относительной погрешности измерений 0.45 % в данном эксперименте.

Для оценки качества алгоритмов вычисления характеристик были произведены комплексные эксперименты, которые заключались в вычислении величины ускорения свободного падения (с учётом сопротивления воздуха и без него). Для этого использовалась физическая модель, для которой было известно аналитическое решение. Полученные значения

координат подставлялись в аналитическое уравнение для выбранной модели, что позволяло получить значения коэффициентов уравнений. Эти коэффициенты выражаются через ускорение свободного падения и коэффициент линейной части сопротивления воздуха. Например, для процесса свободного падения уравнения представлены формулами (8) и (9) – без учёта сопротивления воздуха и с учётом сопротивления воздуха.

$$m \frac{d^2 x}{dt^2} = mg \quad (8)$$

$$m \frac{d^2 x}{dt^2} = mg - m\alpha \frac{dx}{dt} \quad (9)$$

## 8. ЗАКЛЮЧЕНИЕ

В данной работе был предложен новый алгоритм отслеживания движения объекта в пространстве и вычисление характеристик его движения бесконтактным способом. При этом в качестве источника информации предложен объект с инфракрасными маркерами, который наблюдается двумя камерами, которые снимают в инфракрасном диапазоне.

Для решения проблемы восстановления соответствий проекций точек предложен новый метод, основанный на использовании эпиполярных ограничений и схемы голосования для повышения устойчивости результатов.

Были проведены эксперименты, позволяющие оценить качество предложенных алгоритмов и точность измерений.

## 9. ЛИТЕРАТУРА

- [1] H. Bay, T. Tuytelaars, L. V. Gool, SURF: Speeded Up Robust Features, 2008, CVIU'08, Vol. 110, No. 3, pp. 346-359
- [2] R. Hartley, P. Sturm, Triangulation, 1994
- [3] R. Hartley, A. Zisserman. Multiple view geometry in computer vision. Cambridge, UK, 2000. Cambridge University Press
- [4] Hopcroft, John E.; Karp, Richard M. (1973), "An  $n^{5/2}$  algorithm for maximum matchings in bipartite graphs", SIAM Journal on Computing 2 (4): 225–231
- [5] Z. Khan, R. Herman, K. Wallen, T. Balch, An outdoor 3-D visual tracking system for the study of spatial navigation and memory in rhesus monkeys, 2005, Behavior Research Methods, vol. 37
- [6] M. Loaiza, A. Raposo, M. Gattass, A Novel Optical Tracking Algorithm for Point-Based Projective Invariant Marker Patterns, 2007
- [7] S. Smith, Real-time motion segmentation and shape tracking, 1995, In Proc. 5th Int. Conf. on Computer Vision
- [8] M. Weber, H. B. Amor, T. Alexander, Identifying Motion Capture Tracking Markers with Self-Organizing Maps, 2008, Virtual Reality Conference, VR'08. IEEE, pp. 297-298
- [9] A. Yilmaz, M. Shah, Contour-Based Object Tracking with Occlusion Handling in Video Acquired Using Mobile Cameras, 2004, IEEE TPAMI, vol. 26, pp. 1531-1536
- [10] Y. Zhao, J. Westhues, P. Dietz, J. Barnwell, S. Nayar, M. Inami, M. Nol, V. Branzoi, E. Bruns, Lighting Aware Motion Capture using Photosensing Markers and Multiplexed Illuminators, 2007, ACM TOG, Vol. 26, Issue 3, Article 36

## Index of Authors

*A*

*A. Adinetz* 101

*V. Adzhiev* 189

*I. Afanasyev* 33

*B*

*O. Barinova* 95

*B. Barladyan* 12

*Yu. Bayakovski* 51

*A. Bezugly* 73

*N. Biasi* 33

*A. Bondarev* 184

*C*

*M. De Cecco* 33

*A. Chaudhary* 80

*A. Chernomorets* 76

*E*

*V. Eruhimov* 17, 27

*F*

*A. Fornaser* 33

*V. Frolov* 39

*G*

*V. Galaktionov* 39

*K. Gorokhovskiy* 84

*H*

*P. Heribanova* 256

*I*

*A. Ignatenko* 89

*K*

*A. Kharlamov* 39

*A. Konushin* 108

*V. Konushin* 108

*A. Krylov* 73

*K. Kryzhanovsky* 45

*A. Kuharenko* 108

*S. Kumar* 80

*L*

*A. Lukin* 68

*T. Lukina* 108

*E. Luniakova* 51

*I. Lysenkov* 17

*M*

*N. Makhneva* 73

*E. Malikova* 189

*M. Matrosov* 89

*G. Menshikova* 51

*S. Milyaev* 95

*N*

*A. Nasonov* 76

*P*

*A. Parshin* 61

*A. Pasko* 189

*E. Patana* 55

*M. Pestun* 51

*V. Pilyugin* 189

*Ja. Polec* 256

*N. Potekaev* 73

*Z. Pushchina* 45

*R*

*J. Raheja* 80

*M. Rychagov* 55

*S*

*I. Safonov* 45, 55

*A. Selmo* 33

*A. Semashko* 73

*L. Shapiro* 12

*M. Sharma* 80

*A. Shestov* 23

*V. Shumskiy* 61

*V. Sinitsyn* 68

*S. Sivovolenko* 89

*A. Spizhevoy* 27

*M. Storozhilova* 68

*T*

*D. Tarcsiova* 256

*M. Tavernini* 33

*V*

*I. Valiev* 12

*J. Vasavada* 80

*D. Vatolin* 23

*D. Vetrov* 101

*A. Vil'kin* 45

*A. Voloboy* 12

*P. Voronin* 101

*A. Voronov* 23

*K. Vostryakov* 39

*Y*

*A. Yatchenko* 73

*Liu Yuncai* 67

*D. Yurin* 68

*Z*

*D. Zakharkin* 51

*E. Zappia* 33

## Index of Authors (Russian)

*Б*

- И. Белаго* 180, 233  
*А. Бобков* 163  
*В. Бобков* 200  
*А. Бөсөүрин* 144  
*Д. Боголепов* 169, 227  
*И. Бугаев* 169

*В*

- А. Волобой* 242

*Г*

- Н. Гаврилов* 174  
*Д. Гладкий* 180, 233  
*В. Горбачев* 204  
*А. Гордеев* 233  
*А. Груздев* 269  
*В. Гудков* 113, 119, 273, 275

*Д*

- С. Двоенко* 138  
*К. Дорофеев* 273

*Е*

- С. Ериев* 242  
*А. Ефимов* 266

*Ж*

- Д. Жданов* 248

*З*

- С. Завалишин* 152  
*Р. Зейналов* 292

- И. Знаменская* 199

*И*

- А. Игнатенко* 269  
*Н. Ирошников* 218

*К*

- А. Казаков* 144  
*Е. Кайнарова* 204  
*И. Козлутин* 266  
*А. Конушин* 133, 262, 288, 292  
*В. Конушин* 133  
*А. Крыловецкий* 214  
*А. Кудряшов* 200  
*С. Кузиковский* 180, 233  
*В. Кузнецов* 125

*Л*

- А. Ларичев* 218  
*А. Леонов* 163  
*Д. Лепихова* 275  
*О. Логунова* 278  
*Т. Лукина* 133

*М*

- И. Малин* 282  
*И. Мацко* 278  
*И. Метелев* 204  
*Б. Моисеев* 284
- П*
- Е. Паевельева* 128  
*С. Поздняков* 242  
*Е. Попов* 236  
*И. Посохов* 278  
*А. Потягалова* 218

*Р*

- А. Разгулин* 218  
*К. Ротер* 288  
*А. Рыбинцев* 133  
*К. Рябинин* 193

*С*

- Р. Салимзубаров* 158  
*А. Самойлов* 236  
*И. Сафонов* 152  
*М. Синдеев* 288  
*Д. Сопин* 169, 227

*Т*

- Б. Турлапов* 169, 174, 227

*У*

- Д. Ульянов* 169, 227  
*О. Ушмаев* 119, 125

*Ф*

- В. Фролов* 269

*Х*

- П. Харебов* 158  
*М. Харинов* 208  
*Ю. Хохлова* 180

*Ч*

- В. Чебров* 163  
*И. Черников* 214  
*Н. Четвериков* 262  
*А. Чигорин* 284  
*Динь Шанг* 138

*Ю*

- Г. Юзаков* 149

*Я*

- Е. Яковлев* 204

УДК 004.9  
ББК 32.973.26-018.2  
Г78

Г78 **ГрафиКон'2012:** 22-я Международная конференция по компьютерной графике и зрению: Москва, МГУ имени М.В. Ломоносова, 01–05 октября 2012 г.: Труды конференции. – М.: МАКС Пресс, 2012. – 300 с.

ISBN 978-5-317-03808-3

УДК 004.9  
ББК 32.973.26-018.2

---

Подготовка оригинал-макета:  
*М.М. Мизотин, А.В. Насонов*

Напечатано с готового оригинал-макета

Подписано в печать 05.09.2011 г.  
Формат 60x90 1/8. Усл.печ.л. 33,75. Тираж 100 экз. Изд. № 367.

Издательство ООО “МАКС Пресс”  
Лицензия ИД № 00510 от 01.12.99 г.

119992, ГСП-2, Москва, Ленинские горы, МГУ им. М.В. Ломоносова,  
2-й учебный корпус, 627 к.  
Тел. 939-3890, 939-3891. Тел./Факс 939-3891.

