

**25th Anniversary of the International Conference
GraphiCon2015
of The «GraphiCon» Scientific Society
of The Institute of Computing for Physics and Technology**

September 22-25, 2015, Protvino, Park Drakino, Moscow region, Russia

GraphiCon2015 The Conference Proceedings

**Юбилейная 25-я Международная конференция
GraphiCon2015**

**АНО научного общества «Графикон»
Института физико-технической информатики**

22-25 сентября 2015 г. Протвино, Парк Дракино, Московская область,
Россия

**GraphiCon2015 Труды Международной научной
конференции**

©Институт физико-технической информатики, 2015

**Комитеты GraphiCon2015****ISBN 978-5-88835-041-6**

9 785888 350416

Организационный Комитет:

Александр Бугаев, ИРЭ РАН — Почетный Председатель
Георгий Генс, ЛАНИТ — Генеральный спонсор
Игорь Куницкий, РДTeX — Генеральный спонсор
Надежда Шалашилина, ЛАНИТ — Почетный Секретарь
Станислав Клименко, ИФТИ — Председатель Оргкомитета
Владимир Галактионов, ИПМ им.М.В.Келдыша РАН — Зам. Председателя Оргкомитета
Валерий Афанасьев, МФТИ — Председатель жюри конкурса студенческих работ
Владимир Захарушкин, МФТИ — Технический директор
Кирилл Чувиллин, МФТИ — Составитель сборника Трудов
Владимир Пестриков, МФТИ — Руководитель волонтеров
Александр Нычкин, ИФТИ — Зам. Председателя Оргкомитета
Айрат Хасьянов, КФУ — Организатор GraphiCon2016, Казань
Влада Кугуракова, КФУ — Организатор GraphiCon2016, Казань
Раиль Фатхуллин, Таттелеком — Организатор GraphiCon2016, Казань
Антон Обоймов, — веб-мастер

Программный Комитет**Со-председатели:**

Александр Бухановский, Университет ИТМО
Виктор Дебелов, ИВМиМГ СО РАН
Антон Коношин, ВМК МГУ
Сергей Ротков, ННГАСУ
Денис Шергин, Unigine

Члены Программного Комитета:

Алешин Владимир (МФТИ), Астахов Юрий (ИФТИ), Афанасьев Валерий (МФТИ), Барладян Борис (ИПМ РАН), Бобков Валерий (ИАПУ ДВО РАН), Бондарев Александр (ИПМ РАН), Брусенцев Петр (AVILab), Васин Юрий (ННГУ), Ватолин Дмитрий (МГУ), Вельбицкий Игорь (Фонд Глушкова, Украина), Венгринович Валерий (ИПФ НАН, Белоруссия), Волобой Алексей (ИПМ РАН), Галактионов В.А. (ИПМ РАН), Дижевский Андрей (ИФТИ), Долговесов Борис (ИАиЭ СО РАН), Ерёмченко Евгений (Неогеография), Жданов Александр (ИТМиВТ), Жилияков Евгений (БелГУ), Захарушкин Владимир (МФТИ), Игнатенко Алексей (МГУ), Иевлева Ольга (ЮФУ), Копылов Андрей (ТулГУ), Крылов Андрей (МГУ), Лавренко Елена (УМР), Лаврентьев Михаил (НГУ), Массель Людмила (ИСЭ СО РАН), Местецкий Леонид (МГУ), Муратова Галина (ЮФУ), Никитин Игорь (ИФТИ), Никитина Ляля (ИФТИ), Палташев Тимур (США), Пилогин Виктор (МИФИ), Сметанин Юрий (РФФИ), Судейкин Михаил (МФТИ), Толок Алексей (ИПУ РАН), Турлапов Вадим (ННГУ), Чеповский Андрей (ВШЭ).

IPC Members: Arnold David (UK), Sabine Coquillart (INRIA, France), Marina Gavrilova (University of Calgary, Canada), Andres Iglesias (University of Cantabria, Spain), Beckhaus Steffi (Germany), Wolfgang Heiden (BRS, Germany), Hagen Hans (Germany), Andre Hinkenjann (IVC BRS, Germany), Hui-Huang Hsu (Taipei), Jehn-Ruey Jiang (Taipei), Michalcak Georg (Switzerland), Magnenat-Thalmann Nadia (Switzerland), Gianluca Mura (Politecnico Di Milano, Italy), Nielson Gregory (USA), Prasolova-Forland Ekaterina (NTNU, Norway), Reiser Martin (NTU, Singapore), Michael Rychagov (Samsung Research Center, Russia), Samet Refik (Ankara Uni, Turkey), Vladimir Savchenko (Hosei University, Japan), Seaman Bill (Duke Uni, USA), Sourin Alexei (NTU, Singapore), Sourina Olga (FhG IGD, Germany), Tom Sederberg (Brigham Young University, USA), Jiao Ying Shi (P.R.China), Shih Timothy K. (Taipei), Skala Vaclav (Czech Republic), Solomonick Abraham (Israel), Thalmann Daniel (Switzerland), Wetzel Iryna (Intergraph, Switzerland), Wolodchenko Alexander (Germany).

Генеральные спонсоры:

АО ВНИИАЭС

Российский фонд фундаментальных исследований

Корпорация ЛАНИТ

ЗАО РДТЭХ(информационные технологии)

Организаторы и спонсоры:

Институт физико-технической информатики

Московский физико-технический институт (государственный университет)

Международный центр по ядерной безопасности

ЗАО РТСофт (средства и системы автоматизации)

Полиграфическое объединение «ПРИНТ-Ателье»

Институт прикладной математики им.М.В.Келдыша РАН

НИИ Научномкомых компьютерных технологий Университета ИТМО

Институт вычислительной математики и математической геофизики СО РАН

Факультет вычислительной математики и кибернетики МГУ им.М.В.Ломоносова

Кафедра инженерной геометрии, компьютерной графики и автоматизированного проектирования ННГАСУ

AVILab

Unigine

Адрес для связи:

Институт физико-технической информатики

Протвино, Московской обл., Заводской проезд 6, 142281, Россия

Тел. +7-4967-744761, e-mail: Stanislav.Klimenko@gmail.com

Рекомендовано к изданию Объединённым научно-техническим советом Кафедры физико-технической информатики МФТИ и Института физико-технической информатики.

©Институт физико-технической информатики, 2015

Дорогие коллеги!

25 лет назад, в феврале 1991 в Москве прошла 1-я международная конференция по компьютерной графике и зрению ГрафиКон'91. Она была организована Академией наук СССР, Союзом Архитекторов СССР и некоторыми другими организациями при содействии и поддержке американской ассоциации ACM SIGGRAPH. Тогда на волне эйфории от перестройки ряд ведущих американских специалистов приехал в Москву, где по существу, как сейчас говорят, дали мастер-классы на тему, что такое современная компьютерная графика, зачем она, как устроена и что может. Они также передали советским организаторам первого Графикона детальное ноу-хау как готовить, организовывать и проводить конференции по компьютерной графике.

Среди американских докладчиков были профессора ведущих университетов и суперкомпьютерных центров, такие как Д. Гринберг, Д. Фоли и М.Бейли, руководители компаний мировой величины, уже вошедших в историю компьютерной графики, как президент компании "Pixar" Э.Кэтмулл. Свои доклады представили также Д.Ласситер из "Pixar", который только что получил первый в истории Оскар за компьютерную анимацию, а также легендарный Джим Кларк, создатель компании "Silicon Graphics" долгие годы бывшей законодателем мод в области профессиональных графических станций. Главными организаторами со стороны ACM/SIGGRAPH были Б.Акино, Д.Браун и Э.Гудрич. Их энтузиазм и искреннее желание помочь советским специалистам выйти из изоляции на международный уровень позволил им получить немалый бюджет от ACM и вообще сделать 1-ю конференцию Графикон возможной.

Среди организаторов с советской стороны следует выделить две фигуры. Это Сергей Голованов, которому после посещения одной из грандиозных ежегодных конференций ACM/SIGGRAPH пришла идея об организации некоего подобия их конференции в тогдашнем СССР и который проговорил эту идею с американцами и руководством отделения информатики (ОИВТА) РАН. Вторым человеком, вклад которого в организацию конференций Графикон можно назвать решающим, был Юрий Матвеевич Баяковский. Юрий Матвеевич был среди тех, кому РАН в лице академика А.А. Самарского поручила организацию новой международной конференции. Именно Ю.М. Баяковский стал движущей силой многих последующих конференций. В значительной степени благодаря его энтузиазму, увлеченности машинной графикой, высочайшей квалификации, умению подбирать людей мы обязаны тем, что сейчас проводим 25-ю по счету конференцию.

С 1991 г. ежегодные конференции Графикон проводились в разных городах России, на них представлялись работы по широкому спектру направлений компьютерной графики, визуализации, обработки изображений и компьютерного зрения. Конференции порой раз-

личались по уровню докладов и организации, однако позволяли получить понимание того, что происходит в данный момент в упомянутых областях в нашей стране.

Компьютерная графика с ее богатым арсеналом методов и технологий продолжает оставаться весьма востребованной в самых различных областях и приложениях. Еще более актуальными, как показывает практика, проявляют сейчас себя такие ее смежные области как компьютерное зрение и обработка изображений. Бурное развитие современных промышленных и компьютерных технологий ставит все новые задачи, которые обеспечивают новые вызовы и фронт работ для специалистов этих областей.

Именно благодаря деятельности энтузиастов в области компьютерной геометрии и графике в различных городах СССР и Российской Федерации это направление в науке получило высокий приоритет в различных нормативных государственных документах, где указано, что компьютерная геометрия и графика относятся к приоритетным направлениям развития науки и техники и входят в состав критически важных технологий науки и промышленности.

Организационный и программный комитеты Юбилейной 25-й международной конференции по компьютерной графике, визуализации и зрению «GraphiCon2015» приветствуют участников конференции и желает плодотворной работы.

От имени комитетов:

Владимир Галактионов

Виктор Дебелов

Станислав Клименко

Сергей Ротков

Приглашенные и ключевые доклады

T-Splines: The Invention and Commercialization of a CAD Surfacing Technology

Dr. Thomas W. Sederberg

Abstract

Because of its beautiful mathematical properties, NURBS surfaces have many been the standard surface representation in the CAD industry for over thirty years. However, it is well known that that NURBS also have some serious limitations. For example, nearly every trimmed NURBS model is not water tight, i.e., they have small mathematical gaps. This causes problems for downstream processes such as analysis. Also, most NURBS models have many control points that carry no geometric information and hence make them difficult for designers to model with. T-Splines were invented in 2003 to solve these problems. This talk will discuss the invention of T-Splines, share lessons learned in the creation of a company to commercialize T-Splines, and discuss the use of T-Splines for iso-geometric analysis.

About author

Dr. Thomas W. Sederberg is a professor of computer science at Brigham Young University, where he also serves as associate dean of the BYU College of Physical and Mathematical Sciences. He has been active in computer graphics research since 1975. He received the Computer Graphics Achievement Award from ACM SIGGRAPH in 2006 and the Pierre Bezier Prize from the ACM Solid Modeling Association in 2013.

Концепция непрерывной морфологической модели цифрового бинарного изображения

Местецкий Леонид Моисеевич

Аннотация

Цель нашего подхода состоит в том, чтобы использовать эффективную элегантную модель диаграммы Вороного множества отрезков и многоугольных фигур для построения непрерывного представления формы объектов в растровых бинарных изображениях. Предлагаемый подход позволяет обеспечить, во-первых, корректное определение границы и скелета для этих объектов, а во-вторых, дает возможность использования высокоэффективных алгоритмов вычислительной геометрии для получения скелетов.

Предложенный метод включает следующие этапы:

- аппроксимация бинарных изображений многоугольными фигурами;
- построение диаграммы Вороного полученного множества многоугольных фигур методами вычислительной геометрии;
- выделение скелета из диаграммы Вороного в удобном формате;
- регуляризация скелета на основе стрижки.

Об авторе

Местецкий Леонид Моисеевич – доктор технических наук (1992), профессор (1998), действительный член Российской Академии естественных наук (2004). По окончании мехмата МГУ (1971) работал научным сотрудником в научно-исследовательском Институте Министерства Обороны (1971-1992), преподавал в Тверском государственном университете (1992-2005). С 2000 г. профессор кафедры «Математические методы прогнозирования» на факультете вычислительной математики и кибернетики МГУ, с 2004 - профессор кафедры «Интеллектуальные системы» МФТИ, заведующий кафедрой информатики и математики Международного университета в Москве. Опубликовал более 150 работ по прикладной математике, моделированию, компьютерной графике, обработке и распознаванию изображений. Читает лекции по основным курсам «Обработка и распознавание изображений», «Математические методы распознавания образов», и спецкурсам «Математические методы в биометрии», «Непрерывные морфологические модели и алгоритмы».

Concept of Continuous Morphological Model of Digital Binary Image

Abstract

The aim of the concept of continuous morphological model for raster binary image is to use the correct and elegant model of Voronoi diagram of line segments and polygonal figures to obtain the continuous description of a discrete object shape. Such concept should ensure that, firstly, the correct definition of the boundary and skeleton for this object, and secondly, to enable highly efficient use of computational geometry algorithms for obtaining the skeleton.

The proposed approach consists of parts:

- *Approximation of binary images by polygonal figures;*
- *Construction of Voronoi diagram of obtained set of figures by methods of computational geometry;*
- *Obtaining the skeleton from the Voronoi diagram in a convenient format;*
- *Regularization of the skeleton based on the pruning.*

About author

Leonid Mestetskiy – earned an MS in mathematics from Lomonosov Moscow State University (1971). Between 1971-1992 he worked in the Defence Research Institute as a research fellow (1971-1975), a senior research fellow (1976-1991), and the leading research fellow (1992). In 1992-2000 he worked as a professor in Tver' State University, Department of Information Systems and Technologies. From 2000 to present he holds a professorship in Lomonosov Moscow State University in the Department of Mathematical Forecast Methods. Since 2004 he is also a professor of the Moscow Institute of Physics and Technology, Department of Intelligent Systems. Also since 2004, he works as the head of the Informatics and Mathematics Department at the International University in Moscow. He received the Ph.D. degree (1982), D.Sc. degree (1991), in computer science, and professor degree in information technologies (1998). He is a Member of the Russian Academy of Natural Sciences (2004). He has published more than 150 papers on applied mathematics, modeling, computational geometry, computer graphics, image processing and recognition. He lectures on the basic courses «Image processing and recognition», «Mathematical methods for pattern recognition», and special courses «Mathematical methods in biometrics», «Continuous morphological models and algorithms».

Возможности применения российского программного обеспечения виртуального прототипирования VR Concept в промышленности и научных исследованиях

Денис Захаркин, ООО «АВИ Лаб», Антон Лебедев, ОАО «корпорация "ВНИИЭМ"», Петр Брусенцев, ООО «АВИ Лаб»

Аннотация

VR Concept – инструмент трехмерного, виртуального прототипирования, предназначенный для автоматизации работы главного конструктора над проектом, посредством проведения регулярных совещаний на виртуальном цифровом прототипе/макете. Данный подход, позволяет снизить количество ошибок на всех стадиях проектирования, и уменьшить кол-во согласований за счет наглядного представления.

Позволяет в кратчайшие сроки разворачивать виртуальный прототип изделия с его функциональной и трехмерной имитационной моделью, на системах виртуальной реальности, любой конфигурации.

Позволяет проводить инспектирование электронного макета продукта на всех этапах жизненного цикла, что снижает риск ошибок еще на этапе проектирования.

На этапе разработки компоновки, позволяет проводить имитационное прототипирование, изменяя параметры и компоновку продукта, находить оптимальное соотношение тактических характеристик.

Позволяет совмещать твердотельную модель с результатами суперкомпьютерного моделирования и анализа физических процессов. Сокращает потребность в создании физических макетов и стендов.

Система визуализации «GLView» для имитационно-тренажерных комплексов и систем виртуального окружения

М.В. Михайлюк, М.А. Торгашев

Аннотация

В докладе представлена архитектура и основные возможности системы визуализации трехмерных виртуальных сцен «GLView» в масштабе реального времени в моно и стерео режимах. Эта система может использоваться в моделирующих и имитационно-тренажерных комплексах управления сложными динамическими процессами, системах виртуального окружения, виртуальных лабораториях, мультимедийных руководствах и других областях.

Сведения об авторах

Михайлюк Михаил Васильевич, д.ф.-м.н., профессор, Научно-исследовательский институт системных исследований РАН,

Email: mix@niisi.ras.ru

Торгашев Михаил Александрович, к.ф.-м.н., зав. сектором, Научно-исследовательский институт системных исследований РАН,

Email: mtorg@mail.ru

Визуальный анализ кластерных структур в многомерных объемах данных

А.Е. Бондарев, В.А. Галактионов, Институт прикладной математики им. М.В. Келдыша РАН

Аннотация

Доклад посвящен вопросам построения алгоритмов для визуального анализа кластерных структур в многомерных объемах данных. Целью работы является построение комплекса алгоритмов визуализации и визуальной аналитики, позволяющего изучение кластерных структур в многомерных объемах данных без применения алгоритмов кластеризации, вносящих изменения в исходные данные. Для анализа кластерных структур в многомерном объеме данных предлагается использовать методы отображения точек исходного многомерного пространства на вложенные в это пространство многообразия меньшей размерности. Данный подход базируется на построении самоорганизующихся карт SOM, применении метода главных компонент PCA и построении упругих карт Elastic Maps с последующей реализацией процедуры отжига для этих карт. Для реализации полной и последовательной обработки многомерного массива данных вышеупомянутые методы и подходы выстраиваются в последовательность применяемых методов и алгоритмов, образуя единую технологическую цепочку обработки данных. Применение подобной цепочки позволяет получить информацию о кластерной структуре исследуемого объема многомерных данных на разных уровнях глубины анализа и детализации информации.

Опыт создания мультимедийных систем реального времени для обучения и презентаций

В.С. Долговесов, Институт Автоматики и Электростроения СО РАН

Аннотация

В докладе представлены подходы к применению современных компьютерных технологий в образовательных и презентационных приложениях. В частности, рассматриваются конкретные решения, базирующиеся на мультимедийных системах реального времени, таких как системы виртуальной реальности, системы приёма, обработки и трансляции видео и аудио потоковых данных, генерации графики, интерактивного управления.

About the autor

Boris S. Dolgovesov (Ph.D) is a head of Synthesizing Visualization System Laboratory at Institute of Automation and Electrometry SB RAS.

His contact email is bsd@iae.nsk.su

Задачи и решения компьютерной графики и геометрического моделирования в цифровой биомедицине

Вадим Е. Турлапов, Нижегородский государственный университет им. Н.И. Лобачевского

Аннотация

В докладе дается обзор современного состояния развития в основном открытого программного обеспечения компьютерной графики и геометрического моделирования, обслуживающего задачи трехмерной цифровой биомедицины. Раскрывается арсенал библиотек, программных комплексов и систем, как ресурс вычислительной геометрии и геометрического моделирования, медицинской 3D визуализации, сегментации, геометрической реконструкции. Демонстрируются новые 3D технологии в медицине, созданные на базе этих и аналогичных библиотек. Обсуждаются проблемы биомедицины пока нерешенные графическим сообществом и пути их решения, в том числе: детектирование и количественная оценка 3D аномалий для автоматизации диагностики; персонализация параметризованной модели анатомии человека, путем экстракции параметров из персональной томограммы.

Ключевые слова: Компьютерная графика, геометрическое моделирование, 3D цифровой биомедицине, проблемы, решения, обзор

Работа поддержана грантом (соглашение от 27 августа 2013 г. № 02.В.49.21.0003 между МО РФ и ННГУ).



Об авторе

Вадим Турлапов является профессором кафедры математического обеспечения и суперкомпьютерных технологий Нижегородского государственного университета им. Н.И. Лобачевского (ННГУ), руководит лабораторией компьютерной графики и магистерской программой «Компьютерная графика». Закончил радиофизический факультет ННГУ, работал в НИИ прикладной математики и кибернетики при ННГУ, НГТУ им.Р.Е.Алексеева, НФ ГУ-ВШЭ, закончил аспирантуру и докторантуру МАИ (Москва). Является автором идеальной классификации пространственных рычажных механизмов, формализовавшей связь между структурной формулой механизма и уравнениями его кинематики (1997-98), а также автором идеи, принципа построения и системы команд проективно-геометрического сопроцессора для ЭВМ (1992), со-автором высокопроизводительных методов научной визуализации и глобального освещения (2011-2014). В настоящее время – член ACM, член научного совета сообщества GraphiCon, член программного комитета конференции Russian Supercomputing Days и рецензент журнала “Supercomputing Frontiers and Innovations”.

Problems and solutions of computer graphics and geometric modeling to digital biomedicine

Vadim E. Turlapov, *Lobachevsky State University of Nizhni Novgorod, Russia*

Abstract

In connection with the progress of tomography, bioimaging, microscopy, and computer technology, digital biomedicine is intensively developing as a basis for innovative medicine and public health. Computer graphics and geometry play a key role in this development. Here we review the current state of development of mainly open-source software of computer graphics and geometric modeling, three-dimensional digital service tasks of biomedicine. The arsenal of libraries, software packages and systems, which can be used in solving the actual problems, is surveyed. The focus is on the libraries of computational geometry and geometric modeling, 3D medical imaging, segmentation, geometric reconstruction. New 3D medical technologies based on these and similar libraries are demonstrated. We discuss the problems of biomedicine not yet solved by the graphic community as well as the ways to solve them. These problems include (i) detection and quantification of 3D abnormalities to automate diagnosis and (ii) personalization of the parameterized model of human anatomy by extracting parameters from a personal tomography.

Keywords: *Computer Graphics, Geometric Modeling, 3D Digital Biomedicine, Problems, Solutions, Review*

About the autor

Vadim Turlapov is Professor of the Software and Supercomputer Technologies Department of the Lobachevsky State University of Nizhny Novgorod (UNN), head of the Laboratory of Computer Graphics and the master's program Computer Graphics. He graduated from the Radiophysics Faculty, UNN, worked in the Institute of Applied Mathematics and Cybernetics, graduated from the Post-graduate and Doctoral School of MAI (Moscow). He is the author of an ideal classification of spatial linkages, which formalize the link between the structural formula of the mechanism and its kinematics equations (1997-98), as well as the author of the idea, principles and instruction set for a projective geometric coprocessor (1992), and the coauthor of high performance methods in scientific visualization and global illumination (2011-14). Presently, member of ACM, a member of the scientific council of the GraphiCon community, a member of the program committee of the Russian Supercomputing Days conference and reviewer of the journal "Supercomputing Frontiers and Innovations".

Содержание

<i>Турлапов В. Е.</i> Задачи и решения компьютерной графики и геометрического моделирования в цифровой биомедицине	1
<i>Чембаев В.Д., Будаев В.П., Желтов В.С., Нотфуллин Р.С., Селиванов В.А.</i> Локальные оценки в решении уравнения глобального освещения	7
<i>Барладян Б.Х., Бирюков Е.Д., Волобой А.Г., Ершов С.В., Шапиро Л.З.</i> Моделирование работы камеры заднего вида	12
<i>Авербух В.Л., Бахтерев М.О., Васёв П.А., Манаков Д.В., Стародубцев И.С.</i> Развитие подходов к разработке специализированных систем компьютерной визуализации	17
<i>Пацко В.С., Стародубцев И.С., Федотов А.А.</i> Физическая визуализация множеств достижимости в задачах управления	22
<i>Фролов В.А., Галактионов В.А.</i> Простой и недорогой подход к реализации регенерации путей	28
<i>Жданов Д.Д., Алейников А.А., Кудреватых Е.В., Дерябин Н.Б.</i> Построение физически корректной модели источника света по конечному набору лучей	34
<i>Жданов Д.Д., Алейников А.А., Кудреватых Е.В., Валиев В.И.</i> Модель сложного источника излучения в методе двунаправленной стохастической трассировки лучей	39
<i>Жданов Д.Д., Ершов С.В., Волобой А.Г.</i> Адаптивный выбор глубины трассировки обратного луча в методе двунаправленной стохастической трассировки лучей	44
<i>Джсен-Руй Янг, Фу-Сианг Чанг</i> Мульти-серверная динамическая балансировка нагрузки для сетевых виртуальных окружений	50
<i>Зубов А.Ю., Сеньюкова О.В.</i> Сегментация изображений магнитно-резонансной томографии головного мозга с помощью сопоставления с несколькими атласами	56
<i>Галактионов В.А., Пестун М.В.</i> Алгоритмы построения и распознавания навигационных описаний маршрутов для картографических компьютерных систем	62
<i>Васин Ю.Г., Утешева Т.Ш.</i> Повышение эффективности технологических процессов обработки и анализа графических данных в ГИС	68
<i>Афанасьев В.В., Игнатенко А.В., Волобой А.Г.</i> Итерационный метод восстановления спектра по цвету RGB	72
<i>Павлов В.А., Завьялов С.В., Волвенко С.В.</i> Анализ совместной работы методов сопровождения объектов в видеопотоке, получаемом с летательного аппарата	75
<i>Кудряшов А.П., Бобков В.А.</i> Зашивка дыр в объединённом наборе триангуляционных сеток	80
<i>Новоторцев Л.В., Волобой А.Г.</i> Нахождение областей, содержащих здания, на аэрофотоснимках	85
<i>Волкова С.С.</i> О методе подсчета посетителей при наклонном расположении камеры	89

<i>Гурьева М.Н.</i>	Применение самоорганизующейся карты Кохонена для сегментации гиперспектральных изображений	93
<i>Михайлюк М.В., Торгашев М.А.</i>	Система визуализации «GLView» для имитационно-тренажерных комплексов и систем виртуального окружения	96
<i>Коротин А.С., Попов Е.В.</i>	Оценка точности цифровых моделей рельефа, применяемых для территориальных исследований	102
<i>Кудряшов А.П., Соловьёв И.В.</i>	Распознавание дополнительных объектов на топографическом плане при реконструкции городских сцен	107
<i>Лебедев Л.И., Васин Ю.Г.</i>	Двухкритериальный алгоритм распознавания объектов графических изображений на базе КЭКМ	112
<i>Золотов В.А., Петрищев К.С., Семенов В.А.</i>	Исследование методов пространственного индексирования динамических сцен на основе регулярных октодеревьев	115
<i>Зачесов А.А.</i>	Метод сжатия полигональных сеток с потерями	123
<i>Рябинин К.В., Чуприна С.И.</i>	Адаптация систем научной визуализации к сторонним решателям	127
<i>Дерябин Н.Б., Соколов В.Г., Жданов Д.Д., Копылов М.С.</i>	Автоматизация генерации серий реалистичных изображений с использованием языка сценариев Python	132
<i>Куксова С.А.</i>	Сравнение методов распознавания символов номерного знака автомобиля	137
<i>Зипа К.С., Земцов Е.Е., Игнатенко А.В.</i>	Расчет объективной и субъективной интегральной яркости объекта по изображению широкого динамического диапазона	140
<i>Начаров Д.В., Михайлюк Ю.П.</i>	Метод преобразования динамического диапазона цифровых изображений с целью улучшения различимости объектов	145
<i>Рябушева А.Д., Лебедев Л.И.</i>	Структурная оптимизация алгоритма распознавания на базе КЭКМ	150
<i>Касьянов В.Н., Золотухин Т.А.</i>	Visual Graph – система визуализации сложно структурированной информации большого объема на основе графовых моделей	154
<i>Панфилова К.В.</i>	Компенсация линейного смаза цифровых изображений с помощью метода Люси-Ричардсона	163
<i>Дебелов В.А.</i>	Интерференция света, изотропные прозрачные объекты, трассировка лучей	168
<i>Сумин Д.А., Бондарев А.Е., Волобой А.Г.</i>	Автоматическое приближение набора точек поверхностями второго порядка в задачах анализа многомерных данных	174
<i>Мельман С.В., Бобков В.А., Черкашин А.С.</i>	Технология и система визуализации больших объемов синоптических данных	178
<i>Андреев С.В., Бондарев А.Е., Бондаренко А.В., Визильтер Ю.В., Галактионов В.А., Гудков А.В., Желтов С.Ю., Жуков В.Т., Иловайская Е.Б., Князь В.А., Мануковский К.В., Новикова Н.Д., Ососков М.В., Силаев Н.Ж., Феодоритова О.Б., Бондарева Н.А.</i>	Моделирование и визуализация работы энергетической установки сложной формы в режиме стереоанимации	183

<i>Бондарев А.Е., Галактионов В.А.</i>	
Визуальный анализ кластерных структур в многомерных объемах данных	188
<i>Зипа К.С., Игнатенко А.В.</i>	
Алгоритм тональной компрессии на основе восприятия человека	194
<i>Копылов И.В.</i>	
Комбинированный алгоритм локализации номеров железнодорожных цистерн	200
<i>Валуйская Я.А.</i>	
Подсчет людей, используя данные, собранные с помощью сенсора Microsoft Kinect	205
<i>Валиев И.В., Волобой А.Г., Денисов Е.Ю., Ершов С.В., Поздняков С.Г.</i>	
Преобразования XYZ в спектр для свойств поверхности	209
<i>Захарова А.А., Шкляр А.В.</i>	
Значение особенностей зрительного восприятия для интерпретации визуальных моделей произвольных данных	214
<i>Грингауз А.Д., Шальнов Е.В., Коцушин А.С.</i>	
Алгоритм для определения положения пользователей в мировых координатах и его применение для задач слежения	222
<i>Масленников О.П., Мильман И.Е., Сафиуллин А.Э., Бондарев А.Е., Низаметдинов Ш.У., Пиллюгин В.В.</i>	
Разработка и развитие системы интерактивного визуального анализа многомерных данных	227
<i>Князь В.А.</i>	
Оптическая система захвата движения для анализа и визуализации трехмерных процессов	232
<i>Долговесов Б.С., Городилов М.А., Храмов И.Д.</i>	
Синхронизация распределённого воспроизведения мультимедийных данных на полиэкранах	237
<i>Городилов М.А., Долговесов Б.С., Радостев А.Х.</i>	
Разработка супервизорной системы удаленного управления коммутацией медиапотоков	241
<i>Долговесов Б.С., Городилов М.А., Фатьянов Ф.В.</i>	
Система обработки и отображения визуальной информации для тренажерных комплексов	244
<i>Ватаначоте К., Кумара В., Ши Т.К., Клименко А.С.</i>	
Автоматическое создание динамических вставок для спецэффектов видео	247
<i>Васин Д.Ю., Ротков С.И.</i>	
Технология формирования локальных архивов 2D-чертежей	254
<i>Васин Д.Ю., Аратский А.В.</i>	
Распознавание символов на основе инвариантных моментов графических изображений	259
<i>Долговесов Б.С.</i>	
Опыт создания мультимедийных систем реального времени для обучения и презентаций	265

Задачи и решения компьютерной графики и геометрического моделирования в цифровой биомедицине*

В. Е. Турлапов

vadim.turlapov@cs.vmk.unn.ru

Нижегородский государственный университет им. Н.И. Лобачевского, Россия

Дается обзор сегодняшнего состояния развития в основном открытого программного обеспечения компьютерной графики и геометрического моделирования, обслуживающего задачи трехмерной цифровой биомедицины. Раскрывается арсенал библиотек, программных комплексов и систем, как ресурс вычислительной геометрии и геометрического моделирования, медицинской 3D визуализации, сегментации, геометрической реконструкции. Демонстрируются новые 3D технологии в медицине, созданные на базе этих и аналогичных библиотек. Обсуждаются проблемы биомедицины пока нерешенные графическим сообществом и пути их решения, в том числе: детектирование и количественная оценка 3D аномалий для автоматизации диагностики; персонализация параметризованной модели анатомии человека, путем экстракции параметров из персональной томограммы.

Ключевые слова: Компьютерная графика, геометрическое моделирование, 3D цифровой биомедицине, проблемы, решения, обзор

Problems and solutions of computer graphics and geometric modeling to digital biomedicine*

Vadim E. Turlapov

Lobachevsky State University of Nizhni Novgorod, Russia

In connection with the progress of tomography, bioimaging, microscopy, and computer technology, digital biomedicine is intensively developing as a basis for innovative medicine and public health. Computer graphics and geometry play a key role in this development. Here we review the current state of development of mainly open-source software of computer graphics and geometric modeling, three-dimensional digital service tasks of biomedicine. The arsenal of libraries, software packages and systems, which can be used in solving the actual problems, is surveyed. The focus is on the libraries of computational geometry and geometric modeling, 3D medical imaging, segmentation, geometric reconstruction. New 3D medical technologies based on these and similar libraries are demonstrated. We discuss the problems of biomedicine not yet solved by the graphic community as well as the ways to solve them. These problems include (i) detection and quantification of 3D abnormalities to automate diagnosis and (ii) personalization of the parameterized model of human anatomy by extracting parameters from a personal tomography.

Keywords: Computer Graphics, Geometric Modeling, 3D Digital Biomedicine, Problems, Solutions, Review

1. Введение

Биомедицина – раздел медицины, изучающий организм человека с теоретических позиций. Она включает сведения и исследования общие медицине и фундаментальным биологическим наукам, таким, как биохимия, биология, в том числе нейробиология, гистология, генетика, эмбриология, анатомия, физиология, патология, биомедицинский инжиниринг, зоология, ботаника и микробиология. Биомедицина создает фундамент для всех медицинских приложений, диагностики и лечения, делает возможным внутриклеточное и молекулярное понимание механизмов болезни, появление новых лекарственных средств, стволовых клеток для клеточной терапии. На переживаемом нами уровне развития науки и технологий вектор развития цифро-

вой биомедицины отчетливо направлен в сторону трехмерной биомедицины, как самой высокотехнологичной и инновационной. Данный обзор посвящен достижениям компьютерной графики и геометрического моделирования, которые обеспечили современное развитие 3D биомедицины, ряду интересных результатов этого развития и проблемам, которые остаются нерешенными и ожидают усилий со стороны сообщества компьютерной графики. Вызовом для развития трехмерной цифровой биомедицины послужило взрывное развитие томографии и трехмерной электронной микроскопии, создающих последовательности двумерных слоев высокого разрешения. Быстро обнаружилось, что трехмерные данные способны качественно улучшить результаты диагностики и исследований, если будут созданы методы и программное обеспечение (ПО) 3D обработки и визуализации этих данных, и, на их основе, новые трехмерные технологии в медицине и биологии. В связи с этим перед компьютерной графикой и геометрическим моде-

Работа поддержана грантом (соглашение от 27 августа 2013 г. № 02.В.49.21.0003 между МО РФ и ННГУ). Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347

лированием возникли следующие задачи: 1) высокопроизводительная 3D визуализация и сегментация томограмм, хранимых в специальном формате DICOM; 2) параметризованное геометрическое моделирование человека, его органов и систем; 3) трехмерная сегментация и геометрическая реконструкция органов и человека в целом по данным томографии и другим источникам; 4) детектирование аномалий и их количественная 2D-3D оценка (автоматизация диагностики). Разумеется геометрия и графика в биомедицине должны поддерживать и функциональное моделирование человека, его органов, систем и манипулирование ими так, как это делается на этапах пред- и постпроцессинга в инженерных системах. Выходом такого функционального моделирования должны быть исследовательские и обучающие системы виртуальной реальности.

2. Обзор существующих разработок

2.1 Высокопроизводительная 3D визуализация и сегментация томограмм

Пожалуй наиболее существенный вклад в создание программного обеспечения современной биомедицины внесла компания Kitware, Inc.

(www.kitware.com, USA-1998). Она организовала создание целой серии Open Source программных продуктов, жизненно необходимых трехмерной цифровой биомедицине. Прежде всего это:

1) Visualization Toolkit (VTK, www.vtk.org) – программный комплекс для обработки изображений, полигональной 3D графики и научной визуализации;

2) Insight Segmentation and Registration Toolkit (ITK, www.itk.org) – программный комплекс для сегментации трехмерных медицинских данных, получивший в США статус национальной библиотеки программ для создания медицинских систем;

3) CMake – обеспечение сборки сложных систем, которое первоначально разрабатывалось для сборки ИТК и ИТК-приложений. CMake управляет процессом сборки используя простые платформо- и компиляторо-независимые конфигурационные файлы;

4) ParaView (www.paraview.org) – открытая система высокопроизводительной научно-технической визуализации, реализующая практически все существующие на сегодняшний день приемы научной визуализации.

Имеет она и возможности медицинской 3D визуализации. Сегодня Kitware не только продолжает работать в Medical Computing, в HPC & Visualization и Software Process, но и расширила свои интересы на Computer Vision и Data & Analytics. Активными партнерами Kitware являются более 80 университетов и научных центров США.

Десятилетний совместный проект лаборатории Penn Image Computing and Science Laboratory (PICS), университета штата Пенсильвания) и института Scientific Computing and Imaging (SCI, университет штата Юта), поддержанный грантом National Institute of Biomedical Imaging and Bioengineering (NIBIB), закончился созданием компактного и сравнительно простого в использовании комплекса ИТК-SNAP (www.itksnap.org). Комплекс ориентирован на интерактивную сегментацию с одновременной работой оператора в трех ортогональных проекциях и на использование в каждой из проекций возможностей метода активного контура (snake) развитых авторами. К настоящему времени пользователями ИТК-SNAP в мире сделаны уже многие сотни публикаций. В сфере 3D сегментации и визуальной реконструкции микрофотографий с электронных микроскопов (ЭМ), которая пока требует существенно большего участия специалиста, разработан открытый программный комплекс ILASTIK (http://ilastik.org), ориентированный на интерактивную классификацию, сегментацию и последующий анализ. Модульное ПО обеспечивает автоматизированную пиксельную и объектную классификацию, полуавтоматический трекинг объектов, полуавтоматическую сегментацию и подсчет объектов без их детектирования.

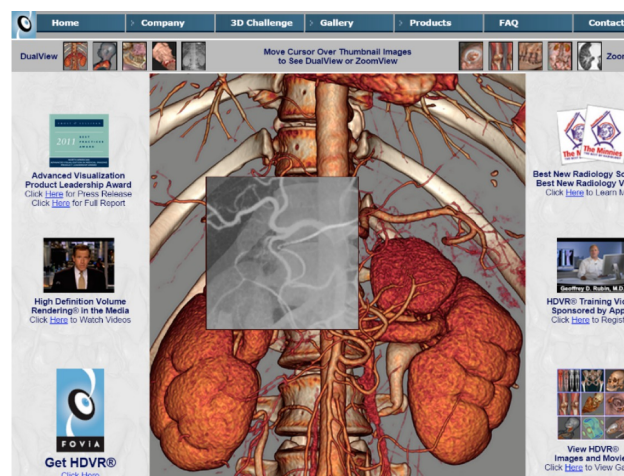


Рис. 1: Сайт компании Fovia Inc., представляющий HDVR®

ILASTIK разработан специальной командой университета Гейдельберга (Heidelberg Uni., Germany) при финансовой поддержке ряда сообществ и организаций: Heidelberg Collaboratory for Image Processing (HCI); HHMI (Howard Hughes Medical Institute) Janelia Farm Research Campus; CellNetworks Excellence Cluster (www.cellnetworks.uni-hd.de). В медицинской 3D визуализации результатов томографии есть и коммерческие программные комплексы. Самым известным из них остается продукт

Fovia HDVR® (High Definition Volume Rendering) фирмы Fovia Inc. (California, USA, www.fovia.com), встроенный в ПО большей части зарубежных томографов (Рис.1). Существенной чертой этого продукта является отказ от использования графических процессоров.

На российском рынке есть продукт фирмы Inobites (www.inobites.ru, Москва), позиционируемый как отечественный аналог HDVR®. Этот продукт практически уступает HDVR® только производительностью. Возможно рекордную на сегодняшний день производительность имеет открытый DICOM визуализатор InVols, разработанный в ННГУ им. Н.И. Лобачевского. Invols использует лучшие современные алгоритмы DVR, включая предварительное интегрирование и блочную декомпозицию, оптимизирован под архитектуру GPU. Еще в апреле 2012 он был способен визуализировать в реальном времени томограмму из 5168 слоев: «GPU RayCasting for a huge dataset» (<https://www.youtube.com/user/ngavrilov86>). Производительность InVols продолжает с каждым годом расти вместе с производительностью GPU. Также в ННГУ решена и проблема измерения и обеспечения заданного качества 3D визуализации томограммы в условиях отсутствия эталона и, в результате, предложен подход сравнения алгоритмов в осях производительность-качество [2].

2.2 Параметризованное моделирование

Линия трехмерного моделирования человека, его органов и систем, представлена достаточно большим числом фирм и комплексов моделей. Исторически абсолютное большинство 3D моделей были созданы для целей наглядного изучения анатомии и не являются параметризованными. Авторы наиболее продвинутых продуктов уже начали движение в сторону создания параметризованных моделей. Один из лидеров в этом фирма Zygote Inc. (<http://www.zygote.com/>).

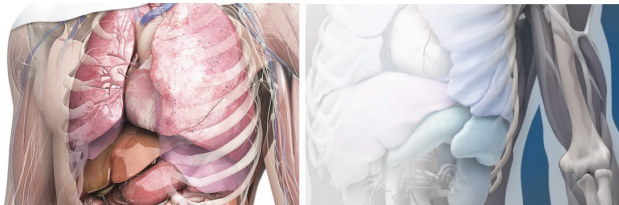


Рис. 2: Поверхностная текстурированная и твердотельная модели с сайта Zygote Inc.

С 1994 года фирма оказывает наукоемкие услуги по 3D моделированию в биомедицине, индустрии развлечений и на профессиональном рынке. Сегодня наряду с поверхностными полигональными текстурированными моделями в форматах систем 3d-моделирования: 3D Studio Max;

Blender (OBJ); Cinema 4D; Generic OBJ; Maya (<http://www.zygote.com/poly-models>), предлагаются также твердотельные CAD-модели (Рис.2) для использования в биоинженерных нуждах внутри CAD систем: ProE/Creo; IGES; ParaSolid; SolidWorks; Step. Рыночная цена пары поверхностных моделей Male-Female – около \$25000. Чем выше степень параметризации модели, тем технологичнее ее применение на указанных рынках.

Так английская фирма Plasticboy предлагает новую версию анатомических моделей Male-Female серии Anatomy Models Store, параметризованную для движения (rigged model) в среде Autodesk Maya: V07 Anatomy 3D models (Motionbuilder Compatible). Остальные форматы моделей (для других 12 систем 3D моделирования) поставляются стандартными (т.е. без движения). Примерная цена – около \$2000 на 1 рабочее место. Пример на сайте компании демонстрирует работу при движении всего костно-мышечного каркаса близкую к естественной (<http://www.plasticboy.co.uk/store>).

Еще одна известная фирма Visible Body (<http://www.visiblebody.com/>) поддерживает уровень моделирования близкий к указанным выше, но продает не модели, а продукты на их основе «под ключ». Это также ведет к постепенному расширению круга параметризованных моделей и полноты параметризации.

Все эти модели созданы как благодаря развитым коммерческим системам 3D моделирования, так и современным открытым библиотекам и средам геометрического моделирования. К числу таких открытых библиотек следует отнести прежде всего CGAL – The Computational Geometry Algorithms Library (www.cgal.org) [4]. Проект по созданию этой библиотеки стартовал в 1996 году как международный европейский проект первоначально 7 сторон: Utrecht University (The Netherlands, prime contractor), ETH Zürich (Switzerland), Free University of Berlin (Germany), INRIA (France), Max-Planck-Institute Saarbrücken (Germany), RISC Linz (Austria), Tel-Aviv University (Israel), и финансировался несколькими крупными европейскими научными программами (GALIA, ECG, ACS, Aim@Shape). С 2003 года стали доступны коммерческие лицензии CGAL. CGAL C++ library, заканчиваясь построением сеток, обладает функциональной полнотой и широко используется для создания прикладных систем с геометрическим базисом в следующих областях: GIS, CAD, молекулярная биология, медицина, компьютерная графика, робототехника и т.д. Отечественным аналогом CGAL является геометрическое ядро C3D (<http://c3dlabs.com>), разработавшееся в интересах отечественной CAD КОМПАС-3D компании АСКОН.

Близким по функциональности проектом, но с чисто французским началом, является открытая система и технология для геометрического проектирования и инженерного анализа (Open CASCADE Technology, <http://www.opencascade.org/>). На основе данной библиотеки построена открытая система трехмерного моделирования, сеточной декомпозиции и инженерного анализа SALOME (www.salome-platform.org), способная обеспечить платформу пред- и пост-обработки для численного моделирования, как в инженерных приложениях, так и в медицине.

Параллельно созданы и более скромные открытые библиотеки, и подобные коммерческие, как например, The Geometric Modelling Library (GML), созданная международной компанией Renishaw (<http://www.renishaw.com/geometricmodelling/en/the-gml-14749>).

На такой мощной компьютерно-геометрической поддержке появились, как отдельные прикладные модели, так и целые программные комплексы (libraries and tools), обеспечивающие функциональное моделирование отдельных органов и систем организма человека. Одним из известных комплексов является VMTK (www.vmtk.org) – the Vascular Modelling Toolkit. VMTK решает задачи 3D реконструкции, геометрического анализа, генерации конечно-элементных сеток, анализа поверхностей и моделирования кровотока в сердечно-сосудистой системе человека.

Развитие систем 3D моделирования сделало возможным использование 3D модели пациента, как основы для интегрирования и совместной трактовки результатов его исследования современными методами медицинской диагностики. Исследования и создание систем в этом направлении активно поддерживается компанией IBM, мировым лидером производства суперЭВМ, увидевшей здесь перспективы для массового применения суперЭВМ в реализации технологий на основе больших данных. Результатами проектов IBM 3D Avatar стали три новых технологии, поддерживаемых фирмой Nhum Technologies (www.nhum.com, читается как New Me). По смыслу – это новый трехмерный Я, построенный от начала до конца на трехмерном образе пациента (3D-аватаре конкретного человека), его медицинских исследований, 3D планировании и наблюдении процесса лечения. Первая технология Study Insight – сервис интеграции данных различных исследований пациента с возможностью коллективного Интернет-доступа к ним врачей и проведения телеконсилиумов по пациенту. Технология Visual Health – сервис навигации по 3D модели анатомии человека (около 3000 объектов) для врачей с любого стандартного web-клиента с привязкой клинических данных к органам, образно говоря, – трехмерная медицинская карта пациента (Рис.3).

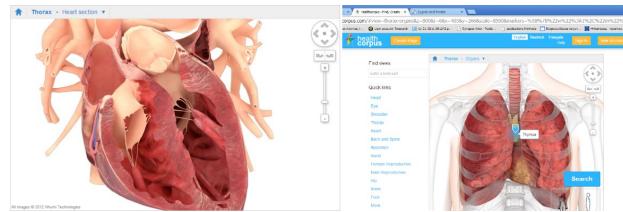


Рис. 3: Примеры моделей, используемых в системах Visual Health и Healthcorpus

2.3 Трехмерная сегментация, геометрическая реконструкция и анализ органов и человека в целом

Как видим, даже общая трехмерная модель анатомии начинает играть существенную роль в медицинских системах. Тем более – персональная модель анатомии конкретного пациента. С ее построением с еще большим основанием можно говорить о персональной медицине. Имеет смысл начинать прием врача с 3D реконструкции тела пациента. Наиболее дешевый способ для этого – использование изображений с видеокамер. Можно говорить, что эта задача уже подхвачена специалистами и нашла неплохие решения. Одно из решений можно увидеть на сайте фирмы Body Labs (www.bodylabs.com) – подготовлена бета-версия продукта BodyKIT для высокоточной реконструкции 3D модели тела и захвата его движения с MS Kinect. В проекте также декларированы возможности провести сравнение, морфинг, анимацию, и усреднение тел. Другое решение – на сайте компании Itseez, поддерживающей и развивающей всемирно известную библиотеку компьютерного зрения OpenCV, (<http://itseez.com/products/itseez3d/>) – уже работающее ПО 3D реконструкции тела человека и других объектов (Рис.4).



Рис. 4: Пример 3D реконструкции на сайте Itseez

Для медицинских технологий, использующих задачи 3 и 4, найдено интересное прикладное решение, получившее название Виртуальный анатомический стол (Dissection Table). Впервые та-

кой стол был передан в Стэндфордский университет (Stanford Uni.) фирмой Anatomage Inc. (www.anatomage.com) в 2011 для экспериментального использования и развития. Главной идеей стола было соединение 3D визуализации индивидуальных данных томографии с параметризованной моделью анатомии (Рис.5).



Рис. 5: Пример объединения 3D модели внутренних органов и кровеносной системы со скелетом и контуром тела извлеченными из реальной томограммы, представленный на сайте <http://medical.anatomage.com>

Это предлагалось использовать прежде всего для целей обучения студентов. К объектам модели можно легко привязать любую обучающую информацию, а конкретная томограмма должна давать информацию о реальном состоянии и патологиях органов. Однако универсальная реализация данной концепции за пределами частного случая, представленного на сайте оказалась слишком трудоемкой и не завершена до настоящего времени. И развитие системы в прошедшие 4 года пошло по линии накопления библиотеки распространенных патологий органов (к настоящему времени в системе Anatomage Table собрано около 120 патологий), и накопления инструментария сегментации и геометрической реконструкции и методов анализа отдельных частей тела. Этот программный инструментарий получил название Invivo (<http://medical.anatomage.com/medical-products/invivo5>). На сегодняшний день он содержит 8 функций и сертифицирован для применения в медицине. Каждый экземпляр Anatomage Table поставляется с пакетом Invivo.

Более простой и прагматичный продукт Sectra Visualization Table предложен в 2012 году шведской фирмой Sectra (http://www.sectra.com/medical/-sectra_table/), которая уже более 20 лет работает на рынке медицинской визуализации. Это стол для 3D визуализации томограмм в реальном времени, который позиционируется как полезный для предоперационной подготовки хирурга, коллектив-

ных обсуждений результатов томографии и изучения патологий студентами на реальном материале. Программное обеспечение 3D визуализации поставляется вместе со столом.

В России также в настоящее время завершается разработка виртуального стола более подобного Anatomage Table. Она выполняется силами малого предприятия при Саратовском медицинском университете на основе разработанного там атласа нормальной и патологической анатомии человеческого тела InBody Anatomy (<http://inbody.pro>, <http://www.npo-leader.com/>) с более чем 100 патологиями.

В нижегородском университете в 2013 году реализован облачный сервис высокопроизводительной 2D-3D визуализации томограмм для организации на его основе персональных кабинетов врача и пациента. Экспериментальный вариант сервиса (<http://85.143.3.10:4526>) способен работать также в режиме виртуального стола типа Sectra Table (Рис.6).

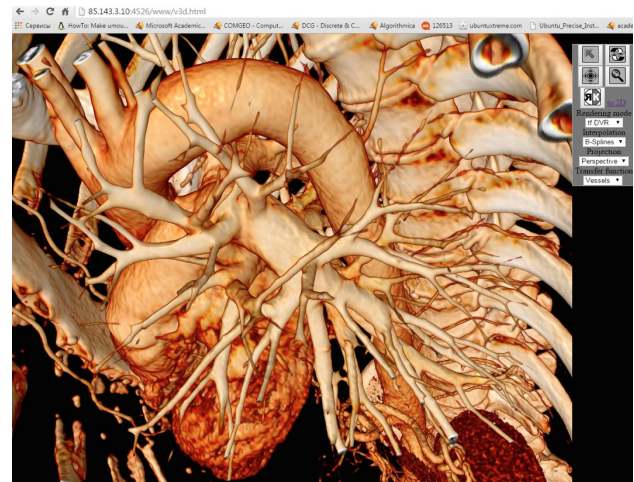


Рис. 6: Пример 3D визуализации средствами облачного сервиса ННГУ

3. Актуальные нерешенные и решаемые проблемы

Часть решаемых в настоящее время проблем можно найти (<http://medical.anatomage.com/medical-products/invivo5>) среди новых функций Invivo5. Это Compare Cases – попарное сравнение случаев (количественное детектирование динамики), причем не только в 2D, но и в 3D (например, сравнение объема легких на вдохе и выдохе). Это также Isolate Anatomy and Capture Videos – пользователь может реконструировать отдельную деталь скелета и выполнить с ней различные манипуляции: масштабировать, повернуть, параллельно перенести. Еще шире возможности опционального компонента Medical Design (MD). Он может как геометрически

реконструировать фрагменты томограммы (Create Digital Models from Patient Scans), так и импортировать в созданные модели внешние объекты, например, искусственный клапан – в сердце.

Очевидно, реконструкция выполняется на основе сегментации, а ее возможности сегодня крайне ограничены и нуждаются в дальнейшем развитии и регуляризации. Очень полезны в этом плане работы по исследованию диагностики отдельных органов, примером которых являются [5,7].

Особенно сложна сегментация и диагностика мягких тканей. Полноценное детектирование аномалий органа должно начинаться с точного детектирования его границ. Что чаще всего крайне сложно сделать по данным томограммы из-за того, что томограф имеет ограниченное разрешение, и при шаге срезов в 0.5 мм не может достоверно детектировать объекты величиной или толщиной (для тонких границ) меньше 1 мм. Эта проблема вплотную соприкасается с проблемой слияния 3D модели нормальной анатомии с томограммой тела пациента или фрагмента тела. Однако, здесь возможно движение навстречу друг-другу со стороны углубления и специализации параметризации 3D модели и со стороны методов сегментации. Есть надежда на регуляризацию алгоритмов сегментации за счет применения таких опорных моделей.

Есть также важнейшая задача полноценного вовлечения ультразвуковой диагностики в 3D реконструкцию и задача реконструкции на этой основе не статических, а динамических (особенно доплеровских) объемных моделей: во многих случаях исследования динамики органа/системы ультразвуковое исследование остается золотым стандартом. Примером работы в этом направлении является работа [5]. Также актуальна обработка томограмм с появляющихся на рынке так называемых «быстрых» МРТ томографов (FLASH MRI).

Довольно широким фронтом ведутся работы по функциональному моделированию отдельных органов и систем тела человека с перспективой создания «виртуального клона» по журналистской терминологии. Здесь можно сослаться на работы по электро-механическому моделированию сердца [6] и кровотока в сосудистой системе человека [1]. Во всех этих случаях за компьютерной графикой остается геометрографическое сопровождение функционального моделирования персонального виртуального организма. Свежим примером такой функции является визуализация динамики сокращающегося киберсердца пациента (<https://youtu.be/2LPboySOSvo>), построенной в RIKEN на японском суперкомпьютере K и представленной впервые 04 июня 2015 на конференции SIGGRAPH 2015.

Литература

- [1] Gamilov T., Ivanov Yu., Kopylov P., Simakov S., Vassilevski Yu. Patient specific haemodynamic modeling after occlusion treatment in leg // *Mathematical Modelling of Natural Phenomena*. 9(6), 85-97 (2014)
- [2] Gavrilov, N. I.; Turlapov, V. E. Novel approach to development of direct volume rendering algorithms based on visualization quality assessment // *Programming and Computer Software*. Volume: 40 Issue: 4 Pages: 174-184
- [3] Senyukova O., Lukin A., Vetrov D. Automated Atlas-Based Segmentation of NISSL-Stained Mouse Brain Sections Using Supervised Learning // *Programming and Computer Software*. 2011. Vol. 37, no. 5. pp.245-251.
- [4] The CGAL Project. CGAL User and Reference Manual. CGAL Editorial Board, 4.6.2 edition, 2015.
- [5] Yatchenko A., Krylov A. Cross-Frame Ultrasonic Color Doppler Flow Heart Image Unwrapping // *Lecture Notes in Computer Science*, Vol. 9126, 2015, pp.265-272.
- [6] Костин В.А., Осипов Г.В. Самосогласованная электромеханическая активность в упруго закрепленном волокне сердечной мышцы // *Проблемы прочности и пластичности*. 2014. Т. 76. № 4. С.357-363
- [7] Насонова А.А., Крылов А.С. Выделение сосудов на изображениях глазного дна и его оценка качества // *Биотехносфера*, №. 3, Издательство «Политехника», 2014, С.24-25.

Об авторе

Вадим Евгеньевич Турлапов является профессором кафедры математического обеспечения и суперкомпьютерных технологий Нижегородского государственного университета им. Н.И. Лобачевского (Россия), руководит магистерской программой Компьютерная графика и лабораторией Компьютерной графики и мультимедиа.

Email: vadim.turlapov@cs.vmk.unn.ru

Локальные оценки в решении уравнения глобального освещения*

В.Д. Чембаев¹, В.П. Будаков¹, В.С. Желтов¹, Р.С. Нотфулин¹, В.А. Селиванов²
 chembervint@gmail.com | budakov@mppei.ru | zheltov@list.ru | renat@notfullin.com

¹Кафедра светотехники, Национальный исследовательский университет «МЭИ», Москва, Россия

²Кафедра телевидения, Московский технический университет связи и информатики, Москва, Россия

В статье рассматриваются локальные оценки метода Монте-Карло в решении уравнения глобального освещения. Локальные оценки позволяют непосредственно вычислять яркость в заданной точке в заданном направлении. На их основе может быть построен алгоритм визуализации трехмерных сцен. Также рассматривается принципиальная возможность видо независимой визуализации трехмерной сцены на основе разложения тела яркости по сферическим гармоникам.

Ключевые слова: Локальная оценка, двойная локальная оценка, глобальное освещение, Монте-Карло

Usage of Local Estimations at the Solution of Global Illumination Equation*

V.D. Chembayev¹, V.P. Budakov¹, V.S. Zheltov¹, R.S. Notfulin¹, V.A. Selivanov²

¹Light Engineering Department Moscow Power Engineering Institute, Moscow, Russia

²TV Department Moscow Technical University of Communications and Informatics

In this article, we consider local estimations of the Monte Carlo method for solving the equation of the global illumination. The local estimations allow directly calculating the luminance at a predetermined point in a given direction. Whereby based on it can be built visualization of 3D scenes. Also considering the possibility of a species independent visualization of 3D scene based on the brightness object decomposition in spherical harmonics.

Keywords: Local Estimation, Double Local Estimation, Global Illumination, Monte Carlo

Introduction

Visualization of 3D scenes is produced on the basis of solving the global illumination equation, which represents Fredholm integral equation of the second kind [4]

$$L(\mathbf{r}, \hat{\mathbf{l}}) = L_0(\mathbf{r}, \hat{\mathbf{l}}) + \frac{1}{\pi} \int L(\mathbf{r}', \hat{\mathbf{l}}') \sigma(\mathbf{r}; \hat{\mathbf{l}}, \hat{\mathbf{l}}') \left| (\hat{\mathbf{N}}, \hat{\mathbf{l}}') \right| d\hat{\mathbf{l}}' \quad (1)$$

where $L(\mathbf{r}, \hat{\mathbf{l}})$ is the radiance at the point \mathbf{r} in the direction $\hat{\mathbf{l}}$, is the bidirectional scattering distribution function (reflectance or transmittance), L_0 is the radiance of the direct radiation straight near the sources, $\hat{\mathbf{N}}$ is the normal at the point \mathbf{r} to the surface of the scene.

The global illumination equation (1) does not have the analytical solution, and the numerical simulation methods are used for its solution such as ray tracing, finite element method and photon mapping.

Among the methods used statistical methods for modeling global illumination equation based on Monte Carlo methods had the most development. In this case all the methods used, including the most advanced, such as Metropolis light transport [5] are of direct simulation Monte Carlo, where the determination of the

required characteristics (brightness or luminosity) is based on counting the hits or just raytracing.

In this paper, we propose an alternative approach - local estimations of the Monte Carlo method, which are based not on the count hits in a neighborhood of the point under, but on the evaluation of the probability of transition from the trajectory of a Markov chain to the point [3] [6].

Mathematics

From the integral equation for the solid angle, you can go to the well-known integral equation of Fredholm second kind on surfaces

$$L(\mathbf{r}, \hat{\mathbf{l}}) = L_0(\mathbf{r}, \hat{\mathbf{l}}) + \frac{1}{\pi} \int_{(\Sigma)} L(\mathbf{r}', \hat{\mathbf{l}}') \sigma(\mathbf{r}; \hat{\mathbf{l}}, \hat{\mathbf{l}}') F(\mathbf{r}', \mathbf{r}) d^2 r' \quad (2)$$

where

$$F(\mathbf{r}', \mathbf{r}) = \frac{\left| (\hat{\mathbf{N}}(\mathbf{r}), \mathbf{r} - \mathbf{r}') (\hat{\mathbf{N}}(\mathbf{r}'), \mathbf{r} - \mathbf{r}') \right|}{(\mathbf{r} - \mathbf{r}')^4} \Theta(\mathbf{r}', \mathbf{r})$$

and

$$\hat{\mathbf{l}}' = \frac{\mathbf{r} - \mathbf{r}'}{|\mathbf{r} - \mathbf{r}'|}$$

And based on it, you can construct an algorithm for its solution by Monte Carlo method. However, wandering along the Σ surfaces of the scene visualization is not a trivial task. Conventional scheme of wandering of the

Работа выполнена при финансовой поддержки Министерства образования и науки Российской Федерации научной исследовательской работы №2487 базовой части государственного задания №2014/123.

Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347

Monte Carlo method in the space, which requires the integral to integration over the volume. Integral over the volume

$$d^3r' = |\mathbf{r} - \mathbf{r}'|^2 dr' d\hat{\mathbf{l}}',$$

$$d\hat{\mathbf{l}}' = \frac{|\hat{\mathbf{N}}(\mathbf{r}'), \mathbf{r} - \mathbf{r}'|}{(\mathbf{r} - \mathbf{r}')^2} d^2r'. \quad (3)$$

For integration over dr' we will use equivalent transformation with usage δ -function properties

$$\int_{(\Sigma)} L(\mathbf{r}', \hat{\mathbf{l}}') \sigma(\mathbf{r}; \hat{\mathbf{l}}', \hat{\mathbf{l}}) F(\mathbf{r}', \mathbf{r}) d^2r' \equiv$$

$$\equiv \int_0^\infty \oint L(\mathbf{r}', \hat{\mathbf{l}}') \sigma(\mathbf{r}; \hat{\mathbf{l}}', \hat{\mathbf{l}}) \frac{F(\mathbf{r}', \mathbf{r})}{|\hat{\mathbf{N}}(\mathbf{r}'), \hat{\mathbf{l}}'|} |\mathbf{r} - \mathbf{r}'|^2 \times$$

$$\times \frac{|\hat{\mathbf{N}}(\mathbf{r}'), \hat{\mathbf{l}}'| d^2r'}{|\mathbf{r} - \mathbf{r}'|^2} \delta(\xi_0 - |\mathbf{r} - \mathbf{r}'|) dr' \quad (4)$$

where ξ_0 is a solution of the $\Pi(\mathbf{r} - \xi_0 \hat{\mathbf{l}}') = 0$, $\Pi(\mathbf{r}) = 0$ – surface Σ equation. The surface equation can be included directly into (4), because the ratios $\xi_0 - |\mathbf{r} - \mathbf{r}'| = 0$ and $\Pi(\mathbf{r} - |\mathbf{r} - \mathbf{r}'| \hat{\mathbf{l}}') = 0$ are equivalent. At that it is important to consider the δ -function properties.

$$\int_a^b \delta(f(x)) dx =$$

$$= \frac{1}{\left| \frac{df(x)}{dx} \right|_{x=x_0}} \int_a^b \delta(x - x_0) dx, f(x_0) = 0. \quad (5)$$

Accordingly we will get for global illumination equation

$$L(\mathbf{r}, \hat{\mathbf{l}}) = L_0(\mathbf{r}, \hat{\mathbf{l}}) +$$

$$+ \frac{1}{\pi} \int L(\mathbf{r}', \hat{\mathbf{l}}') \sigma(\mathbf{r}; \hat{\mathbf{l}}', \hat{\mathbf{l}}) G(\mathbf{r}', \mathbf{r}) d^3r', \quad (6)$$

where new geometric factor

$$G(\mathbf{r}', \mathbf{r}) = \frac{|\hat{\mathbf{N}}(\mathbf{r}'), \mathbf{r} - \mathbf{r}'|}{(\mathbf{r} - \mathbf{r}')^3} \Theta(\mathbf{r}', \mathbf{r}) \times$$

$$\times \left| \frac{d\Pi(\mathbf{r} - \xi \hat{\mathbf{l}}')}{d\xi} \right|_{\xi=|\mathbf{r}-\mathbf{r}'|} \delta\left(\Pi(\mathbf{r} - |\mathbf{r} - \mathbf{r}'| \hat{\mathbf{l}}')\right), \quad (7)$$

where $\Pi(\mathbf{r} - |\mathbf{r} - \mathbf{r}_0| \hat{\mathbf{l}}') = 0$, $\hat{\mathbf{l}}' = \frac{\mathbf{r} - \mathbf{r}_0}{|\mathbf{r} - \mathbf{r}_0|}$. The solution (6) can be shown as Neumann series

$$L(\mathbf{r}, \hat{\mathbf{l}}) = L_0(\mathbf{r}, \hat{\mathbf{l}}) +$$

$$+ \frac{1}{\pi} \int \frac{1}{\pi} \int L_0(\mathbf{r}_1, \hat{\mathbf{l}}_1) \sigma(\mathbf{r}_2; \hat{\mathbf{l}}_1, \hat{\mathbf{l}}_2) G(\mathbf{r}_1, \mathbf{r}_2) d^3r_1 \times$$

$$\times \sigma(\mathbf{r}; \hat{\mathbf{l}}_2, \hat{\mathbf{l}}) G(\mathbf{r}_2, \mathbf{r}) d^3r_2 + \dots \quad (8)$$

All members of the series - definite integrals, which will be calculated by the Monte Carlo

$$L(\mathbf{r}, \hat{\mathbf{l}}) = L_0(\mathbf{r}, \hat{\mathbf{l}}) +$$

$$+ \frac{1}{\pi} \frac{1}{N} \sum_{i=1}^N \frac{L_0(\mathbf{r}_{1i}, \hat{\mathbf{l}}_{1i})}{p_1(\mathbf{r}_{1i}, \hat{\mathbf{l}}_{1i})} \frac{\sigma(\mathbf{r}; \hat{\mathbf{l}}_{1i}, \hat{\mathbf{l}}) G(\mathbf{r}_{1i}, \mathbf{r})}{p_2(\mathbf{r}_{1i}, \hat{\mathbf{l}}_{1i} \rightarrow \mathbf{r}, \hat{\mathbf{l}})} +$$

$$+ \frac{1}{\pi^2} \frac{1}{N} \sum_{i=1}^N \frac{L_0(\mathbf{r}_{1i}, \hat{\mathbf{l}}_{1i})}{p_1(\mathbf{r}_{1i}, \hat{\mathbf{l}}_{1i})} \frac{\sigma(\mathbf{r}_{2i}; \hat{\mathbf{l}}_{1i}, \hat{\mathbf{l}}_{2i}) G(\mathbf{r}_{1i}, \mathbf{r}_{2i})}{p_2(\mathbf{r}_{1i}, \hat{\mathbf{l}}_{1i} \rightarrow \mathbf{r}_{2i}, \hat{\mathbf{l}}_{2i})} \times$$

$$\times \frac{\sigma(\mathbf{r}; \hat{\mathbf{l}}_{2i}, \hat{\mathbf{l}}) G(\mathbf{r}_{2i}, \mathbf{r})}{p_2(\mathbf{r}_{2i}, \hat{\mathbf{l}}_{2i} \rightarrow \mathbf{r}, \hat{\mathbf{l}})} + \dots \quad (9)$$

Join the sums into one

$$L(\mathbf{r}, \hat{\mathbf{l}}) = L_0(\mathbf{r}, \hat{\mathbf{l}}) +$$

$$+ \frac{1}{N} \sum_{i=1}^N \left(\frac{1}{\pi} \frac{L_0(\mathbf{r}_{1i}, \hat{\mathbf{l}}_{1i})}{p_1(\mathbf{r}_{1i}, \hat{\mathbf{l}}_{1i})} \frac{\sigma(\mathbf{r}; \hat{\mathbf{l}}_{1i}, \hat{\mathbf{l}}) G(\mathbf{r}_{1i}, \mathbf{r})}{p_2(\mathbf{r}_{1i}, \hat{\mathbf{l}}_{1i} \rightarrow \mathbf{r}, \hat{\mathbf{l}})} +$$

$$+ \frac{1}{\pi^2} \frac{L_0(\mathbf{r}_{1i}, \hat{\mathbf{l}}_{1i})}{p_1(\mathbf{r}_{1i}, \hat{\mathbf{l}}_{1i})} \frac{\sigma(\mathbf{r}_{2i}; \hat{\mathbf{l}}_{1i}, \hat{\mathbf{l}}_{2i}) G(\mathbf{r}_{1i}, \mathbf{r}_{2i})}{p_2(\mathbf{r}_{1i}, \hat{\mathbf{l}}_{1i} \rightarrow \mathbf{r}_{2i}, \hat{\mathbf{l}}_{2i})} \times$$

$$\times \frac{\sigma(\mathbf{r}; \hat{\mathbf{l}}_{2i}, \hat{\mathbf{l}}) G(\mathbf{r}_{2i}, \mathbf{r})}{p_2(\mathbf{r}_{2i}, \hat{\mathbf{l}}_{2i} \rightarrow \mathbf{r}, \hat{\mathbf{l}})} + \dots \right). \quad (10)$$

The last expression can be interpreted as a Markov chain wandering ray with the contribution by the kernel

$$k(x_i \rightarrow x) = \frac{\sigma(\mathbf{r}; \hat{\mathbf{l}}_i, \hat{\mathbf{l}}) G(\mathbf{r}_i, \mathbf{r})}{p_2(x_i \rightarrow x)}. \quad (11)$$

Note that the geometrical form factor (7) contains δ -function, which makes it impossible to direct modeling.

Geometric explanation is shown in Figure 1, which shows the inability to calculate the contribution from the node of the Markov chain to the examined point in examined direction. Feature can be eliminated by integrating over the angle that will be equivalent to the calculation of illumination at a given point \mathbf{r} . This approach is called the local estimation.

To calculate the brightness directly, you can enter additional intermediate node corresponding to the point of intersection with the element of the scene in the opposite direction $\hat{\mathbf{l}}$ from the point \mathbf{r} . This approach is called double local estimate [1]. Figure 2 shows an example of the process for determining the point of visualization.

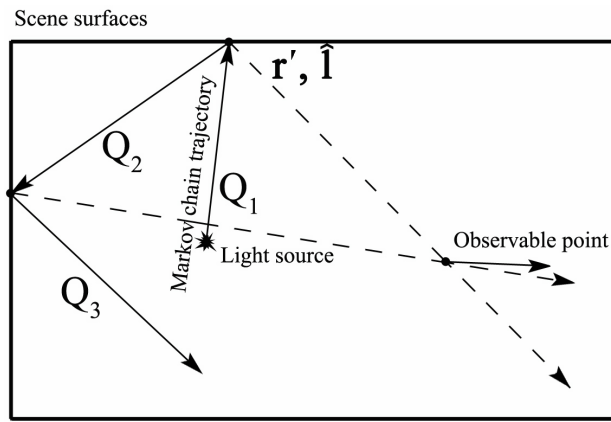


Figure 1: Geometric description of the impossibility of calculating the contribution of brightness from the node of the Markov chain to the studied point in the study area

Implementation and practice

In our work we are working on the implementation of algorithms of local and double local estimations in the .NET Framework environment. Using the managed code is not highly effective in terms of performance, but it allows to focus on the algorithms during the search of optimum.

In the implementation of algorithms in three-dimensional computer graphics directly ray tracing takes significant amount of time. In order to abstract from the problems of optimization of search of intersections through space and focus on our algorithms of local estimations we used the Intel Embree [2] library. High efficiency and simplicity of the interface of this library makes it easy to use it.

Currently in program implemented diffuse rectangular light source, and as a reflection model used mirror, Phong and diffuse reflection models.

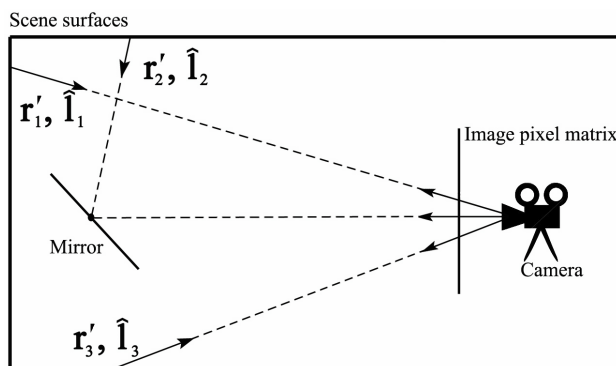


Figure 2: Scheme for determining the points of visualization

Let's consider the basic steps of the visualization of three-dimensional scenes by double local estimation.

In accordance with the general algorithm for calculating the brightness by double local estimation [1], we must determine points and directions (points of visualization), in which we will make the calculation. During visualization of three-dimensional scene we determine these points as points of intersection of the rays that forming the images, issued from the camera. And if the material has a reflection component, then a chain of visualization points is iteratively built.

Figure 2 shows a schematic diagram of determination of visualization points. The brightness of these particular points on specified directions will determine the image. If there are refractive materials in the scene, visualization points are formed similar to the reflection points.

Calculation of high order reflections by double local estimations is a process of multiple constructing a Markov chain of rays from the light source and the calculation of the probability of transition from the anchor point to the point of visualization. Moreover the image of the whole scene can be obtained by only one ray.

Figure 3 shows the visualization of the famous scene Dabrovic Sponza double local estimation by one ray. After calculating the contribution of each point in the image, we are casting a new direction of the ray after reflection and looking for the next point of intersection. From it, we again expect to contribute to all points of the image.

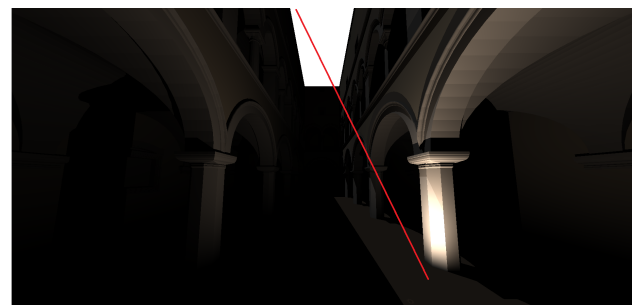


Figure 3: Rendering of the scene, one of multiplicity one ray from the source

The result of accounting of the two multiplicities of the ray reflection is shown in Figure 4.

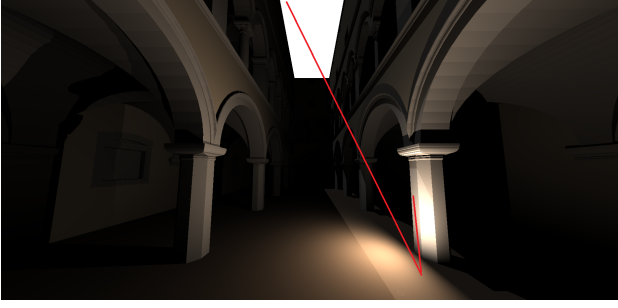


Figure 4: Rendering scene for two multiplicities of one ray from the source

Figure 5 shows an image of one ray with the three multiplicities reflection.

Iteratively, we continue to build a Markov chain and calculate the probability of a transition from a node in the chain calculation points. Chain builds until the weight of the ray above a certain threshold and the ray is not left the scene.

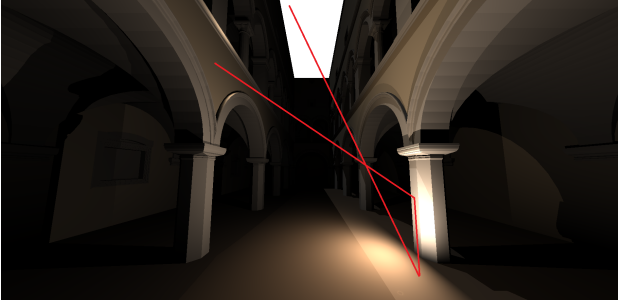


Figure 5: Rendering of the scene in three multiplicities of one ray from the source

Not difficult to see that one ray can really get a full picture of the scene, but it will be biased. Having casted a large number of primary rays, averaging and normalizing the resulting image can be obtained taking into account the multiple reflections of light, shown in Figure 6.

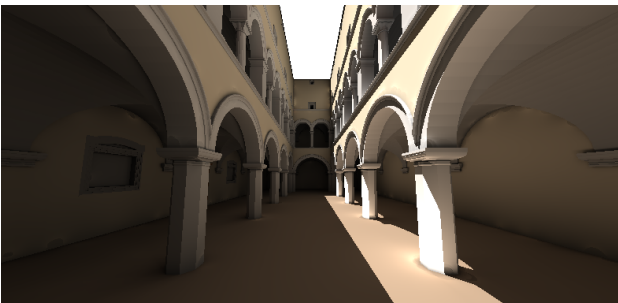


Figure 6: Rendering of the scene by double local estimation

Note that the resulting image contains an explicit artifacts. Not hard to see bursts of brightness near the

bottom of the columns, as well as non-uniformity in many areas of the scene containing corners. This is due to the fundamental problem of double local estimations - infinitely dispersion [3]. One approach to solving this problem is to integrate.

At the same time, we note that local estimations devoid of noise, since they use the same rays for all points of the scene, in contrast to the popular method of Metropolis Light Transport.

It should be noted that a similar scheme of rendering the phenomenological approach without strict mathematical basis was obtained for the first time in [8] and called Instant Radiosity. However, the method is based on local estimates entered in the [3] in 1963.

Presentation of the brightness object

Visualizations discussed in the previous section are dependent on the camera position. However, a double local estimation will allow to calculate the brightness at each point for a variety of directions, which opens the possibility of calculation of multiple reflections in the three-dimensional scene without depending on the position of the camera. Moreover, in contrast to the already existing visualization techniques based on the finite element method (radiosity) [7], in this case we do not use diffuse and full reflection model.

This raises the question, how many areas to be used for storage of the brightness at each point. Obviously, the more the better and the more they must be, the more acute the reflection function. At that it is costly to store information. This question requires a separate research, but one solution would be to use the expansion of the angular distribution of brightness in spherical harmonics

$$L(\mathbf{r}, \hat{\mathbf{l}}) = \sum_{n=0}^N \sum_{m=-n}^n C_n^m(\mathbf{r}) Y_n^m(\hat{\mathbf{l}}) = \sum_{n=0}^N \sum_{m=0}^n (A_n^m(\mathbf{r}) \cos \varphi + B_n^m(\mathbf{r}) \sin \varphi) P_n^m(\hat{\mathbf{l}} \cdot \hat{\mathbf{z}}) \quad (12)$$

where

$$C_n^m(\mathbf{r}) = \oint Y_n^m(\hat{\mathbf{l}}) L(\mathbf{r}, \hat{\mathbf{l}}) d\hat{\mathbf{l}} \quad (13)$$

In this case, in luminance simulation of global illumination equation we can use not a double local estimation but the local estimation directly, the integration is actually taken into account in the deposit. In this case, the expression (13) can be estimated from a random one node and access (9) for computing

$$C_n^m(\mathbf{r}) = \frac{1}{N} \sum_i L_0(\mathbf{r}, \hat{\mathbf{l}}_i) Y_n^m(\hat{\mathbf{l}}_i) + \frac{1}{\pi N} \sum_{i=1}^N \frac{L_0(\mathbf{r}_{1i}, \hat{\mathbf{l}}_{1i}) \sigma(\mathbf{r}; \hat{\mathbf{l}}_{1i}, \hat{\mathbf{l}}) G(\mathbf{r}_1, \mathbf{r})}{p_1(\mathbf{r}_{1i}, \hat{\mathbf{l}}_{1i}) p_2(\mathbf{r}_{1i}, \hat{\mathbf{l}}_{1i} \rightarrow \mathbf{r}, \hat{\mathbf{l}})} Y_n^m(\hat{\mathbf{l}}) +$$

$$\begin{aligned}
& + \frac{1}{\pi^2} \frac{1}{N} \sum_{i=1}^N \frac{L_0(\mathbf{r}_{1i}, \hat{\mathbf{l}}_{1i})}{p_1(\mathbf{r}_{1i}, \hat{\mathbf{l}}_{1i})} \frac{\sigma(\mathbf{r}_{2i}; \hat{\mathbf{l}}_{1i}, \hat{\mathbf{l}}_{2i}) G(\mathbf{r}_{1i}, \mathbf{r}_{2i})}{p_2(\mathbf{r}_{1i}, \hat{\mathbf{l}}_{1i} \rightarrow \mathbf{r}_{2i}, \hat{\mathbf{l}}_{2i})} + \\
& + \frac{\sigma(\mathbf{r}; \hat{\mathbf{l}}_{2i}, \hat{\mathbf{l}}) G(\mathbf{r}_{2i}, \mathbf{r})}{p_2(\mathbf{r}_{2i}, \hat{\mathbf{l}}_{2i} \rightarrow \mathbf{r}, \hat{\mathbf{l}})} Y_n^m(\hat{\mathbf{l}}) + \dots \quad (14)
\end{aligned}$$

The last expression contains integration over the angle that allows you to use directly the local estimate.

Conclusions and future work

Local estimations of the Monte Carlo method allow to calculate directly the brightness at a given point in a given direction. As a result, a double local estimations method can be used to visualize the three-dimensional scene based on multiple reflections.

Based on double local estimations can be built view-independent estimation of the scene, the same method of radiosity. In this case, by contrast, will be used by any model of reflection.

Requires separate consideration of the question of infinite variance and methods of minimizing its impact on the final image.

Bibliography

- [1] *Budak V., Zheltov V.* Local Monte Carlo estimation methods in the solution of global illumination equation // WSCG 2014 Communication Papers Proceedings, 2014, P. 25-31.
- [2] *Ingo Wald, Sven Woop, Carsten Benthin, Gregory S. Johnson, and Manfred Ernst Embree* - A Kernel Framework for Efficient CPU Ray Tracing // ACM Transactions on Graphics (proceedings of ACM SIGGRAPH), 2014
- [3] *M. H. Kalos* On the Estimation of Flux at a Point by Monte Carlo // Nuclear Science and Engineering, 1963, Vol. 16, N.1, p.111-117
- [4] *Kajiya J. T.* The rendering equation // Computer Graphics (Proc. SIGGRAPH'86), 1986. V.20, N4. – P.143-150
- [5] *Veach, E.; Guibas, L. J.* Metropolis light transport // Proceedings of the 24th annual conference on Computer graphics and interactive techniques - SIGGRAPH '97
- [6] *G.I. Marchuk* Monte-Carlo Methods in Atmospheric Optics – Berlin: Springer-Verlag, 1980
- [7] *Cindy Goral, Kenneth E. Torrance, Donald P. Greenberg and B. Battaile* Modeling the interaction of light between diffuse surfaces // Computer Graphics, 1984, Vol. 18, No. 3
- [8] *Alexander Keller* Instant radiosity // SIGGRAPH '97 Proceedings of the 24th annual conference on Computer graphics and interactive techniques, p. 49-56

Моделирование работы камеры заднего вида*

Б.Х.Барладян, Е.Д.Бирюков, А.Г.Волобой, С.В.Ершов, Л.З.Шапиро

obb@gin.keldysh.ru | peb@gin.keldysh.ru | voloboy@gin.keldysh.ru

measure@spp.keldysh.ru | pls@gin.keldysh.ru

Институт прикладной математики им. М.В.Келдыша РАН, Москва

В работе рассматриваются интерактивные алгоритмы преобразования изображений получаемых с помощью стандартной камеры «рыбий глаз» в изображения, соответствующие широкоугольной, нормальной камере и виду сверху всей сцены. Эти преобразования используются в современных системах автомобильной навигации.

Ключевые слова: *камера заднего вида, визуализация, навигация, расширенная реальность.*

Rear camera simulation*

B.Kh.Barladian, E.D.Birukov, A.G.Voloboy, S.V.Ershov, L.Z.Shapiro

The Keldysh Institute of Applied Mathematics Russian Academy of Science

Interactive algorithms of 2D mapping of images produced by a standard “fisheye” camera to the wide, normal and top down images are considered. These transformations are used in modern car navigation systems. Suggested algorithms are fast and convenient for users. The algorithms are integrated into CATIA (Dassault systems) CAD system.

Keywords: *rear camera, navigation, augmented reality, ray tracing.*

1. Введение

Ограниченное пространство для парковки автомобилей на улицах современных городов требует от производителей оборудования оснащения автомобилей различными вспомогательными системами для облегчения водителю процесса парковки. К таким системам относятся камера кругового обзора, заднего вида, системы мониторинга «мертвых зон», системы PARKTRONIC и активного парковочного помощника [1,2]. Особые затруднения испытывают водители при парковке задним ходом и параллельной парковке. В связи с этим оборудование автомобилей парковочными системами с камерой заднего вида стало на сегодняшний день фактически стандартом для современных автомобилей. Это особенно актуально, поскольку часть пространства позади автомобиля не видна ни в зеркала заднего вида, ни в заднее стекло. В настоящее время производители предлагают на рынке большое количество систем, использующих одну или несколько камер для обзора «мертвых зон» автомобиля, которые не видны ни в зеркала, ни в окна [1-4]. Это могут быть как простейшие системы, просто передающие изображение, получаемое с данной камеры на монитор, так и продвинутые системы, обеспечивающие полуавтоматическую парковку при движении задним ходом или параллельную парковку [2].

Наиболее существенным с точки зрения парковки является получение удобного для водителя изображения с камеры заднего вида, поскольку здесь на-

ходит наиболее невидимая часть пространства. Для обеспечения максимального обзора камеры заднего вида используют широкоугольные объективы или даже объективы класса «рыбий глаз» с углом зрения 180 градусов и более [1-4]. В силу особенностей конструкции конкретного автомобиля камеру, в некоторых случаях, приходится располагать в таких местах, что получаемое изображение не является симметричным, относительно оси автомобиля, и направление оси камеры будет иметь достаточно острый угол по отношению к плоскости дороги. В силу всех этих специфических особенностей положения, ориентации камеры, а также использования широкоугольных объективов, получаемые изображения могут быть не слишком удобны для водителя с точки зрения эргономики и читаемости обстановки. Для повышения информативности получаемых изображений применяются различные алгоритмы исправления искажений и преобразования исходного изображения в изображения, как бы получаемых камерами без искажений, с меньшим углом зрения или даже для камеры, смотрящей на автомобиль сверху (bird's eye view, top down [1,5]). Информативность этих изображений может быть дополнительно улучшена путем визуализации на них направляющих линий, т.е. линий, по которым будет двигаться автомобиль при заданном положении руля. В настоящей работе рассматриваются интерактивные алгоритмы преобразования изображений, получаемых с помощью стандартной «рыбий глаз» камеры, в более информативные для водителя представления. Результатом работы разработанных алгоритмов является 2D отображение, описанное в формате, пригодном

Работа поддержана грантом РФФИ 13-01-00454, а также фирмой Integra Inc. (Япония). Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347.

для реализации в аппаратной или программной части камеры или навигатора, установленных на автомобиле.

2. 2D отображение

Как уже упомянуто выше все результирующие изображения являются некоторыми двумерными отображениями исходного изображения, полученного с помощью объектива «рыбий глаз». Выбор алгоритмов и способов представления этого отображения в значительной мере определяет удобство и эффективность разрабатываемых интерактивных процедур. В работе [6] для представления такого отображения исходное изображение разбивалось на некоторые прямоугольные подобласти, для которых применялась сложная система индивидуальных масштабных коэффициентов, заданных в вершинах этих прямоугольников. В настоящей работе был выбран более простой, прозрачный и эффективный для реализации подход для построения двумерного отображения. На выходном изображении отображение определено в узлах некоторой регулярной сетки (рис. 1).

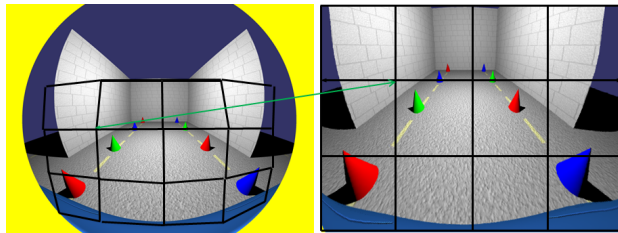


Рис. 1: Преобразование изображения. Слева – исходное изображение, справа – преобразованное.

Для каждого узла сетки на итоговом изображении задается соответствующая точка на исходном изображении. Для нахождения соответствующей точки на исходном изображении по точке итогового изображения внутри прямоугольника сетки используется билинейная интерполяция между вершинами сетки. Выбор шага сетки определяется спецификой задачи. В наших экспериментах сетка с разрешением 100×100 обеспечивала приемлемую точность для большинства рассмотренных преобразований. Большая точность потребовалась лишь в некоторых случаях для преобразования в изображение, имитирующего камеру, смотрящую на автомобиль сверху (см. раздел 4).

3. Интерактивное создание 2D отображения

Как уже было упомянуто выше изображения, непосредственно передаваемые с камеры заднего вида, не слишком удобны для водителя с точки зрения оценки дорожной обстановки. Причинами

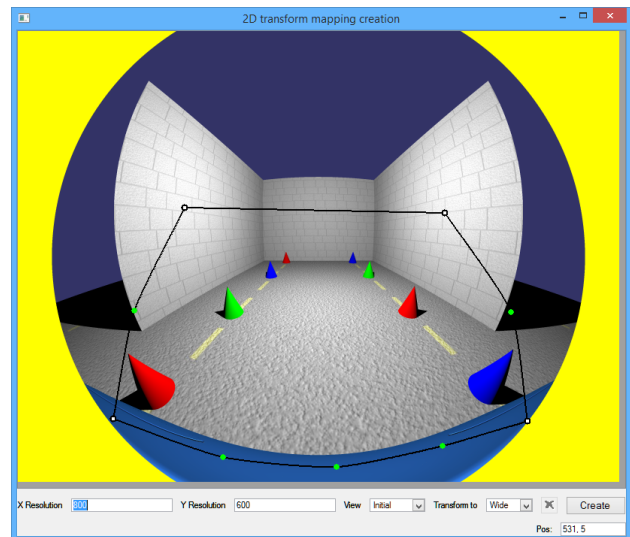


Рис. 2: Исходное изображение камеры заднего вида.

этих неудобств являются специфические положение и ориентация камеры и использование объектива «рыбий глаз». Для исправления этих недостатков производители автомобильных навигаторов обеспечивают возможность преобразования исходного изображения в изображения, соответствующие камере с менее широкоугольным и нормальным объективами [5]. Фактически каждое из этих новых изображений является некоторой частью исходного изображения, части которого могут быть растянуты или сжаты для получения необходимых эффектов. В реальной системе эти преобразования должны работать в режиме реального времени и, соответственно, должны быть реализованы в аппаратуре. В настоящей работе рассматривается интерактивный способ построения таких отображений.

Интерактивная процедура создания 2D отображений реализована в системе Lumicept, интегрированной в систему автоматизированного проектирования САТІА [7]. Система позволяет получать реалистические изображения путем трассировки лучей для камеры заднего вида с учетом положения и ориентации камеры относительно автомобиля и необходимого окружения. Система позволяет моделировать камеру с объективом «рыбий глаз» с углом зрения больше 180 градусов и учитывать дисторсию объектива.

На рис. 2 показано изображение камеры заднего вида, полученное с помощью моделирования в системе [7]. Наша интерактивная система позволяет построить контур на этом изображении, который будет являться границей соответствующего преобразованного изображения – широкоугольного, нормального или вида камеры, смотрящей на автомобиль сверху. Контур строится из четырех кри-

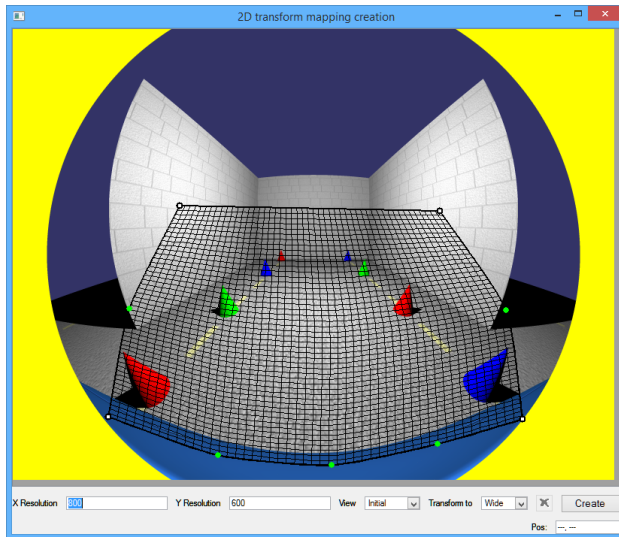


Рис. 3: Сетка точек для контура на рис. 2.

вых, соответствующих левому, правому, нижнему и верхнему границам изображения. Первоначально контур представлен в виде прямоугольника из четырех граничных отрезков. Пользователь может произвольно добавлять и удалять новые точки на этом контуре и перемещать их. Удалять первоначальные угловые точки нельзя. Их можно только перемещать. Каждая кривая строится как сплайн, проходящий через заданные точки. При перемещении точек автоматически поддерживается ограничение, не допускающее пересечения кривых. Для построения отображения, описанного в разделе 2, необходимо на контуре, построенном пользователем на исходном изображении, построить точки, соответствующие однородной сетке производного изображения.

Для этого нами был разработан следующий алгоритм:

1. На верхней и нижних граничных кривых контура строятся наборы точек, равномерно распределенные по X координате. Количество точек определяется заданной размерностью отображения по X координате.
2. Промежуточные наборы точек между верхней и нижней границами контура получаются линейной комбинацией соответствующих наборов точек построенных в п.1 с весами, равными расстояниям до соответствующих границ. Количество этих промежуточных наборов точек определяется заданной размерностью отображения по Y координате.
3. На левой и правой граничных кривых контура строятся наборы точек, равномерно распределенные по Y координате. Количество точек определяется заданной размерностью отображения по Y координате.
4. Естественным условием для построенных в п.2. промежуточных наборов точек является совпа-

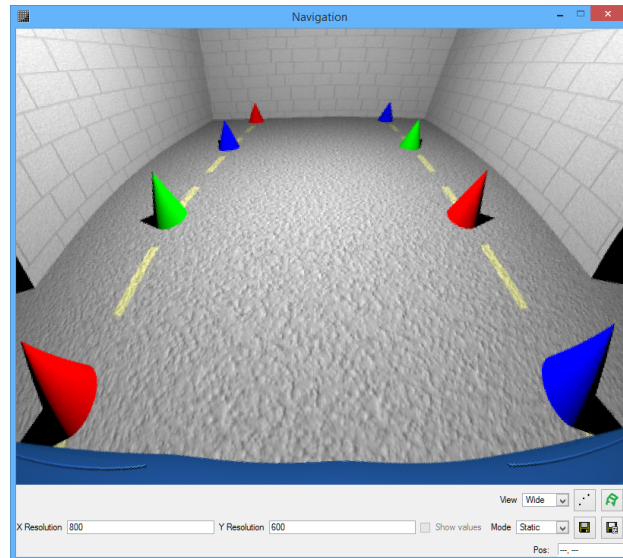


Рис. 4: Преобразованное изображение камеры заднего вида.

дение начальных и конечных точек с соответствующими точками, построенными в п.3.

5. Для выполнения условия из п.4. для каждого набора точек, построенного в п.2. строится матрица 3×2 (смещение, масштабирование и поворот), удовлетворяющая условию п.4 и все точки этого набора точек корректируется с помощью построенной матрицы.

Точки, построенные по описанному алгоритму, показаны на рис. 3 как вершины сетки. Результирующее изображение, полученное с помощью построенного отображения, показано на рис. 4.

4. Построение вида сверху

Описанный в разделе 3 алгоритм позволяет отобразить область внутри любого построенного пользователем контура как изображение, полученное некоторой специфической камерой. Способ построения позволяет не только выделять полезную с точки зрения водителя область изображения, но и производить определенные деформации этой области. Однако этот подход не позволяет получить качественный вид сверху, т.е. имитировать камеру, смотрящую на сцену сверху. Для получения вида сверху из исходного изображения, подаваемого с камеры заднего вида с известными параметрами, была разработана специальная процедура. Эта процедура базируется на следующих предпосылках:

1. Предполагается что поверхность, по которой в данный момент передвигается автомобиль можно рассматривать как плоскость с точки зрения данной задачи. Параметры этой плоскости, рассматриваемой как «земля», известны.

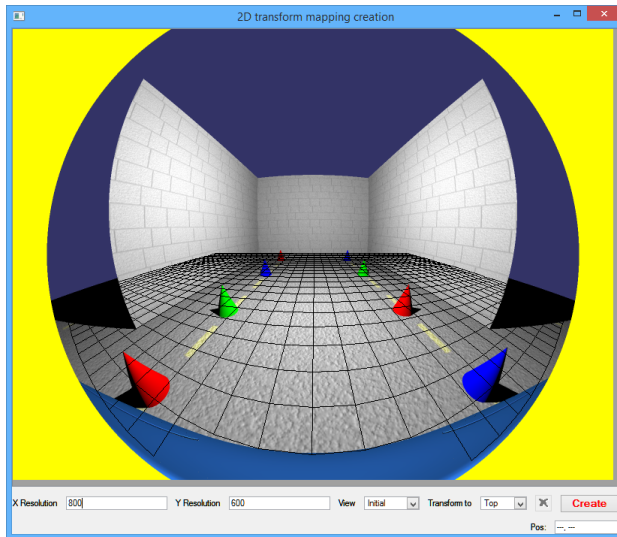


Рис. 5: Сетка для отображения в идеальный вид сверху.

2. Известны все параметры используемой камеры заднего вида, такие как положение, ориентация и функция искажения.
3. Заданы параметры для камеры, смотрящей на сцену сверху. Это может быть, как и идеальная («pinhole») так и ортогональная камера с параллельной проекцией.

Реализация процедуры встроена в систему, описанную в [7], и для вызова соответствующей процедуры пользователь должен указать обе камеры и элемент сцены, который будет использоваться для определения плоскости «земля». Процедура строит 2D отображение, описанное в разделе 2, которое затем используется для построения, так называемого, идеального вида сверху. Для построения отображения нам необходимо для каждого узла сетки на итоговом изображении вида сверху найти соответствующую точку на исходном изображении, полученном с помощью камеры заднего вида. Алгоритм работает следующим образом:

1. Трассируется луч из камеры, расположенной сверху, через пиксель, соответствующий заданному узлу сетки до пересечения с плоскостью «земли».
2. Строится луч из камеры заднего вида в полученную точку пересечения.
3. Находится соответствующий пиксель для этого луча в заданной камере заднего вида с учетом функции искажений. Если такой пиксель не существует (т.е. реальная камера не видит заданную точку пространства), то в качестве координат пикселя будет записано $(-1, -1)$. Программа, строящая преобразованное изображение, отобразит такие пиксели красным цветом.

Точки, построенные по описанному алгоритму, показаны на рис. 5 как вершины сетки. Полученное с

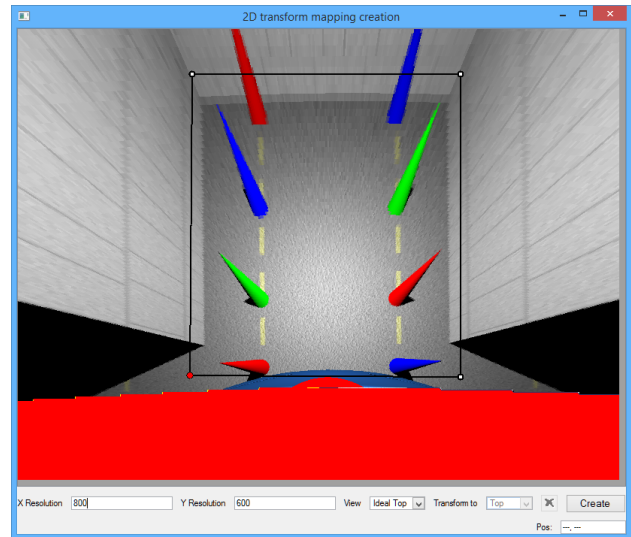


Рис. 6: Идеальное изображение для вида сверху

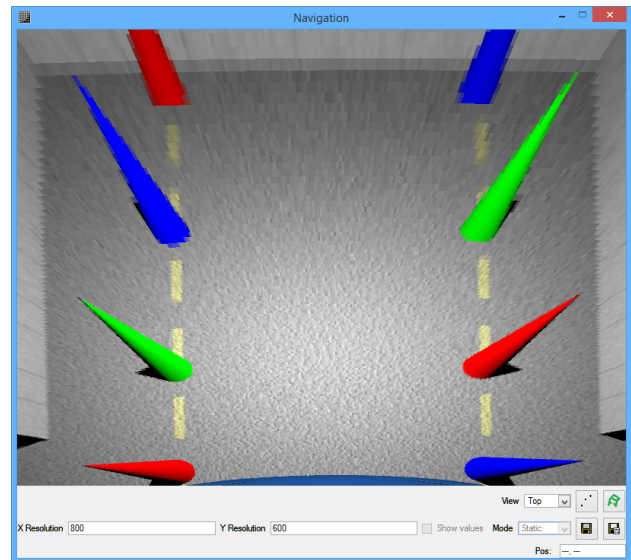


Рис. 7: Результирующее изображение для контура, приведенного на рис. 6.

помощью описанной процедуры идеальное изображение для вида сверху приведено на рис. 6.

Стандартная процедура, описанная в разделе 3, позволяет вырезать из этого изображения наиболее полезную область с точки зрения водителя. Результирующее 2D отображение будет получено в этом случае как произведение двух отображений – отображения, получаемого с помощью алгоритма, описанного в разделе 3, и отображения, описанного в текущем разделе выше. Для контура, изображенного на рис. 6 результирующее изображение приведено на рис. 7.

Следует отметить, что метод построения монитора кругового обзора, предложенный в работе [8] для получения изображения ортогональной (ортографической) камеры, расположенной над автомо-

билем, дает результат, близкий к предложенному методу. Основной недостаток построения монитора кругового обзора состоит в использовании фиксированного набора плоских секторов, копланарных с плоскостью земли. Изображение, полученное каждой камерой, преобразуется в текстуру соответствующего сектора путем обратной проекции изображения на плоскость земли. Таким образом, при использовании только одной камеры, результирующее изображение будет ограничено соответствующим сектором. В то же время в нашем подходе оно будет ограничено только областью видимости реальной камеры заднего вида.

5. Результаты

Разработанные алгоритмы построения 2D отображений были встроены в программный комплекс расчета освещенности в системе CATIA (Dassault systems) [7]. Построенные 2D отображения используются для их реализации в аппаратной или программной части реальной камеры заднего вида или автомобильного навигатора. Время построения 2D отображения не превышает нескольких секунд, время построения 2D отображения для идеального вида сверху может занимать несколько минут, если задать высокое разрешение сетки.

Литература

- [1] Hughes C., Glavin M., Jones E., Denny P. Wide-angle camera technology for automotive applications: a review. IET Intell. Transp. Syst., 2009
- [2] http://techcenter.mercedes-benz.com/ru_BY/360_degree_camera/detail.html
- [3] http://techcenter.mercedes-benz.com/ru_BY/reversing_camera/detail.html
- [4] <http://www.nissan-global.com/EN/TECHNOLOGY/OVERVIEW/avm.html>
- [5] <http://owners.honda.com/vehicles/information/2015/Fit/features/Rearview-Camera->
- [6] Barladyan B.Kh., Shapiro L.Z., Valiev I.V., Voloboy A.G.. Interactive camera distortion correction // Conference proceedings of 22-th International Conference on Computer Graphics and Vision, Lomonosov Moscow State University, October 01-05, 2012, pp.12-16.
- [7] Барладян Б.Х., Волобой А.Г., Шапиро Л.З.. Построение реалистичных изображений в системах автоматизированного проектирования // Труды 23-й Международной Конференции по Компьютерной Графике и Зрению, Институт автоматизации и процессов управления ДВО РАН, Дальневосточный Федеральный Университет, 16-20 сентября 2013 года, С.148-151.
- [8] Валиев И.В., Волобой А.Г.. Моделирование монитора кругового обзора Труды 20-ой международной конференции по компьютерной графике и зрению ГРАФИКОН-2010, 20-24 сентября 2010, Санкт-Петербург, Россия, С.269-272.

Развитие подходов к разработке специализированных систем компьютерной визуализации*

В.Л. Авербух, М.О. Бакhtерев, П.А. Васёв, Д.В. Манаков, И.С. Стародубцев

averbukh@imm.uran.ru

Екатеринбург, Россия, Институт математики и механики им. Н.Н. Красовского Уральского отделения РАН

В работе описывается развитие подходов к проектированию и реализации специализированных систем научной визуализации, разрабатываемых для представления результатов математического и компьютерного моделирования сложных процессов и явлений. Описаны подходы к реализации таких систем в рамках конструктора специализированных систем научной визуализации, декларативно-императивного подхода, и веб-подхода. Представлены результаты визуализации компьютерных моделей, полученные при помощи этих систем. Предложен новый язык для описания сцен научной визуализации, удобный для пользователей – прикладных математиков. Рассмотрены возможности использования в специализированных системах сред виртуальной реальности, средств естественного интерфейса и средств физической визуализации. Также рассмотрена задача верификации визуализации.

Ключевые слова: научная визуализация, виртуальная реальность, естественный интерфейс

Development of Approaches to Realization of Specialized Visualization Systems*

V.L Averbukh, M.O. Bakhterev, P.A. Vasev, D.V. Manakov, I.S. Starodubtsev

Yekaterinburg, Russia, Krasovskii Institute of Mathematics and Mechanics, Ural Branch of the RAS

The paper is devoted approaches to design and development of specialized visualization systems. These systems are used to visualize results of mathematical and computer modeling of complex processes and phenomena. The approaches to realizations of such systems are described. There are among them the approach realized in Constructor of Specialized Visualization Systems, the Declaratively-Imperative Approach, Web-approach based on Qml. Visualization examples are described. A new language to describe scenes of scientific visualization is proposed. This language is user-friendly for applied mathematicians. Opportunities and problems of using in specialized systems such means as virtual reality environments, natural interfaces, and physical visualization are considered. The problem of visualization verification is set.

Keywords: Scientific Visualization, Virtual Reality, Natural Interfaces

Введение

Специализированные системы являются одним из направлений развития средств научной визуализации, обеспечивающих процессы математического и компьютерного моделирования [1, 2, 3, 4]. В течение целого ряда лет мы разрабатывали специализированные системы для различных задач [5, 6, 7, 8]. Были реализованы системы визуализации для ряда задач оптимального управления, математической физики, медицины. Получены результаты в визуализации некоторых многомерных множеств [9]. В течение ряда лет разрабатывалась система визуализации для задач математической физики, связанных с сеточными методами вычислений. В результате удалось представить, в частности, сетку с шестигранными примитивами и сплошную тетраэдральную. Система позволяла визуализировать сетки размером до 10^8 узлов.

В результате анализа опыта разработок была описана методика проектирования и разработки специализированных систем визуализации. В рамках ме-

тодики определены роли участников процесса разработки: заказчик-пользователь, проектировщик визуализации, специалист в методах компьютерной графики и человеко-компьютерного взаимодействия. (Конечно, возможно объединение различных ролей одним человеком.)

Выделены основные позиции схемы проектирования специализированных систем визуализации, в которую, в частности, входит этап проектирования визуализации.

Проектирование визуализации включает в себя учет знаний об абстракции модельных сущностей, обеспечивая тем самым основу для интерпретации различных аспектов модели. Основой визуального проектирования является проектирование видов отображения.

Вид отображения (*view*) определяется как абстракция графического вывода, содержащая спецификацию визуальных объектов, их атрибутов, их взаиморасположения, возможной динамики и способов взаимодействия [10]. При этом визуальные формы абстрактных данных не связаны ограничениями, накладываемыми определенными графическими системами и методами рендеринга. При про-

Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347.

ектировании вида отображения предусматриваются допустимые изменения картинки, анимация, и способы взаимодействия с визуальными объектами. Вид отображения можно понимать как методику визуального представления данных, своего рода, визуальную процедуру, которая при реализации в конкретных визуальных средах и при связывании с реальными данными выводится на графические устройства как графический вывод (*graphical display*), в который переходят атрибуты этого вида отображения.

При разработке видов отображения очень важно мнение заказчика-пользователя, определяющего цели моделирования и задачи визуализации. Особенно велико его значение при проектировании визуализации сложных математических моделей с абстрактной образностью.

К традиционному подразделению типов визуализации на иллюстративную и когнитивную можно добавить еще доказывающую визуализацию. Иллюстративная визуализация показывает уже известные феномены. Когнитивная, способствующая мышлению, визуализация показывает принципиально новые эффекты, которые позволяют получить новое знание [11]. Доказывающая визуализация должна показывать результаты численного исследования, подтверждая (или опровергая) те или иные идеи математика.

Различные типы визуализации определяют выбор видов отображения и способов визуализации. Если при постановке задачи у пользователя нет готовых представлений об образности визуализации, нет соответствующих известных видов отображения, то задача визуализации требует разработки специализированной системы, для которой возможно потребуются и специальные методы рендеринга, и решение особых системных задач.

Среды разработки специализированных систем

Разработка систем визуализации «с нуля» требует решения многих рутинных задач. Необходимо подобрать окружение исполнения и графическую библиотеку, реализовать структуры данных, алгоритмы рендеринга, взаимодействие с пользователем и т. п.

Многочисленное повторение этих действий формирует потребность их автоматизации. В идеале, требуются какие-то высокоуровневые средства описания визуализации, которые позволят заказчику-пользователю и специалистам по визуализации работать эффективнее.

Далее в разделе приводятся результаты нашей работы в этом интересном для нас направлении.

Конструктор систем визуализации

Среда предоставляет пользовательский интерфейс, позволяющий удобно манипулировать имеющейся сценой, а также включает мощное ядро, дающее возможность хранить трёхмерную сцену и динамически работать с ней посредством программного интерфейса (*API*). При этом процедуры восстановления геометрических представлений выносятся во внешние программные модули (*плагины*), что позволяет настраивать систему под конкретные нужды. Среда играет роль «конструктора», позволяющего создавать новые специализированные под конкретные задачи системы визуализации [12, 13].

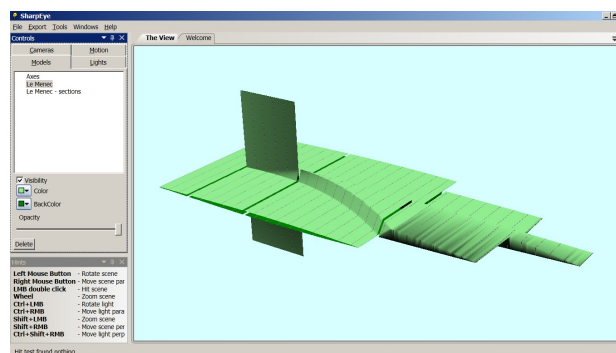


Рис. 1: Внешний вид системы.

Предусмотрен набор классов визуальных объектов, моделей, камер и источников света. Главным типом объектов, с точки зрения программиста, применяющего конструктор, является модель. Модель - это объект трёхмерной сцены, являющийся визуальной сущностью.

По команде загрузки данных программа проводит голосование по списку внешних модулей, определяя, который из них лучше разбирается в загружаемом файле. Модуль-победитель проводит загрузку, формируя сцену с помощью программного интерфейса.

После загрузки данных у пользователя появляется возможность отображения сцены и манипулирования ею. После того, как желаемое изображение получено, его можно экспортировать вовне в растровом или трёхмерном формате.

Система показала свою работоспособность и применяется на практике для решения ряда задач визуализации.

Система написана на языке C# с применением графики WPF (*Media3D*). Язык модулей — .NET-совместимый (на практике использовались C# и IronRuby). Также реализован запуск в качестве модулей внешних процессов с протоколом общения по стандартным потокам. Информация по системе размещена на сайте <http://sharpeye.lact.ru>.

Декларативно-императивный подход

В ходе применения конструктора было установлено, что код модулей на C# и даже на IronRuby зачастую оказывается громоздким. Поэтому авторами были выявлены некоторые типовые шаблоны программирования, используемые при создании модулей, и предложен новый язык для описания сцен научной визуализации [14, 15, 16].

Язык позволяет описать некую древовидную структуру. Виды узлов: строковые значения, двумерные массивы, загрузка файлов, вызов подпрограмм (процессов), передача данных между узлами, шаблоны формирования новых узлов. Эта структура по своей природе декларативна. «Интерактивные» узлы, такие как запуск подпрограмм и загрузка файлов, вычисляются только по требованию. Возможность запуска подпрограмм наделяет язык императивным свойством. Отметим, что подпрограммы могут порождать новые узлы. Пример описания сцены на языке и результат работы приведены на рисунках 2 и 3.

```
1 set scene.model.spheres = array 14000 3 <<file data.txt
2 set scene.model.spheres.radius = 0.05
3 set scene.model.spheres.color = green]
```

Рис. 2: Описание говорит системе интерпретатора: при обращении к переменной `scene.model.spheres` загрузи данные в двумерный массив размера 14000×3 из первых 14000 строк файла `data.txt`, по 3 числа в строке. При обращении к переменной `scene.model.spheres.radius` верни «0.05», а `.color` — верни «green».

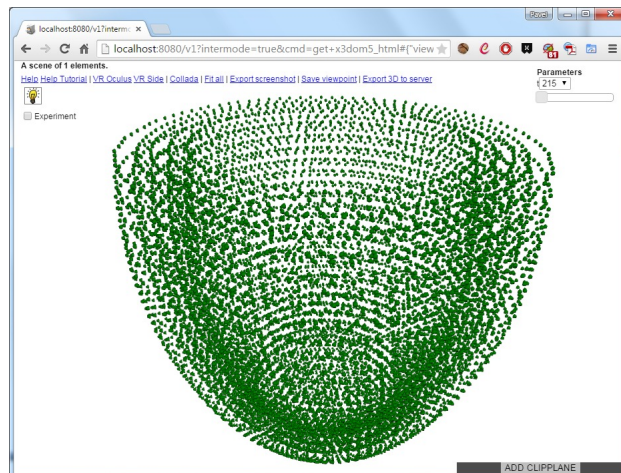


Рис. 3: Система визуализации обратилась к интерпретатору, и по суффиксам имен данных поняла, что необходимо отобразить 1 объект — набор сфер.

Подход показал себя мощным и лаконичным инструментом для создания сцен научной визуализации. Среди его практических применений — задачи

визуализации расчетов работы сердца, визуализация расчетов кристаллических решеток, и другие. Более подробная информация доступна на сайте <http://view.lact.ru>.

Веб-подход на основе Qml

Параллельно с развитием подхода, описанного в предыдущем подразделе, было замечено удобство другого языка, Qml¹, который также является декларативно-императивным и дополнительно — реактивным. Сообщество разрабатывает возможность запуска кодов Qml в веб-браузерах². Одновременно с этим уровень технологий трехмерной графики для браузеров WebGL³ оказался нам достаточно высоким для применения в научной визуализации. Таким образом, существуют предпосылки для альтернативной реализации предложенного подхода, полностью в веб-среде (т. е. в браузере пользователя). В настоящее время это направление находится в стадии развития. Более подробная информация доступна на сайте <http://viewlang.ru>.

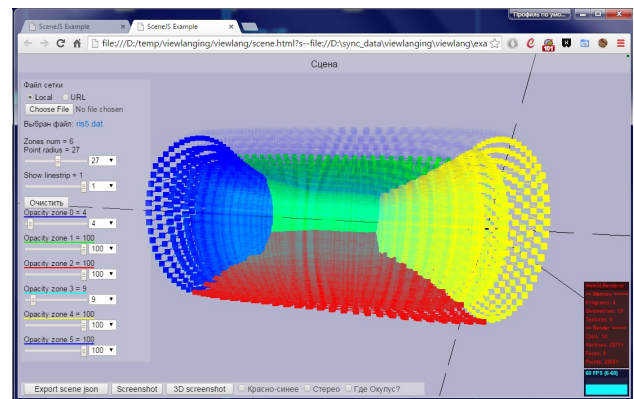


Рис. 4: Загрузка и отображение данных в браузере.

Перспективные исследования и разработки

Использование средств виртуальной реальности

Интерес к использованию сред виртуальной реальности для систем визуализации связан с такими характеристиками, как:

- трехмерность и стереоскопичность;
- динамичность;
- мультисенсорность (подача информации на несколько органов чувств);
- контроль над видимой частью виртуальной среды;

¹Язык QML. <https://ru.wikipedia.org/wiki/QML>

²Проект QmlWeb: <https://projects.kde.org/projects/playground/www/qmlweb>

³WebGL: <https://ru.wikipedia.org/wiki/WebGL>

— возможность обеспечить взгляд изнутри (пространства), обеспечивая погружения в виртуальный мир [17].

Возможности очков виртуальной реальности были использованы в прототипах специализированных систем визуализации для показа изнутри сеток, кристаллов и отдельных участков сердца (рисунки 5 и 6).

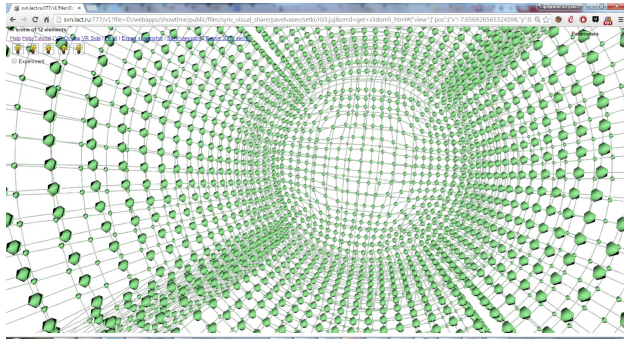


Рис. 5: Отображение модельной сетки «изнутри».

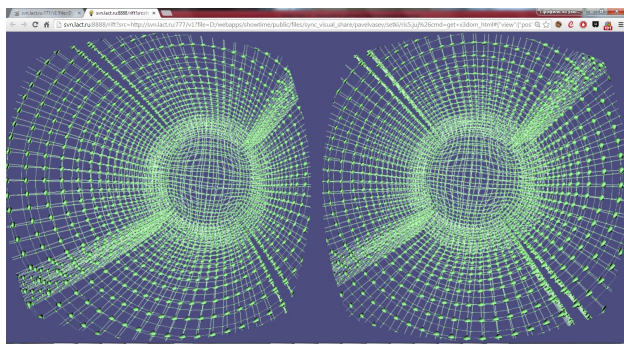


Рис. 6: Сетка с отображением для очков виртуальной реальности «Oculus Rift».

Человеко-компьютерное взаимодействие

Задачи обеспечения взаимодействия возникают при разработке систем компьютерной визуализации на базе сред виртуальной реальности, предполагающих эффект присутствия. При использовании обычных средств ввода при работе также могут возникать проблемы непрерывного перехода от изучения виртуального мира к поиску нужных клавиш или иных устройств. Использование для манипуляций и навигации в виртуальном мире специальных джойстиков также не всегда удобно. Возникает задача разработки новых, удобных для осуществления основной деятельности пользователей человеко-компьютерных интерфейсов [18]. Для ее решения выполнена прототипная реализация системы естественного интерфейса, основанного на распознавании жестов. Проведены испытания средств жестового взаимодействия на базе использования устройства Leap Motion, закрепленного на стерео-очках Oculus Rift (рисунок 7).

Физическая визуализация

Особенности структуры множеств достижимости в задачах управления лучше воспринимаются при их представлении в виде трёхмерных фигур, полученных с использованием 3D-печати. Были начаты работы по использованию технологии 3D-печати для полноразмерного построения подобных множеств. Также предполагается изучение внутренней структуры множеств достижимости с помощью средств виртуальной реальности.

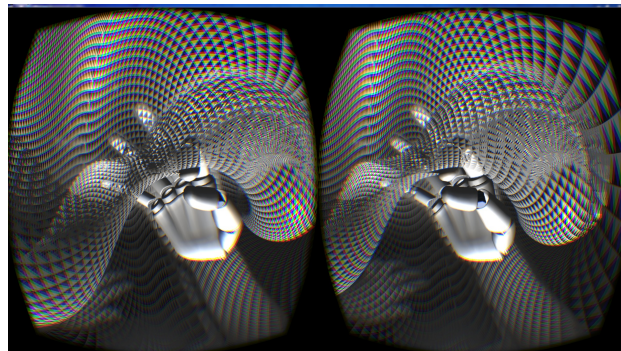


Рис. 7: Работа с сеткой в виртуальной реальности с использованием захвата движений.

Заключение

Научная визуализация обеспечивает этапы анализа и интерпретации результатов математического и компьютерного моделирования. Специализированные системы позволяют увидеть особенности моделируемых явлений и процессов в виде качественной и достоверной «картинки». Достоверность результатов визуализации может быть обеспечена в рамках исследований по верификации и валидации визуализации. Верификация визуализации подразумевает наличие некоторой формальной модели.

Важной задачей является участие математика, проводящего исследование, в проектировании видов отображения для ускорения всего процесса разработки и улучшения качества работы системы визуализации. В настоящее время опробовано три подхода к созданию среды разработки специализированных систем, получен ряд результатов их применения в практике разработки специализированных систем. Сейчас идет работа по реализации систем визуализации сеток с использованием декларативно-императивного подхода и на основе веб-подхода. В рамках разрабатываемых систем предполагается применение сред виртуальной реальности и человеко-компьютерных интерфейсов на базе захвата движений.

Планируются также перспективные исследования и разработки в области многомерной визуализации, задачи которой связаны с моделированием ряда процессов.

Мы рассчитываем на сотрудничество с другими научными коллективами по разработке средств визуализации для различных задач математического и компьютерного моделирования.

Литература

- [1] *dos Santos S., Brodlie K.* Gaining understanding of multivariate and multidimensional data through visualization // *Computers & Graphics*, 2004. – Vol.28, No.3. – pp. 311–325.
- [2] *Bischof H.-P., Dale E., Peterson T.* Spiegel A Visualization Framework for Large and Small Scale Systems // *Proceedings of the 2006 International Conference on Modeling, Simulation & Visualization Methods*, 2006. – pp. 199–205.
- [3] *Wu E., Battle L., Madden S.* The Case for Data Visualization Management Systems // *Proceedings of the Very Large Data Bases Endowment (PVLDB)*, 7(10):903-906, 2014.
- [4] *Godfrey P., Gryz J., Lasek P.* Interactive Visualization of Large Data Sets // *Technical Report EECS-2015-03 March 31 2015. Department of Electrical Engineering and Computer Science. York University. Toronto, Ontario. Canada.*
- [5] *Averbukh V.L., Kumkov S.S., Patsko V.S., Pykhiteev O.A., Yurtaev D.A.* Specialized Visualization Systems for Differential Games // *Progress in Simulation, Modeling, Analysis and Synthesis of Modern Electrical and Electronic Devices and Systems / N.E.Mastorakis (ed).* S.L.: WSES Press, 1999. – pp. 301–306.
- [6] *Зенков А.И.* Разработка подхода к созданию специализированных систем визуализации для высокопроизводительных научных вычислений // *Вопросы атомной науки и техники. Серия: Математическое моделирование физических процессов*, 2003 – Вып.4. – С. 81–86.
- [7] *Averbukh V.L.* The Specialized Systems of Scientific Visualization // *AIP Conference Proceedings*. August 2006. – Vol.849. – pp. 481–486.
- [8] *Авербух В.Л., Байдалин А.Ю., Бахтерев М.О., Васёв П.А., Казанцев А.Ю., Манаков Д.В.* Опыт разработки специализированных систем научной визуализации // *Научная визуализация. Квартал 4. Том 2. Номер 4. 2010. С. 27–39.*
- [9] *Васёв П.А., Первалов Д.С.* О создании методов многомерной визуализации // *Труды 12-й Международной Конференции по Компьютерной Графике и Машинному Зрению ГрафиКон'2002*, Н. Новгород, С. 431–437.
- [10] *Heath M., Malony A., Rover D.* The Visual Display of Parallel Performance Data // *IEEE Computer*, 1995, – Vol.28, No.11. – pp. 21–29.
- [11] *Зенкин А.А.* Когнитивная компьютерная графика. М: Наука, 1991.
- [12] *Васёв П.А.* Проект среды разработки специализированных систем визуализации // *XII Международный семинар «Супервычисления и математическое моделирование»*. Тезисы. Саров. РФЯЦ ВНИИЭФ. 2010. С. 22–23.
- [13] *Васёв П.А., Кумков С.С., Шмаков Е.Ю.* Конструктор специализированных систем визуализации // *Научная визуализация*, 2013. квартал 2, том 4, номер 2, С. 64–77.
- [14] *Бахтерев М.О., Васёв П.А.* Один метод визуализации результатов суперкомпьютерных вычислений // *Материалы III Всероссийской научно-технической конференции «Суперкомпьютерные технологии» (СКТ-2014)*, 2014. Таганрог, НИИ МВС ЮФУ. С. 50–55.
- [15] *Васёв П.А., Бахтерев М.О.* Декларативно-императивный метод конструирования сцен научной визуализации // *XV Международная конференция «Супервычисления и Математическое Моделирование»*. Тезисы. РФЯЦ ВНИИЭФ. Саров. 2014, С. 46.
- [16] *Васёв П.А.* Опыт применения конструктора сцен научной визуализации // *Труды международной научной конференции «Параллельные Вычислительные Технологии» (ПавТ'2015)*. Челябинск, ЮУрГУ. 2015. С. 500.
- [17] *Baker M.P., Wickens Ch.D.* Human Factors in Virtual Environments for the Visual Analysis of Scientific Data // *NCSA-TR032*, August 1995. Tech. Rep. ARL-9508/PNL-95-2. University of Illinois Aviation Research Lab.
- [18] *Авербух В.Л., Авербух Н.В., Стародубцев И.С., Тоболин Д.Ю.* Использование жестовых интерфейсов при взаимодействии с объектами // *Научная перспектива*, 2014 – No.10(56). – С. 57–66.

Физическая визуализация множеств достижимости в задачах управления*

В.С. Пацко, И.С. Стародубцев, А.А. Федотов

patsko@imm.uran.ru | starodubtsevis@imm.uran.ru | andreyfedotov@mail.ru

Екатеринбург, Россия, Институт математики и механики им. Н.Н. Красовского Уральского отделения РАН

Работа посвящена использованию технологии 3D-печати для полноразмерного построения множеств достижимости в задачах управления. В качестве примера рассмотрена нелинейная модель динамики простейшего автомобиля с трёхмерным фазовым вектором и скалярным управлением, ограниченным по модулю (машина Дубинса). Текущее состояние системы включает положение на плоскости и угол направления вектора скорости. Величина скорости считается постоянной. Рассматриваются множества достижимости в момент и к моменту. Они представляют собой невыпуклые множества, граница которых в целом не является гладкой. Особенности структуры рассматриваемых множеств лучше воспринимаются при их представлении в виде трёхмерных фигур, полученных с использованием 3D-печати. Граница множеств достижимости формируется в форматах VRML и STL. Приведены примеры визуализации результатов расчётов, включая вывод на 3D-принтер. Рассмотрены проблемы, возникающие при выводе поверхностей сложной формы на трёхмерную печать.

Ключевые слова: задачи управления, множества достижимости, 3D-печать, FDM-печать, физическая визуализация

Physical visualization of reachable sets in control problems*

V.S. Patsko, I.S. Starodubtsev, A.A. Fedotov

Yekaterinburg, Russia, Krasovskii Institute of Mathematics and Mechanics, Ural Branch of the RAS

The paper deals with a technology of 3D-printing for a full-sized constructing of reachable sets in control problems. As an example, we consider a simple car model with nonlinear dynamics that has a three-dimensional phase vector and scalar control constrained by modulus (the Dubins car). The system state includes a position in the plane and an angle of the velocity vector. The value of the velocity is constant. The reachable sets are investigated both at the given instant and till the given instant. They are non-convex ones, and their boundary is not smooth in the whole. The best interpretation of peculiarities of the sets' structure is provided by their representation in the form of the three-dimensional shapes obtained using the 3D-printing. The boundary of the reachable set is formed in formats VRML and STL. The paper gives examples of visualization of calculation results including 3D-printer samples. The problems associated with the output of the surfaces of complex shapes onto the 3D-printing are considered.

Keywords: control problems, reachable sets, 3D-printing, FDM-technology, physical visualization

Введение

В математической теории управления множеством достижимости $G(t; t_0, z_0)$ в фиксированный момент времени T называется множество, состоящее из всех точек, в каждую из которых управляемая система может быть переведена в оговоренный момент T из начальной точки z_0 , взятой в момент t_0 , при помощи допустимого управления. Под допустимыми управлениями можно понимать кусочно-непрерывные функции времени $t \rightarrow u(t)$, принимающие значения из заданного множества P . Таким образом, если динамика системы описывается стационарным векторным дифференциальным уравнением $\dot{z} = f(z, u)$, $u(t) \in P$, и $z(t; z_0, u(\cdot))$ – положение в момент t решения, выходящего при t_0 из

начальной точки z_0 , то

$$G(T; z_0) = \bigcup_{u(\cdot)} z(T; z_0, u(\cdot)).$$

Объединение берётся по всем допустимым управлениям $u(\cdot)$.

Во многих случаях множество достижимости в классе кусочно-непрерывных управлений является замкнутым. Поэтому нет нужды в расширении класса допустимых управлений. Если для стационарной системы речь идёт о множестве достижимости $G^*(T; z_0)$ к моменту T , то

$$G(T; z_0) = \bigcup_{t \in [0, T]} \bigcup_{u(\cdot)} z(t; z_0, u(\cdot)).$$

Здесь дополнительно осуществляется объединение по всем моментам $t \in [0, T]$.

С появлением 3D-принтеров открылась возможность полноразмерного изображения трёхмерных

Работа выполнена в рамках программы Президиума РАН «Математические задачи современной теории управления» и опубликована при финансовой поддержке РФФИ, грант 15-07-20347.

множеств достижимости. Наличие трёхмерных фигур, сделанных на 3D-принтере, облегчает понимание геометрии множеств достижимости.

Адекватная физическая визуализация снимает «мистику» и связанную с нею «боязнь» использования результатов математической теории управления у студентов и инженеров, занимающихся исследованием и решением прикладных задач.

В то же время, наглядность восприятия может подсказать «разумные» варианты аппроксимации трёхмерных объектов, при которых не теряются их принципиально важные качественные характеристики.

Пример задачи управления

Пусть, например, изучаются множества достижимости в момент t и к моменту для простейшей модели движения автомобиля (машина Дубинса). Динамика записывается в виде

$$\begin{aligned} \dot{x} &= V \cos \varphi, \\ \dot{y} &= V \sin \varphi, \\ \dot{\varphi} &= \frac{k}{V} u, \end{aligned} \quad (1)$$

$$|u| \leq 1, V = \text{const} > 0, k = \text{const} > 0,$$

где x, y – координаты геометрического положения, φ – угол наклона вектора скорости (рис. 1), V – величина скорости, k – максимальное боковое ускорение. Допустимыми управлениями $u(\cdot)$ считаются кусочно-непрерывные функции времени, удовлетворяющие ограничению $|u(t)| \leq 1$. Значения угла φ рассматриваются на промежутке $(-\infty, +\infty)$.

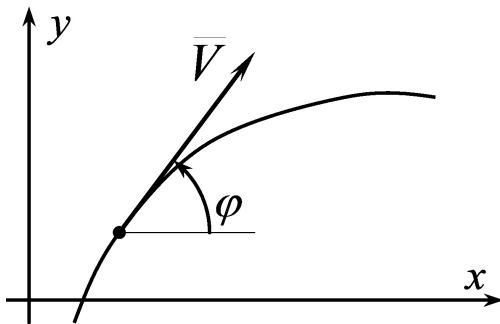


Рис. 1: Система координат.

Фазовый вектор (x, y, φ) системы (1) обозначим через z . Для краткости положим $\alpha = k/V$. Специфика системы (1) такова, что начальное состояние z_0 влияет на множество достижимости лишь с точностью до поворота и переноса. Поэтому полагаем $z_0 = 0$ и пишем $G(T)$, $G^*(T)$ вместо $G(T; z_0)$, $G^*(T; z_0)$.

В статье [1] установлен вид множеств достижимости $G(t)$ в проекции на плоскость геометрических

координат x, y . В статье [2] множества достижимости $G(T)$ исследовались в трёхмерном пространстве x, y, φ .

На рис. 2 показаны множества $G(t)$ для четырёх моментов времени. Видно, что множества не являются выпуклыми. Их граница в целом не является гладкой. При $T \geq \pi/\alpha$ множество $G(t)$ напоминает улитку. С ростом T происходит её «закручивание». В работе [3] изучены множества достижимости $G^*(T)$, а также множества достижимости $G(t)$ и $G^*(T)$ при несимметричных ограничениях на управление u . На рис. 3 показано множество $G(6\pi/\alpha)$ при несимметричном ограничении на управление. Множество достижимости $G^*(1.5\pi/\alpha)$ при симметричном ограничении $|u| \leq 1$ представлено на рис. 4.

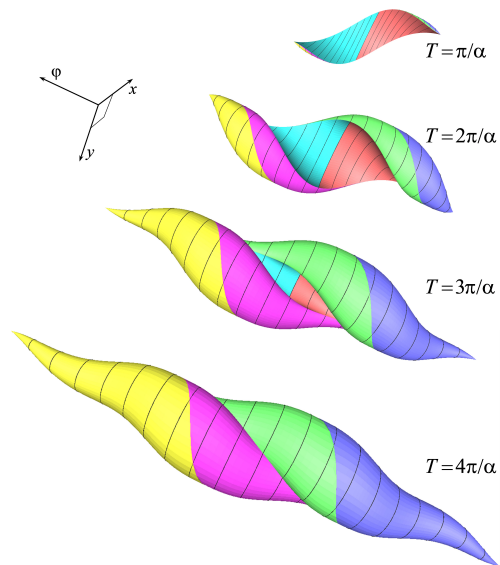


Рис. 2: Развитие множества достижимости $G(T)$.

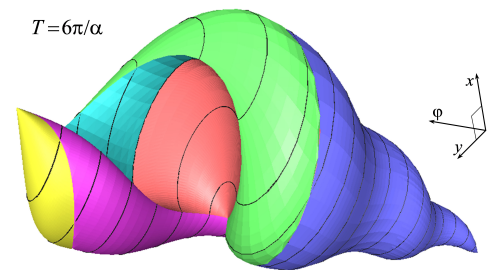


Рис. 3: Множество достижимости $G(T)$ для $T = 6\pi/\alpha$ при несимметричном ограничении $u \in [-1, +0.25]$.

Подготовка данных для 3D-принтера

Для получения в рамках системы (1) компьютерных изображений множеств достижимости ис-

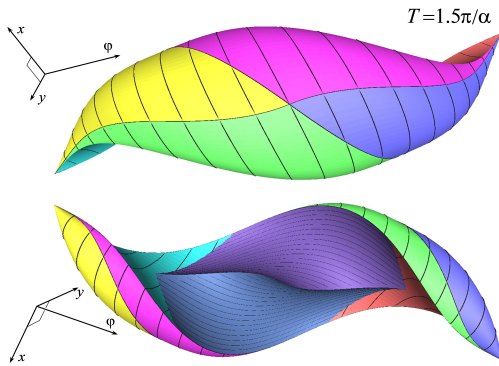


Рис. 4: Множество достижимости $G^*(T)$ для $T = 1.5\pi/\alpha$ в двух ракурсах.

пользовался формат VRML (v2.0). В соответствии с утверждением, доказанным в [2], граница множества достижимости $G(T)$ при симметричном ограничении $|u| \leq 1$ формируется из шести поверхностей, задаваемых в трёхмерном пространстве при помощи двух параметров, в качестве которых используются моменты переключения управления.

Одна из шести поверхностей отвечает значениям $-1, 0, 1$ управления, которые берутся по времени именно в таком порядке. В этом случае параметр t_1 соответствует моменту переключения с управления -1 на управление 0 , а параметр $t_2 \in [t_1, T]$ – с управления 0 на управление 1 . Другие пять поверхностей соответствуют наборам управлений $1, 0, -1$; $1, 0, 1$; $-1, 0, -1$; $1, -1, 1$; $-1, 1, -1$.

Для каждой поверхности задаётся равномерный по времени шаг формирования моментов переключения, в число которых формально включаются моменты 0 и T . Тем самым обеспечивается послойное построение рассматриваемых поверхностей с равномерным по углу φ шагом для всего множества $G(T)$. Это очень важно с точки зрения корректного сопряжения смежных поверхностей. При практическом построении поверхностей интервал $\pm\alpha T$ допустимых значений φ делится на чётное (для охвата нулевого значения) число $2m$ одинаковых отрезков. Для каждого узла на интервале по φ рассчитываются возможные движения системы (1) на промежутке $[0, T]$ из точки z_0 в силу конкретного варианта смены управления.

Шаг перебора моментов переключения для поверхностей, определяемых наборами $+1, 0, +1$ и $-1, 0, -1$, совпадает с шагом разбиения интервала по φ и составляет $\alpha T/m$. Для остальных четырёх поверхностей шаг перебора моментов переключения вдвое меньше.

Специфика поверхностей, образующих границу множества достижимости $G(T)$, состоит в том, что количество точек от слоя к слою по φ отличается не больше, чем на единицу, и один из крайних слоёв состоит из единственной точки. Это использу-

ется для корректного сопряжения слоёв при помощи наборов элементарных треугольных пространственных площадок в процессе формирования соответствующих трёхмерных объектов в форматах VRML и STL.

В случае несимметричного ограничения на управление u отличие состоит лишь в том, что общий интервал допустимых значений φ не является симметричным относительно нуля.

При подготовке к 3D-печати множество $G^*(T)$ составляются данные для построения множества $G(T)$, к которым добавляются наборы треугольных пространственных площадок, сформированные в процессе обработки слоёв, полученных на сетке моментов времени от 0 до T при крайних управляющих воздействиях с одним моментом переключения в качестве параметра.

Таким образом, во всех случаях в текстовый файл в формате STL записывается совокупность поверхностей, содержащая границу множества достижимости. Не лежащие на границе множества части поверхностей находятся внутри формируемого объекта. Триангуляция поверхностей выполняется послойно с использованием свойства непрерывности сечений по φ для множеств достижимости в момент и по t у дополнительных поверхностей для множеств достижимости к моменту.

3D-печать множеств достижимости

Несколько вариантов множеств достижимости в момент и к моменту были подготовлены и распечатаны на 3D-принтере PP3DP UP!Mini. На рис. 5 показаны множества достижимости $G(T)$ в моменты $T = 2\pi/\alpha$, $3\pi/\alpha$, $4\pi/\alpha$.

Исследования в области физической визуализации и, в частности, 3D-печати в последние годы набирают популярность. В работе [4] проводится обзор применения таких технологий в различных областях науки и медицины.

Методы 3D-печати предъявляют более жесткие требования к визуализируемому объекту в сравнении с визуализацией при помощи традиционных средств компьютерной графики.

Воспользуемся следующими понятиями:

- *Слайсер* – программа перевода 3D-модели в управляющий код для 3D-принтера. Она необходима, так как принтер не сможет загрузить сразу 3D-модель. Существуют принтеры, у которых слайсер является частью внутреннего ПО.
- *Слайсинг* – процесс перевода 3D-модели в управляющий код.

Выбор технологии печати

Существует несколько видов 3D-печати, использующих различные технологии и материалы. Для



Рис. 5: Множества $G(T)$ для $T = 2\pi/\alpha, 3\pi/\alpha, 4\pi/\alpha$.

визуализации множеств достижимости была использована технология FDM (Fused Deposition Modeling) печати расплавленной нитью ABS пластика.

Модель режется (слайсится) по слоям. Каждый слой состоит из периметра, заливки и/или поддержки (рис. 6). Модель может иметь разный процент заполнения заливкой. Возможно и отсутствие заливки (пустотелая модель).

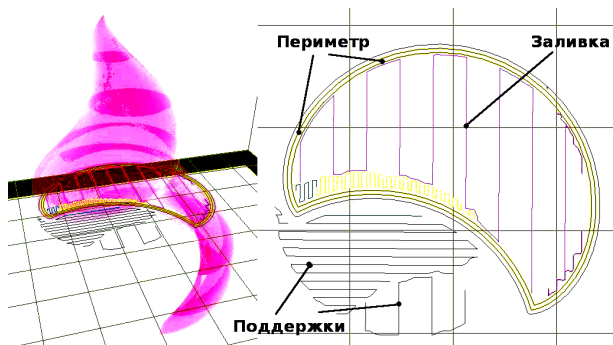


Рис. 6: Слой, получившийся в результате слайсинга.

На каждом слое происходят перемещения по осям XU с нанесением расплава пластика. После печати одного слоя происходит перемещение по оси Z на слой выше, печатается следующий слой и так далее.

Некоторые ошибки модели

Перечислим основные проблем, которые были решены при подготовке множеств достижимости к печати.

Инвертированные нормали

В файлах моделей формата STL нормали задаются в явном виде. В формате VRML (v2.0) нормали рассчитываются с учётом порядка вершин. Наличие

инвертированных нормалей может привести к разным последствиям, среди которых: неправильный расчёт толщины стенки; неправильный расчёт области внутренней заливки; невозможность построения автоматических поддержек.

Неоднородная сетка

1. Отверстия в сетке возникают тогда, когда не создана грань или не заполнена отдельная часть геометрии. Для однозначного расчёта областей печати необходимо выполнение условия «водонепроницаемости» модели, т. е. компактности сечений множества.

2. Внутренние грани – это грани, располагающиеся внутри модели. Модель может быть «водонепроницаемой», но присутствие внутри сетки внутренних граней приводит к неверному расчету внутренней заливки.

3. Налагающиеся грани появляются в случае, когда создаётся дополнительная поверхность поверх уже существующей. Такие грани иногда трудно обнаружить по причине того, что они могут опираться на одни и те же вершины.

4. Общие рёбра связывают более двух граней. При этом может возникать противоречие в определении множества граничных / внутренних точек сечения и множества точек периметра.

5. Геометрия нулевой толщины возникает при наличии нескольких близко расположенных граней, параллельных или имеющих малый двугранный угол между собственными плоскостями. Из-за дискретности слайсинга это может привести к появлению артефактов или невозможности печати некоторых частей объекта.

На рис. 7 показан пример некорректного (из-за ошибок модели) формирования слоёв одного из вариантов множества достижимости в процессе слайсинга. Вследствие наличия инвертированных нормалей и внутренних граней были неверно рассчитаны внешний и внутренний периметры. Множество внутренних точек также было рассчитано неверно из-за наличия незамкнутого контура на внутренней геометрии. Синими линиями отмечены пути с совпадающими рёбрами и частично налагающимися гранями.

На рис. 6 была приведена та же самая модель множества достижимости, но уже после учёта указанных ошибок.

Особенности FDM-технологии печати

Ниже идёт описание того, что нельзя назвать ошибками. Скорее, это особенности технологии, на которые необходимо делать поправку для получения качественного результата.

1. Толщина стенки. Стенки должны быть равными или толще, чем диаметр сопла принтера. Иначе

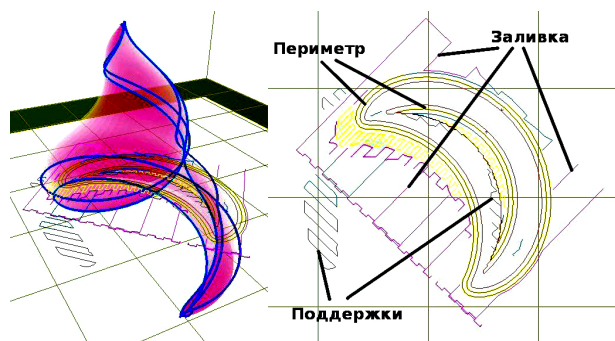


Рис. 7: Пример некорректного формирования слоёв вследствие ошибок модели.

принтер просто не сможет их напечатать. Толщина стенки зависит от диаметра сопла и числа печатаемых периметров. Так, при трёх периметрах и сопле 0.5 мм толщина стенок должна быть от 0.5, 1, 1.5, 2, 2.5, 3 мм, а свыше может быть любой. Т. е. толщина стенки должна быть кратна диаметру сопла, если она меньше $N \cdot d$, где N – количество периметров, d – диаметр сопла.

2. Ориентация объекта. Конечный результат печати объекта может зависеть от ориентации объекта. Отметим следующее:

- Из-за дискретности процесса слайсинга на печатаемой поверхности отображается горизонтальное наслоение, что является неотъемлемым процессом 3D-печати по технологии FDM. Можно либо изменить пространственную ориентацию модели для минимизации влияния наслоения, либо использовать его, чтобы подчеркнуть грани модели.

- Цилиндрические грани, напечатанные с боковой поверхности, могут быть менее округлыми, нежели такие же грани, напечатанные в вертикальной ориентации.

- Прочность/длина: тонкие и прямые элементы будут прочнее при условии их печати в горизонтальном расположении, нежели в вертикальном. Это связано с тем, что связи между слоями являются менее прочными, чем внутри одного слоя.

3. Нависающие элементы. Для каждого нависающего элемента (пример показан на рис. 8) необходима поддерживающая конструкция – *поддержка*. Чем меньше нависающих элементов, тем меньше нужно поддержек. Стало быть, уменьшаются трата материала и время печати. Кроме того, поддержка портит поверхность, соприкасающуюся с ней.

4. Точность. Точность по осям XU зависит от люфтов, жёсткости конструкции, ремней и другой механики принтера. Она составляет примерно 0.3 мм для непрофессиональных принтеров. Точность по оси Z определяется высотой слоя (0.1–0.4 мм). Поэтому высота модели будет кратна высоте слоя.

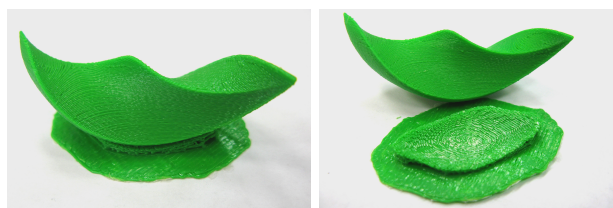


Рис. 8: Множество достижимости к моменту $T = \pi/\alpha$. Пример создания устойчивой поддержки при печати.

Существует и программная сторона проблемы: не каждый слайсер корректно обрабатывает внутренние размеры. Поэтому диаметр отверстий лучше увеличить на 0.1 – 0.2 мм.

5. Изменение геометрии. Необходимо учитывать, что после остывания материал усаживается, поэтому изменяется геометрия объекта. На больших объектах это может привести к «загибанию» краев объекта, в некоторых случаях – даже к отрыву печатаемой модели от стола печати.

Другой аспект термоусадки – появление внутреннего напряжения в материале. Если сделать сплошную заливку модели, то при достаточной площади сечения силы внутреннего напряжения могут привести к искажению (вплоть до разрыва) распечатанной модели. Поэтому применяется частичное заполнение модели материалом в виде различных конфигураций равномерной сетки. При этом обеспечиваются:

- Компенсация внутренних напряжений.
- Добавление прочности модели. Частичное заполнение формирует рёбра жёсткости, которые позволяют еще неостывшим мягким частям модели не деформироваться под собственной тяжестью.
- Уменьшение количества материала и времени печати.

При печати множеств достижимости чаще всего использовались конфигурации внутреннего заполнения в виде *прямоугольной сетки*. В этом случае требуется наименьшее количество материала и времени. На рис. 9 приведён срез распечатанного объекта, на котором видна такая структура заполнения.

Некоторые из описанных особенностей учитываются автоматически (на уровне слайсера или внутренней программы принтера). Например, формирование структуры внутреннего заполнения или расчёт элементов поддержки, как правило, отдаются на автоматический расчёт, а пользователем контролируются только такие параметры как «плотность», «частота» шага и угол скоса, при котором необходимо создавать поддержку.

Другие особенности необходимо обрабатывать вручную, так как в случае неоднозначности (дыры

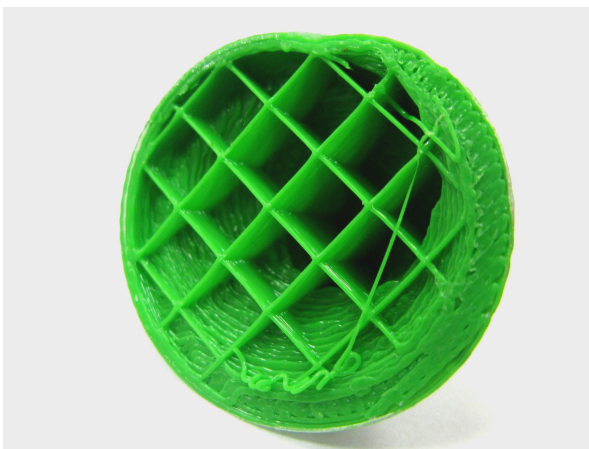


Рис. 9: Срез распечатанной модели.

геометрии, неоднородности сетки, неверные нормали) результат автоматической обработки может быть непредсказуемым.

В настоящий момент проводятся исследования, направленные на автоматизацию процесса подготовки абстрактных математических множеств к физической визуализации.

Заключение

В статье описываются результаты практических расчётов множеств достижимости и их воспроиз-

ведение посредством 3D-печати для динамического объекта, известного, как «машина Дубинса». Множества достижимости имеют нетривиальную геометрию. Тем не менее, в результате обработки полученных пространственных объектов удалось добиться не только весьма хорошего качества пространственного изображения объектов, но и формулировки критериев, необходимых для успешной физической визуализации.

Благодарность

Авторы признательны В. Л. Авербуху за внимание к работе.

Литература

- [1] *Cockayne E. J., Hall G. W. C.* Plane motion of a particle subject to curvature constraints // *SIAM J. Control*, 1975. – Vol. 13, No.1. – pp. 197–220.
- [2] *Пацко В. С., Пятко С. Г., Федотов А. А.* Трёхмерное множество достижимости нелинейной управляемой системы // *Известия РАН. Теория и системы управления*, 2003. – № 3 – С. 8–16.
- [3] *Fedotov A. A., Patsko V. S., Turova V. L.* Reachable Sets for Simple Models of Car Motion / Ed. by A. V. Topalov. Rijeka: InTech Open Access Publisher, 2011. – pp. 147–172.
- [4] *Horvath J.* Mastering 3D Printing. – 1st edition. – Berkely, CA, USA: Apress, 2014.

Простой и недорогой подход к реализации регенерации путей*

В.А. Фролов^{1,2}, В.А. Галактионов¹

vfrolov@graphics.cs.msu.ru | vlgal@gin.keldysh.ru

¹Институт прикладной математики им.М.В.Келдыша, Москва, Россия;

²Московский государственный университет им.М.В.Ломоносова

Монте-Карло Трассировка Путей является центральным алгоритмом расчёта освещенности, вокруг которого строятся более современные методы (такие как BDPT, MLT, ERPT, VCM и другие). Одна из основных проблем, стоящих на пути к реализации эффективной трассировки путей на GPU – малая загрузка GPU вычислениями вследствие сильно различной глубины трассировки: небольшое число потоков трассируют пути на большой глубине, в то время как остальные потоки простаивают. Обычно для решения этой проблемы используется техника, называемая регенерацией путей. Мы предлагаем новый подход к реализации регенерации путей, названный нами «блочной регенерацией по месту». По сравнению с предыдущими подходами наш алгоритм обладает более низкой стоимостью и не перемещает данные лучей в памяти, за счёт чего упрощается его интеграция в более сложные методы (такие как BDPT, MLT, ERPT, VCM и другие) или фотонные карты. Мы протестировали наш алгоритм для Монте-Карло трассировки путей, используя CUDA и OpenCL.

Ключевые слова: OpenCL, трассировка путей, регенерация пути

Cheap and simple in-place block based path regeneration*

V.A. Frolov^{1,2}, V.A. Galaktionov¹

¹Keldysh Institute of Applied Mathematics RAS, Moscow, Russia;

² Moscow State University, Moscow, Russia

Monte Carlo Path Tracing is a core light transport technique which is used for modern methods (like BDPT, MLT, ERPT, VCM and others). One of the main challenge of efficient GPU Path Tracing implementation is inefficient workload caused by paths of different lengths; few threads process the long paths, while other threads are idle. A work distribution technique called “Path Regeneration” is commonly used to solve this problem. We introduce a novel GPU implementation of path regeneration technique called “in place block based path regeneration”. In comparison to previous approaches our algorithm possesses two main advantages: it has lower self-cost and it does not move any per-ray data along threads in memory, thus, our algorithm can be easily integrated to any advanced path tracing technique (like BDPT, MLT and other) or photon mapping. We tested our solution with path tracing using both CUDA and OpenCL.

Keywords: OpenCL, Path Tracing, Path Regeneration

Path tracing (PT) generates Monte Carlo samples by simulating light transport via random path walk in the scene. A path starts with a primary ray at camera, traces in to the scene, randomly reflects several times and finishes at light, some Lambertian surface or an environment. The other light transport methods start a path at the light (Light Tracing, Photon Mapping) or both at the eye and light (BDPT). Due to the stochastic nature of all modern light transport algorithms, an effective GPU implementation with high trace depth becomes a challenge: deep reflection levels have only several active paths while GPU has to execute mostly all of them. Due to SIMD nature even inactive threads (i.e. terminated paths) have to be executed, as long as there is at least one active thread in their warp. Moreover, a completely dead warp could still waste multiprocessor resources reading “inactive” flag from intermediate data in DRAM (see “multiple-

kernel” further) or due to inefficient work distribution implementation in driver when the whole block is still occupies multiprocessor resources until all of its warps become inactive (one possible solution refers to “persistent threads” [1]).

Introduction

Single and multiple kernel There are two main software architectures for GPU path tracing implementation. They are “single kernel” and “multiple kernel”. The naive “single kernel” implementation generally holds the whole algorithm in a single large kernel, essentially the same as a standard CPU path tracer implementation. Such implementation is inefficient due to the limited number of registers on GPUs and register spilling via local memory [2]. The “Uberkernel” is modification of naive “single kernel” implementation that saves GPU registers converting traditional CPU-based code to a state-machine where huge and complex code splits to some simpler parts – each part per one state. NVIDIA OptiX [3] uses this way. The multiple-kernel moves these parts to different ker-

Работа выполнена при финансовой поддержке РФФИ, грант 13-01-00454; Президентской программы ПП-4053.2013.5. и опубликована при финансовой поддержке РФФИ, грант 15-07-20347

nels, which leads to more efficient register usage and significant performance gain for critical code such as BVH traversal [2]. However, “multiple-kernel” implementation has to store explicitly intermediate per-ray data (position, direction, hit normal, material reference, some flags etc.) to DRAM. Despite very simple scenes, it does not affect the performance but can significantly limit flexibility of light transport algorithm implementation. For example, simulating recursion via stack is easy in “single kernel” implementation but is very difficult in “multiple kernel”. Since in “multiple kernel” implementation per-ray data addresses are usually strictly bound to thread index, the other good examples where “multiple kernel” complicate things are rays compaction, rays sorting, path regeneration. During our research, we additionally found out that OpenCL kernel compiler on some devices (HD 5770, Intel and AMD CPUs) failed to compile huge single-kernel path tracing or even multiple-kernel path tracing with complex materials. Thus, “multiple-kernel” path tracing is the natural way of splitting code complexity to prevent “fresh” OpenCL compilers from unexpected faults.

Regeneration overhead

Thus, both single and multiple-kernel PT have their advantages and disadvantages. Wanting to have a stable multi-platform implementation, we prefer “multiple-kernel” approach rather than a “single kernel”. Then we found that with “multiple-kernel” PT implementation existing path regeneration approaches were not stable in the sense of performance gain: while several heavy and complex scenes benefit from path regeneration, the majority of them are not. The reason was that the PT regeneration wins less performance for tracing rays than its self-cost. In some cases (fig. 3,4) total performance went down. Thus, the primary motivation for our research was to propose lite-weight path regeneration algorithm we can use without fear for degrading performance. The second motivation was flexibility and simplicity. Having big plans for advanced Monte Carlo techniques we would like to have simple and flexible PT core. The existing path regeneration approaches lack of these properties.

Related work

Path regeneration

Novak et al. [4] introduce path regeneration technique restarting the terminated paths and tracing additional ones. The newly generated paths for same thread come always from the same pixel when anti-aliasing or DOF enabled. Major disadvantage of this approach is branch divergence growth: regenerated rays (which may be coherent for several bounces) mixed with incoherent rays in the same warp. This leads to significant performance penalty (2.x-3.x) for coherent rays and eliminates performance gain of regeneration.

The Streaming PT regeneration approach [5] uses compaction to move all active threads in the beginning and fills inactive threads with new rays/paths. The coherent primary rays of the new paths in this approach assigned to threads that executed together. However, this approach has other disadvantages. They are self-cost and complexity:

1. While the self-cost of regeneration approach from [4] is near zero, compaction from [5] could simply eat 10-20% of ray tracing performance. Thus, the typical regeneration gain of 20-30% is suppressed by algorithm self-cost. This is mostly because of the forced necessity to move per-ray local data in memory or to use indices. When “multiple-kernel” PT implementation considered, each per-ray data (like ray position, path color, path throughput and etc.) location is bound to thread index. For example, we can read path color in analogue to (a). The compaction changes actual thread index for active paths, so data must be moved or indices should be used like (b). Beside additional indirection level indices in this way break memory coalescing.

(a) `color = in_color[threadId];`

(b) `color = in_color[original_threadId[threadId]];`

2. An attempt to use indices for only several attributes (many of per ray data live only one bounce) forces us to remember which data are affected and which are not. Each time we read attribute we must recall what to use: (a) or (b).

3. More complex light transport algorithms (like BDPT/VCM or MLT) make compaction even trickier and slower. The necessity of storing per-path vertex data replaces simple indices with lists (because single path may change its thread index several times) and makes “moving alternative” more expensive due to larger per-ray data.

To address issue of inefficient global loads and stores when path regeneration used with “multiple kernel” Davidovic et al. use “single-kernel” path tracing [6]. We discussed “single kernel” disadvantages earlier. Moreover, implementation from [6] has approximately the same performance as [5].

Wald [7] concluded that terminated threads in a warp incur no major performance penalties due to the remaining threads executed faster. According to [7] “certain limitations of today’s hardware lead to sources of overhead that significantly affect the final outcome, eventually leading to disappointingly small speed-ups of only 12–16% for even the best performing of our kernels”. This slightly differs from [4, 5, 6] results, but it is consistent with our experiments – existing approaches do have too much overhead. We should notice that there could be at least two reasons for low path regeneration efficiency in [7]; they are simple materials and low path length (maximum 8). Laine et al. [8] further noticed the first reason and introduced “wavefront path tracing” aiming efficient

evaluating extremely complex BSDFs with “multiple kernel” implementation. However, only “small performance benefit” from path regeneration is still marked in their work.

Proposed method relations

Before considering proposed algorithm we should mention two techniques that are not directly related to path regeneration, but, in general, related to efficient path tracing implementation on GPU. They are tile-based work distribution for path tracing [2] and “per-warp Russian Roulette” [9].

The tile-based work distribution [2] splits an entire screen to 16x16 tiles and use them as an atomic unit of work distribution. Tiles are important because they reduce work-distribution algorithm self-cost by dividing actual operations number by 256 (16x16).

Per-warp Russian roulette introduced in [9]. The original Russian roulette randomly terminates a path to restrict its depth – trace fewer paths on high depth but takes them into account with greater weight. For GPU this additionally increases amount of “sparse warps” [4] with dead threads. Per-warp Russian roulette decides to terminate (or not to terminate) the whole warp by slightly changing weights computation scheme.

By combining tiles idea from [2] and “per-warp Russian roulette” from [9] we introduce a new path regeneration method that is as simple and flexible as approach from [4], as efficient as [5] in the sense of ray tracing performance gain, and cheap in the sense of self-cost.

Suggested approach

The proposed algorithm is designed under the assumption that to achieve high actual occupancy we don’t need completely dense thread pool. There is no significant performance penalty if some continues gaps (25-50% of total tiles number in thread pool) with dead threads exist in it. Our algorithm combines several ideas:

1. Similar to [2] we subdivide screen to tiles of 16x16 pixels. Thus, pixels are separated from threads. Each screen tile is mapped to continuous sequence of 256 rays in thread pool.
2. Similar to [4] we perform “in place” path regeneration, assigning new paths to dead threads. In contrast to [4] we never regenerate single thread but always regenerate the whole 16x16 tile simultaneously, i.e. all 256 continuous threads. A thread block always processes other screen tile when it has been regenerated.
3. To terminate the whole block we extend “per-warp” Russian roulette to a “per-block” Russian roulette. This efficiently terminates a block of threads allowing a new tile to come in the same

place (replacing dead tile) when low actual occupancy met for an old tile.

4. During path tracing we store for each tile its maximum trace depth and sort all tiles according to that value before starting next pass. Thus, tiles with high trace depth always come to thread pool first and hang there as long as they are needed. In contrast to that, tiles with low depth come much later and usually process several bounces only. Such strategy reduces total number of kernel launches in range of 30 – 60% depending on scene and maximum trace depth.
5. We don’t regenerate each bounce. For each second bounce we check a number of tiles we want to regenerate. If this number is greater than a threshold (half of a current active tiles number), we invoke regenerate kernel. We call this “threshold based regeneration”.
6. Particularly, for ray-tracing kernels we use local thread compaction via shared memory inside the 256-thread block. Because these kernels are very light-weight in the sense of reading and writing per-ray data, local thread compaction has no valuable overhead. However, due to compaction reduces sparse warps number it speeds up ray tracing performance in average by 8-15%. Such local thread permutations inside block do not lead to per-ray data movements because it changes thread indices only inside trace kernel.

Thus, when too many tiles are dead, according to point (5) we regenerate (fig. 1). Threshold based regeneration (5) decreases regeneration code cost and increases actual occupancy due to when regeneration kernel launched, we know that significant number of continues sequences of threads does regenerate, thus, we prevent empty kernel calls when a kernel is launched but no threads are doing actual work.

```

F E A B 6 7 9 5 D 3 2 1 0 4 8 C

X X X X X X X X | F E A B 6 7 ...
F E A B 6 7 9 5 | D 3 2 1 8 C
F E A B 6 7 9 5 | D 3 2 1 8 C
F E A B 6 7 9 5 | D 3 2 1 8 C (9,5)->(D,3)
F E A B 6 7 D 3 | 2 1 0 4 .. (A,6,7)->(2,1,0)
F E 2 B 1 0 D 3 | 4 8 C
F E 2 B 1 0 D 2 | 4 8 C (1,0)->(4,8)
F E 2 B 4 8 D 2 | C (4,8)->(C,X)
F E 2 B C X D 2 | (C)->(X)
F E 2 B X X D 2 | (D,2)->(X,X)
F E 2 B X X X X |
...
X X X X X X X X |

```

Listing 1. Simplified algorithm walkthrough (without “threshold-based regeneration”). A thread buffer holds 8 tiles (8x256 rays). The total number of tiles is 16. The left

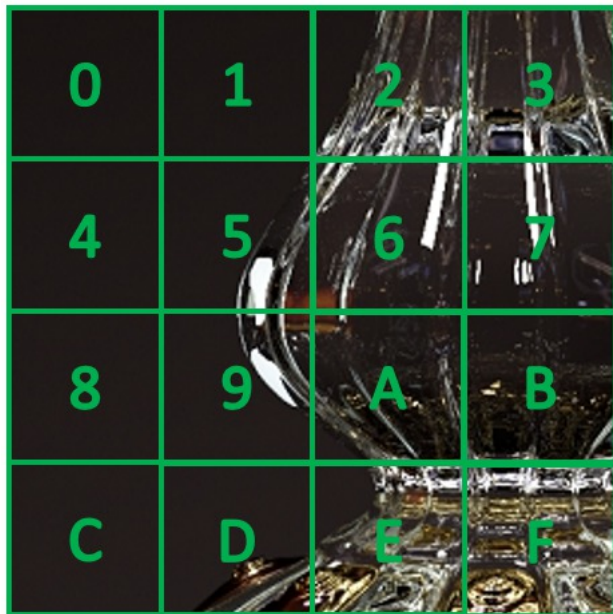


Figure 1: An example of image subdivided to 16 tiles. Each tile is enumerated with hexadecimal number from 0 to F. We sort all tiles basing on their maximum trace depth. Tiles E and F have very large maximum trace depth. They come to the thread buffer first and finish last. Opposite to that tiles 0,1,4,8,C have low maximum trace depth equals to one. They come last and live just 1 bounce in the buffer. Thus, we reduce total kernel calls number.

part represents active tiles in thread buffer. The right part after the slash is a queue of tiles that waiting for process. Symbol “X” represents dead tiles. The notation “(9,5)->(D,3)” means that tiles “9” and “5” were died and new paths were regenerated replacing “9” with “D” and “5” with “3”.

Implementation details

We used the same kernel set in CUDA and OpenCL. Our path-tracing pipeline has 7 kernels in total. They are: regenerate, trace, surfEval, lightSample, shadowTrace, shade, nextBounce. Such fine-grained “multiple kernel” was used due to heavy feature list. Merging kernels in this case leads to poor performance and unexpected faults for OpenCL on some drivers. Almost all of our kernels are heavy in the sense of code complexity: surfEval kernel has Parallax Occlusion Mapping for surfaces with relief; lightSample implements huge sets of light including complex lights with HDR environment and IES distribution; Shade and nextBounce use material tree for flexible material architecture with different type of BRDS in leafs and different types of blending in non-leaf nodes. This is like a “Mila material” in Mental Ray [10]. Thus, in our case not only tracing rays was a kernel that benefits from path regeneration.

Results and discussion

Test setup

Our test setup focuses on different cases when performance benefit should come from different reasons (Fig.A-D). Test # 1 has simple geometry but infinite possible reflections. It is an ideal way to compare different regeneration (and path tracing in general) implementations efficiencies and overheads with no relation to ray tracing performance. Test # 2 is simple in all senses and should not benefit of regeneration at all. It helps us to measure pure regeneration overhead. Tests # 3 (specular/glossy walls) and # 4 are typical real-world examples when path regeneration is applied due to high maximum trace depth.

Comparison to previous approaches

We compare our approach to Streaming PT from [5] (call it “sm” on Fig. 2-5) and path tracing with no regeneration at all (“no regeneration”). For both Streaming PT and “no regeneration” PT we used per block Russian roulette. For Streaming PT we used indices instead of moving per-ray data and each 4-th/8-th bounce for regeneration to have approximately the same regeneration kernel call number on each test scene that we got in our implementation. Otherwise, (each bounce regeneration) Streaming PT costs much and gives us no benefit at all. We used “thrust::copy_if” implementation of Streaming compaction in CUDA and self-implemented compaction based on NVSDK prefix sum sample in OpenCL.

Compared to [5] our algorithm has same performance gain (Test 3, 4) from regeneration but it has lower overhead (Test 1, 3, 4). Main reasons for this are points (1-2), (4), (5). Blocks/tiles (1-2) reduce operation cost by factor of 256. It allows in-place regeneration, though, no per-ray data movement or indices is required. Blocks sorting (4) decreases total kernel calls number. Threshold base regeneration (5) additionally decreases regeneration code self-cost. Besides performance benefit, our algorithm is simpler for implementation because it doesn’t change actual thread indices for rays. Once a tile comes into the thread pool, all its rays gain fixed thread index which will not be changed during any number of bounces. So, per-ray data could be stored at fixed memory locations.

Conclusion

Our experiments showed that path regeneration gives essential benefit only for large trace depth values (greater than 20). For low trace depth values our results in general are consistent with Wald experiments [7].

Thus we believe that reduction of thread divergence within warps (solved via regeneration in [4], [5]) could be successfully solved with per-warp/per-block Russian Roulette only. Path regeneration is needed only when active warps number became lower than the value of warps that GPU process simultaneously with high actual occupancy.

Bibliography

- [1] K. Gupta, J. A. Stuart, J. D. Owens. A Study of Persistent Threads Style GPU Programming for GPGPU Workloads // In proceedings of Innovative Parallel Computing. May 2012. San Jose, CA.

- [2] *Frolov V., Kharlamov A., Ignatenko A.* Biased Global Illumination via Irradiance Caching and Adaptive Path Tracing on GPUs. // Proceedings of GraphiCon'2010 international conference on computer graphics and vision. St. Petersburg, 2010, pp. 49–56.
- [3] *S. G. Parker, J. Bigler, A. Dietrich, H. Friedrich, J. Hoberock, D. Luebke, D. McAllister, M. McGuire, K. Morley, A. Robison, and M. Stich.* OptiX: a general purpose ray tracing engine // In ACM SIGGRAPH 2010 papers: ACM, NY, Article 66, 13 pp.
- [4] *Nocak, J., Havran, V., and Daschbacher* Path regeneration for interactive path tracing // In EUROGRAPHICS 2010, short papers, pp.61–64.
- [5] *Van Antwerpen.* Unbiased physically based rendering on the GPU // M.S. thesis, Delft University of Technology, the Netherlands, 2011.
- [6] *Tomas Davidovic, Jaroslav Krivanek, Milos Hasan, and Philipp Slusallek.* Progressive Light Transport Simulation on the GPU: Survey and Improvements // ACM Trans. Graph. 33, 3, Article 29 (June 2014), 19 pages.
- [7] *Wald, I.* Active thread compaction for GPU path tracing // High Performance Graphics, 2011, pp. 51–58.
- [8] *Laine, S., Karras, T., and Aila, T.* Megakernels considered harmful: Wavefront path tracing on GPUs // Proc of High-Performance Graphics 2013, pp. 137–143.
- [9] *Novak J.* Global Illumination Methods on GPU with CUDA // MS Thesis: Ph. D. thesis.: Czech Technical University, Prague, 2009.
- [10] *NVIDIA* Mental Ray // 2014. <http://www.nvidia.com/object/nvidia-mental-ray.html>

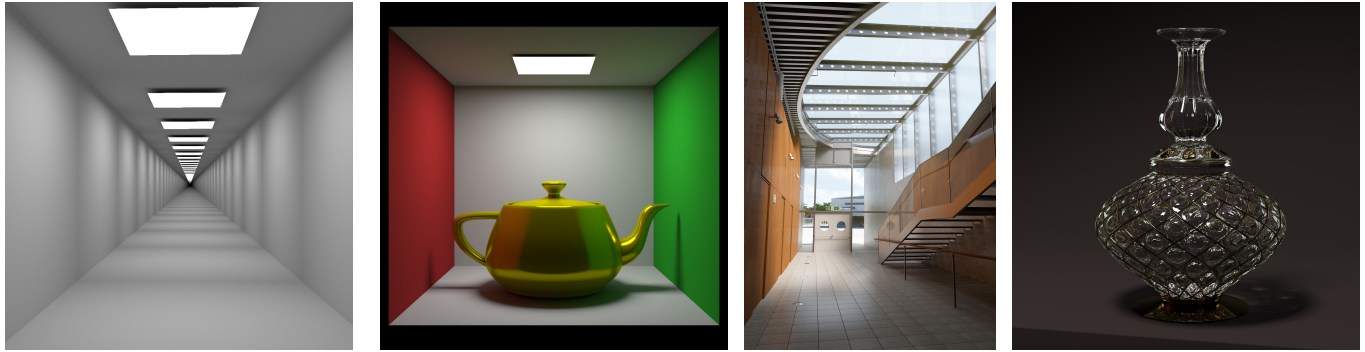


Figure A. Test 1, mirror corridor Figure B. Test 2, cornel box. Figure C. Test 3, arch. Figure D. Test 4, glass.

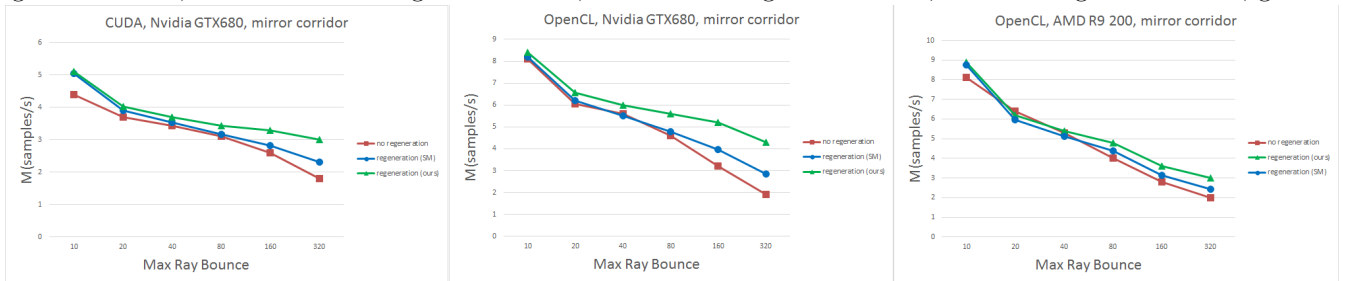


Figure 2. Test 1, mirror corridor paths per second (referred to fig. A).

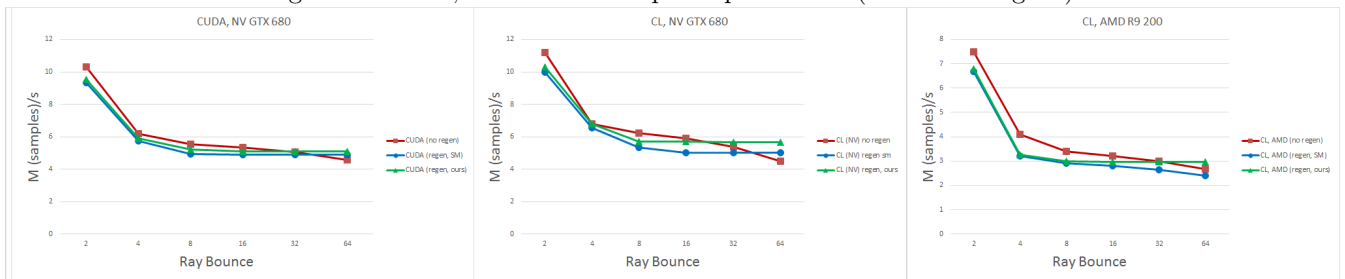


Figure 3. Test 2, cornel box paths per second (referred to fig. B).

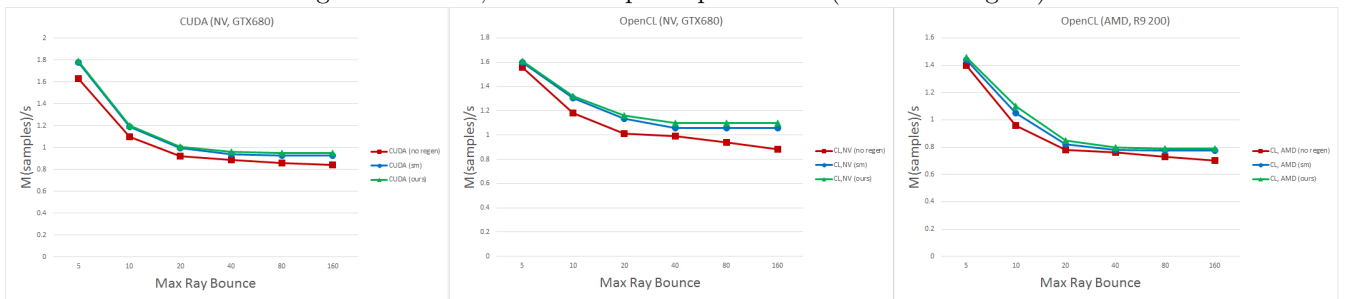


Figure 4. Test 3, arch-1; paths per second (referred to fig. C).

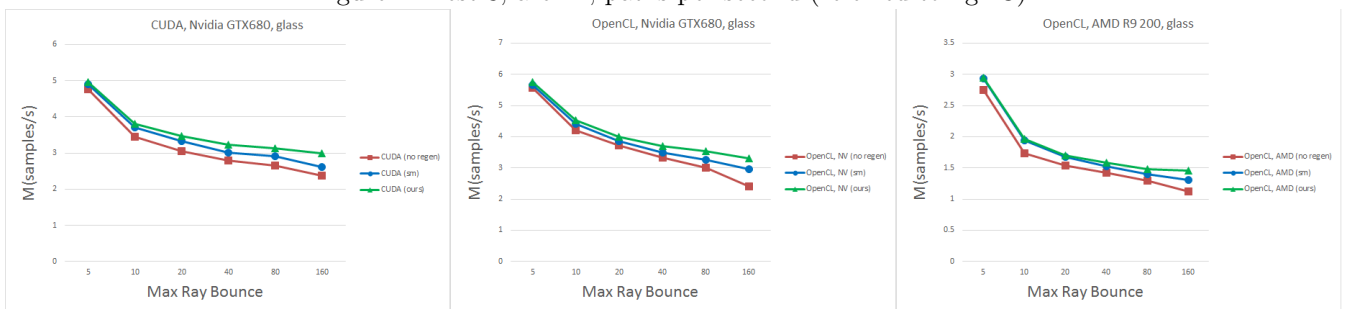


Figure 5. Test 4, glass; paths per second (referred to fig. D).

Построение физически корректной модели источника света по конечному набору лучей*

Д.Д. Жданов¹, А.А. Алейников¹, Е.В. Кудреватых¹, Н.Б. Дерябин²
ddzhdanov@mail.ru|aaaleinik@gmail.com|jonjawa91@gmail.com|dek@keldysh.ru

Санкт-Петербург, Россия, ¹Университет ИТМО;
Москва, Россия, ²Институт прикладной математики им. М.В. Келдыша РАН

Статья посвящена созданию физически корректной модели источника света, заданного в виде конечного набора лучей, полученного либо как результат обработки измерений световых характеристик источника света на установке типа Radiant Imaging, либо как результат прямой стохастической трассировки лучей в сцене, моделирующей излучение сложного источника света. Разработан алгоритм, позволяющий по конечному набору лучей построить непрерывную поверхность излучения источника света и его гониометрическую диаграмму. Модель позволяет сохранить спектральный состав исходных лучей в поверхностном распределении яркости излучения источника света.

Ключевые слова: Стохастическая трассировка лучей, источник света, Radiant Imaging model, пространственное распределение яркости, спектральное излучение.

The model of the complex light source constructed for the limited set of rays*

D.D. Zhdanov¹, A.A. Aleinikov¹, E.V. Kudrevatih¹, N.B. Deryabin²

¹ITMO University, St. Petersburg, Russian Federation;
²Keldysh Institute of Applied Mathematics, Moscow, Russian Federation

The article is devoted to the creation of a physically correct model of the light source which is specified as a finite set of rays obtained either as the result of processing of the measured characteristics of the light source on the installation type Radiant Imaging, or as a direct result of stochastic ray tracing in the scene, simulating the complex radiation of light source. An algorithm is developed that allows for a finite set of rays to construct a continuous surface emission light source and the goniometric diagram. The model allows preserving the spectral content of the source rays in the surface luminance distribution of the radiation light source.

Keywords: stochastic ray tracing, a light source, radiant Imaging model, the spatial luminance distribution, spectral radiation.

Введение

Фотореалистичная визуализация, как метод быстрого прототипирования, актуальна для различных областей применения [1]. Широкое распространение подход получил благодаря тому, что системы со сложной структурой, чрезвычайно трудно описать аналитически, а создание прототипа если и возможно, требует неприемлемо много временных и финансовых ресурсов.

Суть фотореалистичного синтеза изображений сводится к решению уравнения рендеринга [2]. Существуют различные способы численного решения этого уравнения, но идея их одна и заключается в трассировании лучей света или (и) лучей видимости, испускаемых наблюдателем.

Если в большинстве задач для обеспечения корректности результата достаточно использовать элементарную модель источника света, то при проек-

тировании сложных светооптических систем необходимо учитывать естественные особенности излучения. С этой целью практикуется измерение фотометрических характеристик источников света. Результат измерения представляется в виде ограниченного случайного набора лучей, который используется для компьютерного моделирования оптических устройств. Такой подход имеет ряд недостатков, осложняющих моделирование. Во-первых, распределение лучей носит случайный характер, что приводит к невозможности использования такой модели излучателя в методе детерминистической трассировки лучей, который в некоторых случаях оказывается более эффективным. В стохастической трассировке случайный набор лучей, определяющих источник света, согласуется с алгоритмической моделью метода. Однако возникает проблема появления артефактов, природа которых объясняется тем, что для сходимости результата моделирования может потребоваться количество лучей, превышающее общее число лучей, содержащихся в исходном наборе. Из-за этого, при трассировании один и тот же луч может использоваться множество раз. В результате на изображении образуются

Работа выполнена при частичной государственной финансовой поддержке ведущих университетов Российской Федерации (субсидия 074-U01), грантом РФФИ 13-01-00454, а также компанией Integra Inc. Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347

световые пятна, которые соответствуют отдельным лучам модели источника света. Кроме того, этот набор данных нельзя использовать для визуализации источника света.

Проблема ограниченности лучей, возникает также и в случае моделирования сложного осветителя, когда для обеспечения корректности результата моделирования необходимо трассировать десятки миллиардов трасс лучей. В качестве иллюстрации можно привести задачу проектирования освещения на основе световой трубы [3]. В такой сцене лишь малая часть генерируемых активным источником света лучей достигнет осветительной системы, а через нее попадет в освещаемое помещение. На этом примере видно, что природа ограниченности набора исходных данных обусловлена двумя причинами. Во-первых, низкой вероятностью попадания лучей в осветительную систему. А во-вторых, пропорционально сложности структуры возрастает и время расчета трасс лучей, и для обеспечения корректного результата общее время моделирования может быть неприемлемо большим.

Целью работы является разработка принципа построения физически корректной модели источника излучения по дискретному набору случайных лучей.

Исходные данные

В работе в качестве исходных данных использовались результаты измерения световых характеристик, произведенных на аппаратно-программном комплексе Radiant Vision Systems [4].

Упрощенная схема измерительной установки соответствует общей схеме гониофотометра, а процесс измерения характеристик не отличается от традиционного.

Формат, в котором содержится результат измерения, не имеет открытой спецификации, и использовать его можно только в программном обеспечении производителя, которое в качестве результата формирует ограниченный по количеству набор случайных лучей. Хотя формат полученного набора лучей также является проприетарным, но тем не менее имеет открытую спецификацию, в которой содержится общая информация об измеряемом источнике света (суммарный световой поток, используемая цветовая модель, количество лучей в наборе и др.) и список лучей. Каждый луч представляется в виде кортежа $\langle B, D, C \rangle$, где B – точка начала луча, D – направление луча и C – цвет луча. Рисунок 1 визуализирует исходный набор данных, используемых для построения модели.

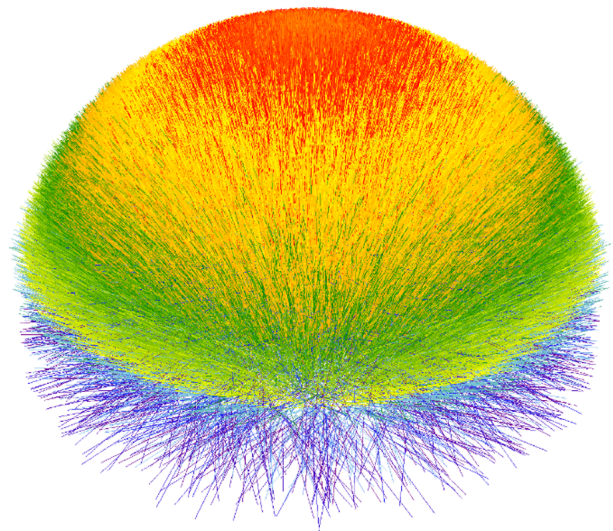


Рис. 1: Визуализация исходных данных. Цвет лучей соответствует полярному углу его направления

Построение геометрии оболочки источника света

Поскольку формат исходных данных в явном виде не содержит информации о геометрии источника света, то единственным материалом, по которому можно восстановить оболочку, является облако точек вершин лучей. Анализ облака точек позволяет формализовать следующие характеристики геометрии. Во-первых, излучатель может быть как поверхностным, так и объемным. Во-вторых, может быть как единым, так и составным объектом, т.е. сложный излучатель может состоять из ряда независимых излучателей (независимые излучатели могут быть как элементами поверхности, так и элементами объема) [5]. Кроме того, учитывая возможное дальнейшее использование источника света для расчета фотометрических характеристик и различных моделей фотореалистичного рендеринга, было выбрано представление геометрии излучателя в виде триангуляционной сети. Данное представление удобно для визуализации и подходит для большинства моделей рендеринга. Принимая во внимание то, что «треугольное» представление источника света является поверхностной моделью излучения, необходимо специальным образом реализовать возможность излучения из объема, ограниченного триангуляционной сетью. В простейшем случае это может быть выполнено переносом излучения из объема на границу источника света. В дальнейшем программные решения, использующие данную модель, могут осуществлять случайное излучение из объема, ограниченного триангуляционной сетью.

Поскольку, при поставленных выше условиях большинство известных методов триангуляции не могут

обеспечить корректного и эффективного построения геометрии, то для решения поставленной задачи был предложен оригинальный метод формирования триангуляционной сети по облаку точек. Главная идея метода заключается в проекции видовых плоскостей на облако точек. В данном методе излучатель анализируется по шести основным видам (так как это минимальное количество, по которым можно изучить трехмерное тело), совпадающим с гранями ограничивающей прямоугольной области.

Для того, чтобы избежать наложения триангуляционных сетей видовых проекций, облако точек дифференцируется на отдельные части, соответствующие видам. Далее, на проецирующей плоскости строится регулярная сеть заданной размерности, определяемой эмпирическим путем и соответствующей качеству создаваемой триангуляционной сети. Для каждой ячейки сетки находится уровень, который соответствует точке с минимальным расстоянием до проецирующей плоскости. Если для ячейки нет соответствующих точек, то ее уровень устанавливается как нулевой. На основании регулярной вершинной сети получаем полигональную сеть.

Представленный алгоритм является неполным, так как полученные триангуляционные сети являются несвязанными и, в большинстве случаев, отстоят друг от друга на значительное расстояние. Для удаления разрывов между сетями был разработан алгоритм их сшивания, основой которого является использование того факта, что проецирующие сети имеют регулярный характер и соприкасаются друг с другом.

В обоих предложенных методах, в силу алгоритмической простоты, могут возникать артефакты в виде недостроенных треугольников. Эта проблема устранима, если для каждой вершины определить, возможно ли однозначно достроить треугольник на основании связи с окружающими вершинами. Необходимо отметить, что при сшивании и построении отсутствующих областей триангуляционной сети, необходимо проверять размеры достраиваемых треугольников и сравнивать их с некоторой максимально допустимой мерой фильтрации, соответствующей тем случаям, когда отсутствие триангуляционной сети оправдано, поскольку излучение в данном месте отсутствует. Данный параметр, также получаемый эмпирическим путем, позволяет управлять качеством построения геометрии. Рисунок 2 визуализирует геометрию оболочки излучателя, полученную разработанным методом.

Таким образом, предложенный алгоритм удовлетворяет поставленным требованиям, прост в реализации и позволяет управлять качеством получаемой триангуляционной сети. Кроме того, количественная мера полученной триангуляционной сети

(разрешения по проекциям) не зависит от числа лучей, участвующих в ее построении, а задается параметром. Это позволит находить баланс между качеством и сложностью полигональной сети в каждой конкретной задаче.

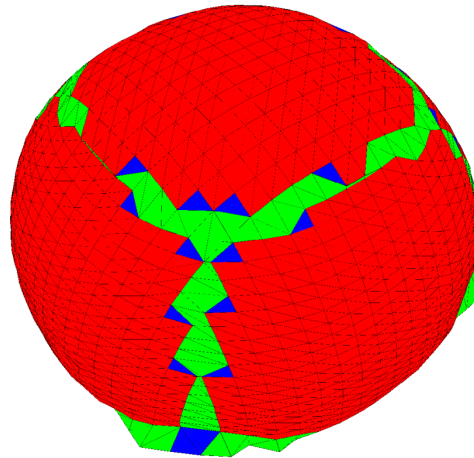


Рис. 2: Визуализация геометрии оболочки излучателя (красным цветом выделены области, построенные методом проекции, зеленым цветом – методом сшивания, синим цветом – методом построения недостающих треугольников)

Расчет поверхностного распределения светового потока

Полученная на предыдущем шаге триангуляционная сеть строилась приближенным методом. Следовательно, нет полного соответствия между полученной геометрией и исходным набором лучей. Поэтому необходимо спроецировать исходное облако лучей на полигональную сеть.

Проверка на пересечение с треугольником каждого луча с каждым полигоном имеет достаточно низкую эффективность. Однако, в подавляющем большинстве случаев мы имеем дело с поверхностными излучателями и точки начал лучей находятся в непосредственной близости от треугольников. Поэтому вероятность пересечения луча с ближайшими треугольниками достаточно высока и оптимальным решением будет использование пространственной структуры поиска, построенной по множеству полигонов [6]. После нахождения точки пересечения луча с треугольником, целесообразно перевести ее координаты в барицентрическую систему координат в базисе этого треугольника.

Для расчета непрерывного распределения энергии, излучаемой с поверхности источника света, заданного дискретной полигональной сетью, необходимо рассчитать распределение светового потока по полигонам и взвешенного светового потока по вершинам, образующим треугольники.

Световой поток, излучаемый каждым треугольником, может быть найден как:

$$F_t = \sum_{i=1}^{N_t} F_i, \quad (1)$$

где N_t - количество лучей, пересекших t -й треугольник, F_i - световой поток, переносимый i -м лучом. Рисунок 3 визуализирует поверхностное распределение светового потока, испускаемого треугольниками.

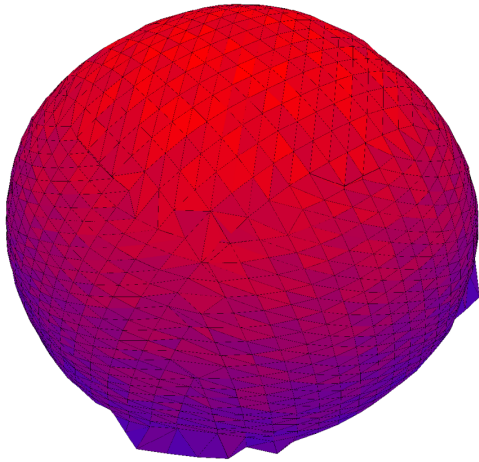


Рис. 3: Визуализация поверхностного распределения светового потока, испускаемого треугольниками

При расчете энергии излучения, испускаемой вершинами треугольника, может быть учтен его спектральный состав. В результате поверхностное распределение по вершинам триангуляционной сети будет иметь спектральное представление. Спектральная плотность распределения светового потока на вершинах полигональной сети может быть определена как:

$$\rho_l(\lambda) = \sum_{i=1}^{N_t} \sum_{j=1}^{N_l} \rho_i(\lambda) * p_{l,i,j}, \quad (2)$$

где N_l - количество треугольников, имеющих общую l -ю вершину, $\rho_i(\lambda)$ - спектральная плотность распределения светового потока i -го луча, $p_{l,i,j}$ - барцентрическая координата луча, соответствующая данной вершине треугольника. Рисунок 4 визуализирует поверхностное распределение плотности светового потока, испускаемого вершинами треугольников.

Для предотвращения появления артефактов, связанных с резкими перепадами плотности распределения световых потоков, используются алгоритмы фильтрации. Поскольку для полигональной сети было построено пространственное дерево поиска,

то одним из наиболее эффективных методов сглаживания будет алгоритм Шепарда [7].

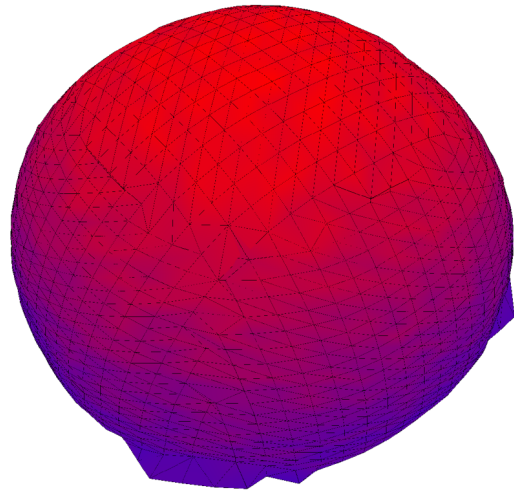


Рис. 4: Визуализация поверхностного распределения светового потока, испускаемого вершинами треугольников

Расчет углового распределения яркости

Для использования данной модели источника света в физически корректном компьютерном моделировании необходимо определить пространственно-угловую зависимость яркости источника излучения.

Постановка задачи допускает, что источник света может иметь неправильную сложную форму. Кроме того, угловое распределение яркости может варьироваться по поверхности излучения. Поэтому, единая угловая диаграмма распределения яркости (или интенсивности) излучения будет искажать (как правило, сглаживать) характер углового излучения, испускаемого разными зонами источника света. Такое сглаживание допустимо, когда освещаемый объект удален от источника света на значительное расстояние, однако при освещении объектов, находящихся в непосредственной близости от источника света, например, освещение торца светопроводящей пластины линейкой светодиодов, ошибка моделирования может быть значительной. Естественным решением данной задачи является расчет ряда гониометрических диаграмм для разных зон одного источника света. Оптимальным решением мог бы быть расчет отдельных диаграмм для каждой вершины построенной треугольной модели. Однако чрезмерное увеличение количества диаграмм влечет за собой сильную дифференциацию имеющегося набора исходных данных, что может привести к значительному зашумлению получаемого углового распределения при недоста-

точном количестве лучей. Как правило, исходный набор лучей содержит порядка миллиона или нескольких миллионов лучей, что является недостаточным для построения сотен и даже десятков гониометрических диаграмм. Поэтому разработанная модель ограничивает количество зон на поверхности источника света, имеющих различные гониометрические диаграммы. Зоны образуют регулярную прямоугольную решетку на одной из проекционных плоскостей (чья нормаль ориентирована по направлению максимальной интенсивности излучения) и параметры решетки определяются по плотности распределения лучей в данной плоскости (на данный момент общее количество зон ограничено 16).

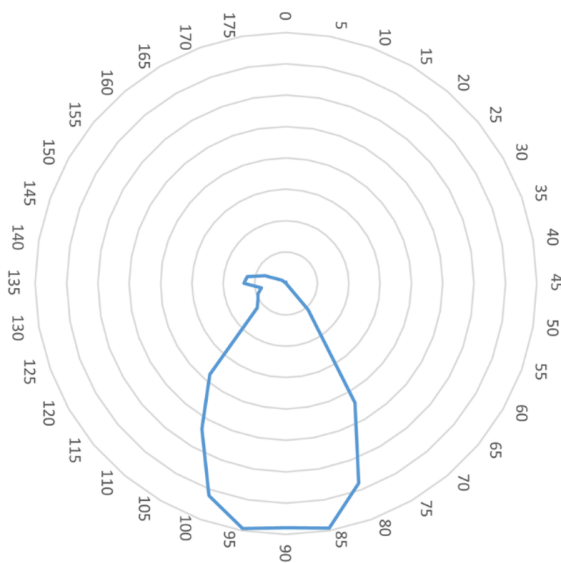


Рис. 5: График сечения распределения интенсивности излучения с одной зоны источника света

Гониометрическая диаграмма строится следующим образом. Для каждого луча, испускаемого с поверхности источника света, определяется зона, в которой осуществляется накопление гониометрической диаграммы. Далее, в системе координат гониометрической диаграммы определяются полярные координаты луча и световой поток, переносимый лучом, добавляется в соответствующую ячейку диаграммы, имеющей вид регулярной решетки по полярным координатам. В результате данной процедуры накапливается распределение интенсивности излучения по зонам источника света. При наличии спектральной составляющей потока излучения в испускаемых лучах накапливается спектральное распределение интенсивности излучения (аналогично плотности спектрального потока излучения в вершинах треугольников). Рисунок 5 представляет пример графика сечения диаграммы интенсивности излучения.

Как правило, вследствие недостаточности исходных данных полученные диаграммы интенсивности имеют значительный уровень шума и должны быть сглажены. Для фильтрации может применяться любой метод, работающий на регулярной сети.

Заключение

В рамках данной работы был реализован алгоритм построения физически-корректной модели источника излучения по дискретному ограниченному набору лучей. Разработанная модель, в силу непрерывности распределения фотометрических характеристик, может быть использована как в детерминистических, так и в стохастических методах трассировки лучей. Кроме того, данный подход позволяет заменить источник света, заданный сложным оптическим устройством, на упрощенную, физически-корректную модель излучателя, что повышает эффективность рендеринга.

Приведенные примеры иллюстрируют возможность построения непрерывной поверхности и гониометрической диаграммы излучения источника света для ограниченного набора лучей.

Литература

- [1] А.Г. Волобой, В.А. Галактионов. Компьютерная графика как эффективный инструмент развития современных технологий // Труды 23-й Международной Конференции по Компьютерной Графике и Зрению, Институт автоматки и процессов управления ДВО РАН, Дальневосточный Федеральный Университет, 16-20 сентября 2013 года, С.186-190.
- [2] Kajiya, J.T. The rendering equation / Siggraph 1986: 143.
- [3] Laouadi A., Saber H.H. Performance of Tubular Daylighting Devices // NRC CONSTRUCTION TECHNOLOGY UPDATE, National Research Council of Canada, No. 82, June 2014.
- [4] Radiant Vision Systems SIG-400, URL: <http://www.radiantvisionsystems.com/products/sig-400>
- [5] Жданов Д.Д., Гарбуль А.А., Волобой А.Г., Галактионов В.А., Ершов С.В., Потемин И.С., Соколов В.Г. Фотореалистичная модель объемного рассеивания в задаче двунаправленной стохастической трассировки лучей // Труды 24-й Международной Конференции по Компьютерной Графике и Машинному Зрению ГрафиКон'2014, Ростов-на-Дону, 30 сентября – 3 октября, С.38-42.
- [6] Wald I., Havran V. On building fast kd-trees for ray tracing, and on doing that in $O(N \log N)$ // RT06 conference, Salt Lake City, UTAH, USA, September 2006, pp.61-69.
- [7] Shepard D. A two-dimensional interpolation function for irregularly-spaced data // 23rd ACM national conference (ACM'68), ACM, New York, NY, USA, 1968, pp.517-524.

Модель сложного источника излучения в методе двунаправленной стохастической трассировки лучей*

Д.Д. Жданов¹, А.А. Алейников¹, Е.В. Кудреватых¹, В.И. Валиев²

ddzhdanov@mail.ru|aaaleinik@gmail.com|jonjawa91@gmail.com|valiev@gin.keldysh.ru

Санкт-Петербург, Россия, ¹Университет ИТМО;

Москва, Россия, ²Институт прикладной математики им. М.В. Келдыша РАН

Данная статья посвящена разработке физически корректной модели источника света, имеющего сложный характер пространственного, углового и спектрального излучения света, и интеграции данной модели в методы стохастической трассировки лучей. Программные интерфейсы разработанной модели позволяют использовать ее в методах прямой, обратной и двунаправленной стохастической трассировки лучей. Представлены алгоритмы данных моделей, обеспечивающие расчет яркости прямого видения источника света и яркостей первичного и вторичного освещений наблюдаемых объектов сцены, выполненные для метода двунаправленной стохастической трассировки лучей. Приведенные примеры иллюстрируют возможность применения разработанных моделей для фотореалистичной визуализации сцен с источниками света, изначально заданными в виде конечного набора лучей.

Ключевые слова: стохастическая трассировка лучей, источник света, прямая видимость источника света, первичная яркость, вторичная яркость, фотореалистичная визуализация.

The model of the complex light source constructed for the limited set of rays*

D.D. Zhdanov¹, A.A. Aleinikov¹, E.V. Kudrevatikh¹, V.I. Valiev²

¹ITMO University, St. Petersburg, Russian Federation;

²Keldysh Institute of Applied Mathematics, Moscow, Russian Federation

The article is devoted to the creation of a physically correct model of the light source with the complex character of spatial, angular and spectral distribution and integration of the model to method of bi-directional stochastic ray tracing. The developed program interfaces allow integration of the models to any kind of stochastic ray tracing simulator. The designed algorithms of the light emitter allows physically accurate calculation of direct vision luminance component and components of direct and indirect light illumination for the method of the bi-directional stochastic ray tracing. Examples of photorealistic images of scenes illuminated with light sources represented as a set of rays illustrate the article.

Keywords: stochastic ray tracing, a light source, direct light source visibility, primary luminance, secondary luminance, photorealistic visualization.

Введение

В настоящее время, задача физически корректного моделирования и проектирования сложных оптических устройств и их фотореалистичной визуализации в условиях реального окружения приобретает все большую актуальность. При проектировании современных осветителей используются новая элементная база, новые конструкторские и технологические решения, поэтому методы детерминистической трассировки луча становятся неприменимыми для корректного моделирования и проектирования систем данного класса. На смену детерминистическим методам приходят методы стохастического моделирования и большинство основных коммерческих систем компьютерного моделирования [1, 2] имеют в своем составе данные модели.

Работа выполнена при частичной государственной финансовой поддержке ведущих университетов Российской Федерации (субсидия 074-U01), грантом РФФИ 13-01-00454, а также компанией Integra Inc. Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347

Стохастические методы моделирования обеспечивают физически корректное решение уравнение рендеринга [3]. Наиболее оптимальным методом решения уравнения рендеринга является метод двунаправленной стохастической трассировки лучей, который объединяет методы прямой и обратной трассировки лучей, что позволяет оптимальным образом интегрировать вторичную яркость на диффузных поверхностях между источником и приемником излучения. Однако, для расчета яркости прямой видимости, первичной и вторичной яркостей метод двунаправленной стохастической трассировки лучей требует реализации определенных программных интерфейсов со стороны элементов сцены.

Одним из элементов сцены, обеспечивающих программные интерфейсы расчета яркости, является источник света. Как правило системы компьютерного моделирования используют модели непрерывных источников излучения, имеющих определенную форму поверхности, в которых значение

яркости излучения может быть вычислено для любой точки поверхности. Однако, современные источники излучения имеют сложный характер пространственного и углового распределения излучения и вычисление и построение корректной модели излучения является дорогостоящей задачей. Поэтому ряд компаний, таких как OSRAM и Radiant Imaging Zemax, осуществляют измерения световых характеристик источников света и простейших осветительных устройств на аппаратно-программных комплексах типа SIG-400 Radiant Vision Systems [4]. После обработки результат измерений может быть сохранен в виде модели, представляющей собой набор случайных лучей (точка начала, направление и цвет луча), испускаемых данным источником света. Если количество лучей в данной модели достаточно велико, то она может корректно описать процесс излучения света в методе прямой стохастической трассировки лучей. Для моделирования процесса излучения достаточно случайным образом выбрать луч в соответствии с плотностью распределения энергий лучей в заданном наборе. Однако для метода двунаправленной стохастической трассировки лучей данное решение непригодно, поскольку не позволяет вычислить яркость прямого виденья источника света, а компонента первичной яркости содержит постоянную составляющую шума, вызванную дискретным характером излучения источника света. Рисунок 1 иллюстрирует недостатки модели источника света, разработанные для метода прямой стохастической трассировки луча и примененные в методе двунаправленной стохастической трассировки лучей.

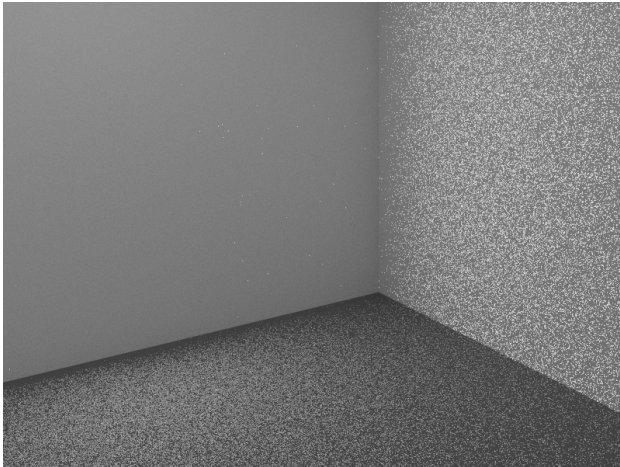


Рис. 1: Недостатки модели источника света, разработанной для метода прямой стохастической трассировки лучей

Сцена представляет собой замкнутый параллелепипед, освещенный светодиаодным кластером, со-

стоящим из 39 светодиаодов, показанных на рисунке 2, и расположенным в области виденья на левой незашумленной стенке. Очевидно, что светодиаод не виден, а изображение содержит неустраиваемый шум.

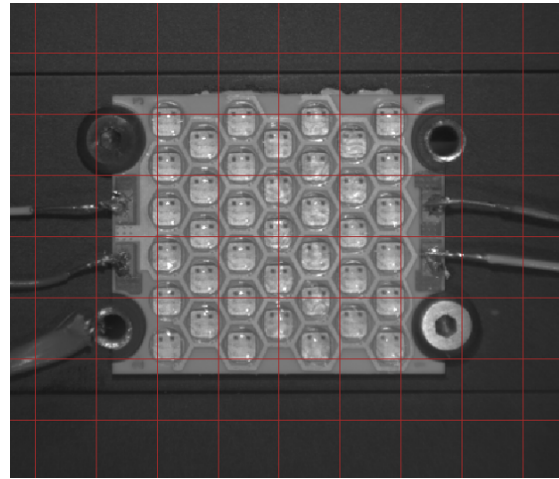


Рис. 2: Фотография светодиаодного кластера

Целью данной работы являлась полномасштабная реализация сложной дискретной модели источника света, позволяющая реализовать методы расчета яркости прямого виденья, первичной и вторичной яркостей, в методе стохастической двунаправленной трассировки лучей. Кроме того, данная модель должна исключить постоянную составляющую шума изображения, вызванную дискретным характером излучения источника света.

Расчет яркости методом двунаправленной трассировки лучей

Для статических сцен видимая яркость световой компоненты c в точке \vec{p} и по направлению \vec{v} определяется следующим уравнением рендеринга:

$$L(\vec{p}, \vec{v}, c) = \tau(\vec{p}, \vec{v}, c) \left(\frac{1}{\pi} \int_{4\pi} BSDF(\vec{p}, \vec{v}, \vec{v}', c) L(\vec{p}, \vec{v}', c) (\vec{n} \cdot \vec{v}') d\omega \right) + L_0(\vec{p}, \vec{v}, c) \quad (1)$$

где: $L_0(\vec{p}, \vec{v}, c)$ – собственная яркость объекта в точке наблюдения,

\vec{n} – нормаль к поверхности в точке наблюдения,

$\tau(\vec{p}, \vec{v}, c)$ – пропускание (прозрачность) среды между наблюдателем и точкой наблюдения,

$BSDF(\vec{p}, \vec{v}, \vec{v}', c)$ – коэффициент яркости поверхности (или двунаправленная функция рассеивания (ДФР)) от источника освещения \vec{v}' в направлении \vec{v} на наблюдателя,

$L(\vec{p}, \vec{v}, c)$ – яркость внешнего (первичного или вторичного) освещения в телесном угле $d\omega$ по направлению \vec{v}' на точку наблюдения \vec{p} .

Поскольку уравнение (1) это уравнение с бесконечной рекурсией, то при наличии значительной доли

первичного и вторичного диффузного освещения, наиболее подходящим способом решения данного уравнения является метод двунаправленной стохастической трассировки лучей. Для прерывания бесконечной рекурсии, возникающей при выполнении интегрирования, наиболее подходящим решением является метод русской рулетки. Рисунок 3 схематично иллюстрирует данный алгоритм.

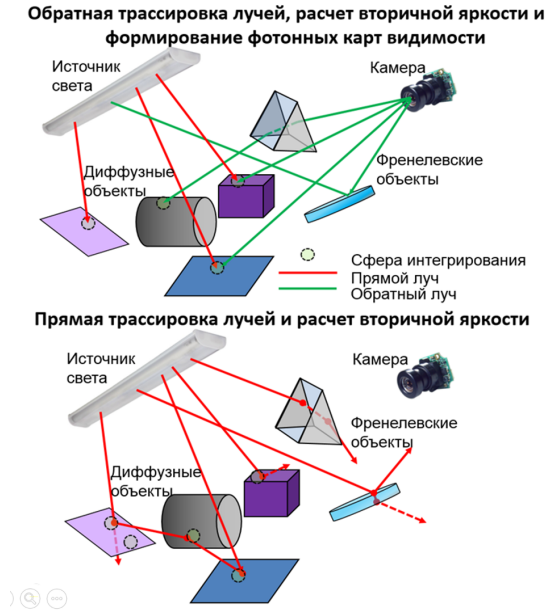


Рис. 3: Метод двунаправленной стохастической трассировки лучей

Двунаправленная трассировка лучей состоит из двух чередующихся процессов: (1) обратной трассировки луча, позволяющей на «зеркальных» трассах вычислить видимую яркость источников света, сформировать карты виденья сцены и для всех видимых диффузных объектов вычислить яркость прямого освещения, и (2) прямой трассировки луча, позволяющей вычислить яркость вторичного освещения в диффузных точках карты виденья сцены. Выражение для вычисления яркости прямого виденья достаточно просто и имеет вид:

$$L_B^V(\vec{p}, \vec{v}, c) = \frac{\sum_{i=1}^{N_B} \tau(\vec{p}_i, \vec{v}_i, c) \cdot L_i(\vec{p}_i, \vec{v}'_i, c)}{N_B} \quad (2)$$

где: N_B – количество лучей, испущенных наблюдателем;

$L_i(\vec{p}_i, \vec{v}'_i, c)$ – яркость источника света, находящегося на i -ой трассе «зеркального» луча.

В данном случае яркость прямого виденья – это средняя яркость всех источников света, находящихся на трассе «зеркальных» лучей, испущенных наблюдателем из одной точки камеры.

Первичная яркость – это яркость непосредственно освещаемых диффузных объектов, которые на-

блюдаются либо непосредственно, либо посредством отражения (пропускания) луча наблюдения от «зеркальных» объектов сцены. Для вычисления первичной яркости используются методы «light sampling» (3) $L_L^D(\vec{p}, \vec{v}, c)$ и «BDF sampling» (4) $L_B^D(\vec{p}, \vec{v}, c)$ [5]:

$$L_L^D(\vec{p}, \vec{v}, c) = \frac{\sum_{i=1}^{N_B} \sum_{m=1}^{M_i} \tau_m(\vec{p}_{im}, \vec{v}_{im}, c) \sum_{j=1}^{N_S} BPDF_{im}(\vec{p}_{im}, \vec{v}_{im}, \vec{v}'_{imj}, c) E_j(\vec{p}_{im}, \vec{v}'_{imj}, c)}{\pi \cdot N_B \cdot N_S} \quad (3)$$

$$L_B^D(\vec{p}, \vec{v}, c) = \frac{\sum_{i=1}^{N_B} \sum_{m=1}^{M_i} \tau_m(\vec{p}_{im}, \vec{v}_{im}, c) \sum_{j=1}^{N_S} L_j(\vec{p}_{im}, \vec{v}'_{imj}, c)}{N_B \cdot N_S} \quad (4)$$

где: M_i – количество сохраненных фотонов на поверхностях сцены для i -ой трассы луча;

N_S – количество случайных попыток вычисления прямой освещенности $E_j(\vec{p}_{im}, \vec{v}'_{imj}, c)$ для метода «light sampling» и яркости источников света $L_j(\vec{p}_{im}, \vec{v}'_{imj}, c)$, находящихся на трассе луча после рассеивания луча на j -ой поверхности для метода «BDF sampling».

В случае метода «light sampling» яркость – это средняя яркость, сформированная случайными точками на источнике излучения, освещающими диффузную точку наблюдения. Метод «BDF sampling» аналогичен методу расчета яркости прямого виденья, за исключением того, что яркость вычисляется, и соответствующим образом усредняется, после N_S попыток однократного диффузного рассеивания на поверхности, наблюдаемой на трассе «зеркального» луча. Вторичная яркость – это яркость, накопленная после двух и более рассеиваний на трассе луча от точки наблюдения до источника света. Данная яркость вычисляется в диффузной области пересечения трасс прямых и обратных лучей. Область пересечения ограничена сферой интегрирования, формируемой в процессе трассировки обратных лучей в виде карты виденья сцены. Выражение яркости имеет следующий вид:

$$L'(\vec{p}, \vec{v}, c) = \frac{\sum_{i=1}^{N_B} \sum_{m=1}^{M_i} \tau_m(\vec{p}_{im}, \vec{v}_{im}, c) \sum_{j=1}^{N_S} BPDF_{im}(\vec{p}_{im}, \vec{v}_{im}, \vec{v}'_{imj}, c) F_{Rj}(\vec{p}_{im}, \vec{v}'_{imj}, c)}{\pi^2 N_B N_F r_{im}^2} \quad (5)$$

где: N_F – количество лучей, испущенных источниками света;

N_R – количество фотонов, попавших в m -ую сферу интегрирования (фотонную карту) на трассе i -го луча;

r_{im} – радиусы сфер интегрирования вторичной яркости для i -го луча на m -ой поверхности наблюдения соответственно;

$F_{Rj}(\vec{p}_{im}, \vec{v}'_{imj}, c)$ – спектральный поток, переносимый j -ым фотоном в точке \vec{p}_{im} по направлению \vec{v}'_{imj} и попавший в сферу интегрирования фотонной карты видимости.

То есть вторичная яркость – это средняя яркость вторичного освещения, попавшего в область сфер интегрирования карт виденья сцены.

Ниже рассматриваются программные модели источников света, обеспечивающие вычисление всех трех компонент яркости.

Расчет яркости прямого виденья источника света

В соответствии с уравнением (2) яркость прямого виденья, как и первичная яркость в методе “BDF sampling” (4), вычисляется в точке пересечения обратного луча с поверхностью источника света. То есть, с точки зрения программной реализации модели источника света, необходимо выполнить две операции, во-первых, найти точку пересечения луча с поверхностью источника света \vec{p}_i , а во-вторых, вычислить яркость источника света в точке \vec{p}_i по направлению \vec{v} .

Принимая во внимание специфику модели источника света, представляющей собой цветное облако лучей, данные операции не могут быть непосредственным образом применены к существующей модели.

Поиск пересечения луча с поверхностью источника света явным образом невозможен. Однако, если заменить точки начала лучей на сферы, то можно найти точку пересечения луча с ближайшей сферой и принять ее за граничную точку тела источника света. Проблемы данного метода заключаются в слишком большом количестве точек (количество точек может достигать десятков миллионов) и неопределенном значении радиуса сферы в начале луча.

Для решения данных проблем в области определения начал лучей строится плотность их распределения. Плотность распределения строится на пространственной решетке, разрешение которой определяется количеством лучей в источнике света (как правило начальное разрешение $128 \times 128 \times 128$), и по данной плотности вычисляется оптимальное количество центров сфер и радиусов в каждой ячейке решетки. Как правило, количество точек на одну ячейку ограничивается 16, точки выбираются максимально равноудаленными друг от друга, а их радиусы должны полностью заполнять ячейку. Кроме того, сферы формируются только по границе области, что позволяет существенно сократить объем исходных данных, формирующих поверхность источника света.

Расчет поверхностной яркости источника света осуществляется по формуле:

$$L_i(\vec{p}_i, \vec{v}'_i, c) = \frac{\Delta^2 F(\vec{p}_i, \vec{v}'_i, c)}{\Delta S(\vec{p}_i, c) \cdot \Delta \Omega(\vec{v}'_i, c) \cdot (\vec{n} \cdot \vec{v}'_i)} \quad (6)$$

где: $\Delta^2 F(\vec{p}_i, \vec{v}'_i, c)$ – поток, испускаемый ячейкой источника света $\Delta S(\vec{p}_i, c)$ в телесном угле $\Delta \Omega(\vec{v}'_i, c)$ и по направлению \vec{v}'_i к нормали излучающей области \vec{n} (ячейки решетки).

Поскольку для аккуратного вычисления распределения светового потока разрешение решетки, используемой для формирования области виденья, является чрезмерным (из-за ограниченного количества лучей в исходных данных), то для вычисления яркости (6) используются дополнительные решетки более низкого разрешения. Для построения пространственного распределения потока начальное распределение решетки ($32 \times 32 \times 32$), а для пространственного распределения интенсивности излучения – ($5 \times 5 \times 5$). Распределения строятся по трем ортогональным осям решетки, что позволяет оптимальным образом выбрать $\vec{n} \cdot \vec{v}'_i$.

Рисунок 4 показывает яркость прямого виденья источника света, представленного на рисунке 2.



Рис. 4: Изображение сцены, сформированное яркостью прямого виденья источника света

Расчет яркости первичного и вторичного освещения

В соответствии с уравнениями (3) и (5) яркости прямого и вторичного освещения вычисляются в два этапа. На первом этапе в соответствии с распределением светового потока по ячейкам источника излучения выбирается ячейка излучения, а затем, используя метод “rejection sampling”, выбирается точка излучения внутри ячейки. Выполнение второго этапа для расчета прямой (3) и вторичной (5) яркостей идет по разным схемам. В первом случае для выбранной точки излучения вычисляется интенсивность светового излучения в точке наблюдения. А во втором случае осуществляется случайный выбор направления луча в соответствии с гониометрическими диаграммами, хранящимися в ячейке, содержащей точку излучения.

Поскольку решетки, на которых определяются пространственные распределения потока и интенсивности излучения, не совпадают друг с другом, то цвет источника света для полученного направления может быть выбран опционально по одному из распределений.

Рисунки 5 и 6 показывают яркости первичного и вторичного освещения сцены источником света, представленным на рисунке 2.

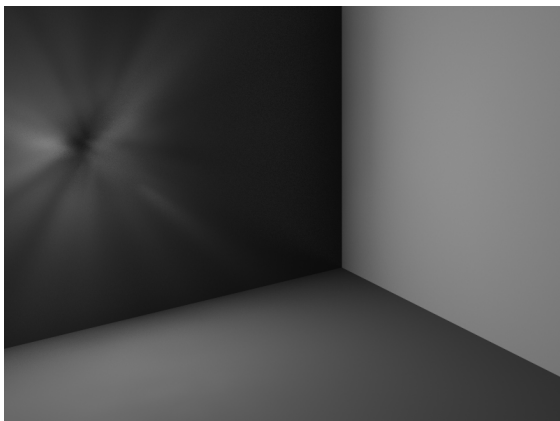


Рис. 5: Изображение сцены, сформированное яркостью первичного освещения

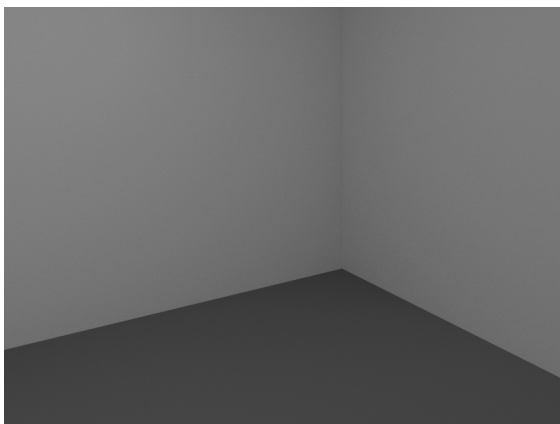


Рис. 6: Изображение сцены, сформированное яркостью вторичного освещения

Заключение

В рамках программного комплекса Lumicept [6] была реализована модель сложного дискретного источника света, обеспечивающая физически корректное решение уравнения рендеринга для метода двунаправленной стохастической трассировки лучей. Разработанные программные решения обеспечивают возможность построения фотореалистичного изображения, включая изображение самого источника света, без постоянной шумовой составляющей яркости первичного освещения. Рисунок 7 иллюстрирует, как возможности разработанных моделей позволяют повысить качество и реализм изображения сцены по сравнению с начальной моделью, представленной на рисунке 1.

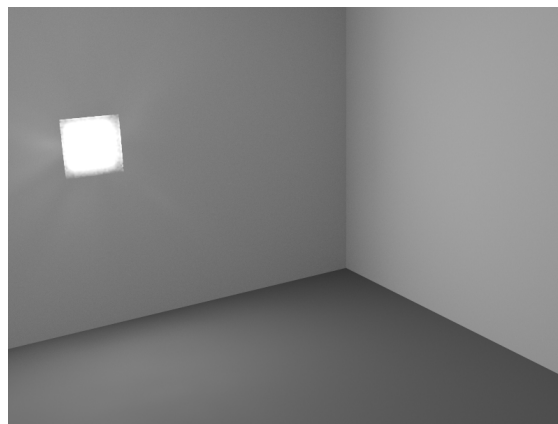


Рис. 7: Полное изображение сцены, построенное с помощью разработанной модели источника света

Литература

- [1] Synopsys, LightTools Illumination Design Software, URL: <http://optics.synopsys.com/lighttools/>, 2015.
- [2] Synopsys, LightTools Illumination Design Software, URL: <http://optics.synopsys.com/lighttools/>, 2015.
- [3] Kajiya, J.T. The rendering equation / Siggraph 1986: 143.
- [4] Radiant Vision Systems SIG-400, URL: <http://www.radiantvisionsystems.com/products/sig-400>, 2015.
- [5] Eric Veach. A dissertation: Robust Monte-Carlo methods for light transport simulation, URL: http://graphics.stanford.edu/papers/veach_thesis/thesis.pdf, 1997.
- [6] Intgera Inc., Lumicept, URL: <http://www.integra.jp/en>, 2015.

Адаптивный выбор глубины трассировки обратного луча в методе двунаправленной стохастической трассировки лучей*

Д.Д. Жданов¹, С.В. Ершов², А.Г. Волобой²

ddzhdanov@mail.ru | measure@spp.keldysh.ru | voloboy@gin.keldysh.ru

Санкт-Петербург, Россия, ¹Университет ИТМО;

Москва, Россия, ²Институт прикладной математики им. М.В. Келдыша РАН

Данная статья посвящена задаче ускорения сходимости физически корректной модели рендеринга сцен, содержащих объекты с «узконаправленными» двунаправленными функциями рассеивания (ДФР). В качестве базового подхода для решения уравнения рендеринга использовался метод двунаправленной стохастической трассировки лучей, основанный на фотонных картах «видимости» объектов сцены. Было показано, что вычисление вторичной яркости на поверхностях с узконаправленными ДФР, приводит к значительному шуму изображения, который остается видимым даже после трассировки миллиардов лучей. Для уменьшения шума на данных поверхностях были разработаны два решения, позволяющих сдвинуть точку интегрирования вторичной яркости на поверхность с более «гладким» ДФР. Приведенные примеры иллюстрируют возможность уменьшения шума изображения сцен, содержащих «узконаправленные» ДФР.

Ключевые слова: *Двунаправленная стохастическая трассировка лучей, фотореалистичная визуализация, двунаправленная функция рассеивания, фотонные карты, вторичная яркость, первичная яркость, диффузная глубина трассировки лучей.*

Adaptive estimation of backward ray trace depth for the stochastic bi-directional ray trace method*

D.D. Zhdanov¹, S.V. Ershov², A.G. Voloboy²

¹ITMO University, St. Petersburg, Russian Federation;

²Keldysh Institute of Applied Mathematics, Moscow, Russian Federation

The article is devoted the problem of acceleration of convergence of physically correct rendering model of scenes containing surfaces with narrow BSDF. As a base approach to solve the rendering equation the method of bidirectional stochastic ray tracing was applied. The method used stochastic backward ray tracing to calculate visibility maps as sources of indirect luminance integrations in these points. The article shows that secondary luminance calculations on the surfaces with narrow BSDF results in high noise, which remains visible, even after tracing of milliards of rays. To solve the problem of high noise two methods were developed. The methods allows moving the point of integration of secondary luminance to surfaces with more diffuse BSDF and therefore reduce noise of the secondary luminance. The examples illustrate how it is possible to reduce noise for scenes with narrow BSDF.

Keywords: *Bidirectional stochastic ray tracing, photorealistic visualization, BSDF, photon maps, direct luminance, indirect luminance, ray tracing depth.*

Введение

Задача физически корректного моделирования и фотореалистичной визуализации оптически сложных сцен имеет большое значение для моделирования и проектирования оптически сложных материалов, используемых в условиях реального окружения. Изображения объектов, имеющих специальным образом обработанные поверхности или покрытия и используемых, например, для придания особого визуального эффекта или специального характера распространения света, должны рассчитываться исходя не только из условий прямого освещения, но и условий вторичного освещения,

создаваемого светом, переотраженным объектами сцены. Как правило, данные свойства поверхности описываются спектральной двунаправленной функцией рассеивания (ДФР), которая далека от равнояркой ДФР. Физически корректный учет всех физических эффектов, включая рассеивание света на поверхностях с ДФР, заложен в решение уравнения рендеринга [1], которое представляет собой бесконечную сумму яркостей, и единственным физически корректным способом решения данного уравнения является метод «русской рулетки». Метод «русской рулетки» может быть использован для различных технологий трассировок лучей (прямой, когда изображение строится лучами, испущенными источниками света, обратной, когда изображение строится лучами, испущенными наблюдателем, и двунаправленной, когда яркость изображения вычисляется на пересечении трасс прямых и обратных лучей), однако для фотореалистичной визуализации

Работа выполнена при частичной государственной финансовой поддержке ведущих университетов Российской Федерации (субсидия 074-U01), грантом РФФИ 13-01-00454, а также компанией Integra Inc. Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347

лизации наиболее подходящим является метод двунаправленной стохастической трассировки лучей. Метод двунаправленной стохастической трассировки лучей имеет большое количество алгоритмических модификаций [2-4], и основные вариации алгоритмов сводятся к порядку расчета трасс лучей, типам фотонных карт, которые сохраняются в результате расчета, и способу хранения и обработки фотонных карт. Алгоритмы также варьируются по возможности учета физических явлений, таких как спектральные и поляризационные характеристики источников света и оптические свойства объектов сцены.

Как и любой метод стохастического моделирования, метод двунаправленной стохастической трассировки лучей формирует зашумленное изображение и, хотя в пределе данный метод должен сформировать изображение, свободное от шума, время для получения данного изображения может оказаться слишком большим. Поэтому на практике используются различные алгоритмы, позволяющие ускорить процесс сходимости вычисления яркости изображения и минимизации шума.

В рамках данной работы представляется один из методов ускорения сходимости процесса вычисления вторичной яркости в сценах, содержащих поверхности с узконаправленным ДФР.

Источники шума на поверхностях с узконаправленным ДФР

Фотореалистичная визуализация оптически сложных сцен сводится к решению уравнения рендеринга для каждой точки изображения сцены. Для статических сцен видимость яркости световой компоненты s в точке \vec{p} и по направлению \vec{v} определяется следующим образом:

$$L(\vec{p}, \vec{v}, c) = \tau(\vec{p}, \vec{v}, c) \left(\frac{1}{\pi} \int_{4\pi} BSDF(\vec{p}, \vec{v}, \vec{v}', c) L(\vec{p}, \vec{v}', c) (\vec{n} \cdot \vec{v}') d\omega \right) + L_0(\vec{p}, \vec{v}, c) \quad (1)$$

где: $L_0(\vec{p}, \vec{v}, c)$ – собственная яркость объекта в точке наблюдения,

\vec{n} – нормаль к поверхности в точке наблюдения,

$\tau(\vec{p}, \vec{v}, c)$ – пропускание (прозрачность) среды между наблюдателем и точкой наблюдения,

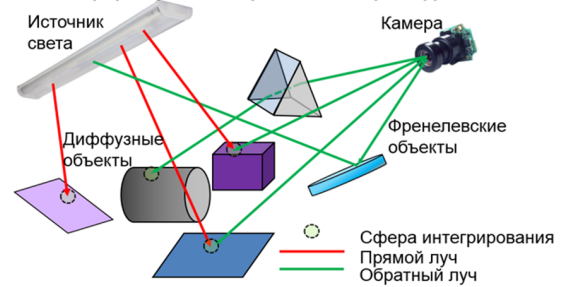
$BSDF(\vec{p}, \vec{v}, \vec{v}', c)$ – коэффициент яркости поверхности (или двунаправленная функция рассеивания (ДФР)) от источника освещения \vec{v}' в направлении \vec{v} на наблюдателя,

$L(\vec{p}, \vec{v}, c)$ – яркость внешнего (первичного или вторичного) освещения в телесном угле $d\omega$ по направлению \vec{v}' на точку наблюдения \vec{p} .

Уравнение рендеринга (1) это уравнение с бесконечной рекурсией, и, если вклад вторичного излучения в суммарное значение видимой яркости значителен, то наиболее подходящим способом решения уравнения рендеринга является двунаправленная стохастическая трассировка лучей. Рисунок

1 схематично иллюстрирует разработанный алгоритм двунаправленной трассировки лучей.

Обратная трассировка лучей, расчет вторичной яркости и формирование фотонных карт видимости



Прямая трассировка лучей и расчет вторичной яркости

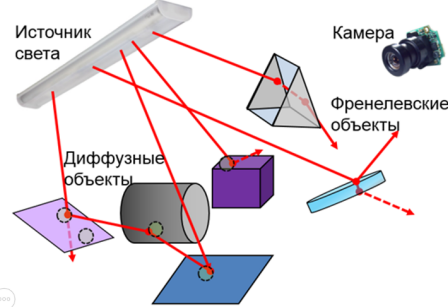


Рис. 1: Двунаправленная стохастическая трассировка лучей

Использование метода «русской рулетки» позволяет наиболее эффективно и физически корректно выполнить интегрирование бесконечной рекурсии. Для решения уравнения рендеринга авторами был разработан алгоритм двунаправленной стохастической трассировки лучей, в котором на первой стадии трассируются обратные лучи, вычисляется прямая яркость и сохраняются фотонные карты видимости сцены. На второй стадии трассируются прямые лучи, и в местах пересечения прямых лучей со сферами фотонных карт видимости сцены вычисляются вторичные и каустические яркости. Алгоритмы расчета каустического и вторичного освещения отличаются только параметром радиуса сферы интегрирования (для расчета каустического освещения используется меньший радиус, поскольку данное освещение создает больший контраст изображения).

Для алгоритма двунаправленной стохастической трассировки лучей, основанного на методе «русской рулетки», уравнение рендеринга (1) удобно представить в виде суммы трех независимых яркостей: во-первых, яркости прямого видения источников света, во-вторых, яркости, формируемой прямым освещением и, в-третьих, яркости, формируемой каустическим и диффузным освещением, – $L^V(\vec{p}, \vec{v}, c)$, $L^D(\vec{p}, \vec{v}, c)$, $L^I(\vec{p}, \vec{v}, c)$ соответственно. Как правило, основной шум изображения формируется при вычислении первичной и вторич-

ной яркостей. Для подавления шума первичная яркость вычисляется как взвешенная сумма яркостей, рассчитанных методами “light sampling” (2) $L_L^D(\vec{p}, \vec{v}, c)$ и “BDF sampling” (3) $L_B^D(\vec{p}, \vec{v}, c)$ [5]:

$$L_L^D(\vec{p}, \vec{v}, c) = \frac{\sum_{i=1}^{N_B} \sum_{m=1}^{M_i} \tau_{im}(\vec{p}_{im}, \vec{v}_{im}, c) \sum_{j=1}^{N_S} B S D F_{im}(\vec{p}_{im}, \vec{v}_{im}, \vec{v}'_{imj}, c) E_j(\vec{p}_{im}, \vec{v}'_{imj}, c)}{\pi \cdot N_B \cdot N_S} \quad (2)$$

$$L_B^D(\vec{p}, \vec{v}, c) = \frac{\sum_{i=1}^{N_B} \sum_{m=1}^{M_i} \tau_{im}(\vec{p}_{im}, \vec{v}_{im}, c) \sum_{j=1}^{N_S} L_j(\vec{p}_{im}, \vec{v}'_{imj}, c)}{N_B \cdot N_S} \quad (3)$$

где: M_i – количество сохраненных фотонов на поверхностях сцены для i -ой трассы луча;
 N_S – количество случайных попыток вычисления прямой освещенности $E_j(\vec{p}_{im}, \vec{v}'_{imj}, c)$ для метода “light sampling” и яркости источников света $L_j(\vec{p}_{im}, \vec{v}'_{imj}, c)$, находящихся на трассе луча после рассеивания луча на j -ой поверхности для метода “BDF sampling”.

Метод “light sampling” позволяет уменьшить шум изображения при освещении сцены источниками света малого размера и на поверхностях с ДФР близкими к Ламбертовым. С другой стороны метод “BDF sampling” улучшает качество изображения для сцен, освещенных протяженными источниками света и на поверхностях с узконаправленным ДФР. В результате, взвешенная сумма двух методов позволяет уменьшить шум изображения как вблизи областей резкого перепада яркости, так и в областях ее плавного изменения и не создает источника повышенного шума даже для поверхностей с узконаправленным ДФР.

Вторичная яркость вычисляется следующим образом:

$$L'(\vec{p}, \vec{v}, c) = \frac{\sum_{i=1}^{N_B} \sum_{m=1}^{M_i} \tau_{im}(\vec{p}_{im}, \vec{v}_{im}, c) \sum_{j=1}^{N_B} B S D F_{im}(\vec{p}_{im}, \vec{v}_{im}, \vec{v}'_{imj}, c) F_{Rj}(\vec{p}_{im}, \vec{v}'_{imj}, c)}{\pi^2 N_B N_F r_{im}^2} \quad (4)$$

где: N_F – количество лучей, испущенных источниками света;

N_R – количество фотонов, попавших в m -ую сферу интегрирования (фотонную карту) на трассе i -го луча;

r_{im} – радиусы сфер интегрирования вторичной яркости для i -го луча на m -ой поверхности наблюдения соответственно;

$F_{Rj}(\vec{p}_{im}, \vec{v}'_{imj}, c)$ – спектральный поток, переносимый j -ым фотоном в точке \vec{p}_{im} по направлению \vec{v}'_{imj} и попавший в сферу интегрирования фотонной карты видимости.

Поскольку световой поток, переносимый фотоном, есть величина, как правило, постоянная, то основным фактором, влияющим на шум, возникающий при расчете вторичной яркости на поверхностях сцены – это ДФР. Если ДФР близка к Ламбертовой, то вариация яркости от фотонов, приходящих в точку наблюдения с различных направлений, будет минимальна. Однако, если ДФР имеет узкона-

правленный характер, то вариация яркости может достигать нескольких порядков, что приводит к появлению ярких точек на изображении, на устранение которых необходимо затратить на порядки больше времени, чем в случае Ламбертового рассеивания. Поэтому для минимизации шума, возникающего при расчете вторичной яркости необходимо устанавливать точку расчета вторичной яркости на поверхностях с максимально диффузными свойствами.

Метод адаптивного выбора глубины трассы обратного луча

Как было отмечено выше, основной путь снижения шума, возникающего при расчете вторичной яркости, это выбор точки на трассе обратного луча, в которой оптические свойства поверхности имеют наиболее диффузный характер. Для выбора оптимальной точки, во-первых, необходимо осуществить трассировку луча на максимально возможную глубину и, во-вторых, произвести анализ оптических свойств всех поверхностей сцены на трассе луча. Данная последовательность действий несколько меняет алгоритм обратной трассировки луча и приводит к тому, что трасса луча сначала сохраняется полностью, а затем, после анализа на неоднородность оптических свойств, переводится в фотонную карту видимости до точки с максимально однородными свойствами.

Для анализа оптических свойств поверхностей был реализован специальный метод, вычисляющий коэффициент неоднородности для данного направления наблюдения (отношение максимальной яркости к средней яркости для выбранного направления наблюдения). Оптимальная глубина трассировки лучей выбирается для каждого направления луча наблюдения (точки изображения). Кроме того, если глубина трассировки обратного луча больше нуля, то для одного направления наблюдения может существовать множество «оптимальных» глубин трассировки луча, которые зависят от того, какую трассу выбрал конкретный стохастический луч. Однако, изменение глубины диффузной трассировки луча в процессе накопления вторичной яркости (т.е. для каждого луча свое оптимальное количество диффузных рассеиваний) приводит к некорректному решению уравнения рендеринга (4). Для минимизации случайных флуктуаций глубины диффузной трассировки луча в уравнении (4) полный расчет вторичной яркости разбивается на итерации, в каждой из которых глубина трассировки луча выбирается исходя из оптимальной глубины трассы первого луча в каждой точке изображения сцены и накопленной статистики оптимальных глубин. Финальное изображение есть усредненный результат всех выполненных итераций.

Рисунок 2 демонстрирует результат применения метода выбора оптимальной глубины трассировки лучей к расчету изображения стеклянной пластины, лежащей на диффузной подложке. Торцевые поверхности пластины гладкие и прозрачные, верхняя поверхность пластины имеет острую ДФР, а нижняя поверхность – широкую. В результате оптимальная глубина трассировки лучей на верхней пластине равна единице (второе диффузное рассеивание), а на прозрачном торце – нулю (первое диффузное рассеивание).

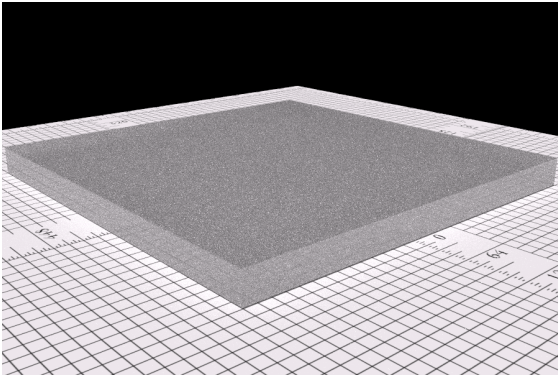


Рис. 2: Изображение, построенное методом выбора оптимальной глубины диффузной трассы луча

В данном случае шум распределен достаточно равномерно по изображению верхней и торцевым поверхностям пластины. Если глубину диффузной трассировки установить равной нулю для всей сцены, то шум на изображении верхней пластины возрастет, а при единичной глубине – шум возрастет на изображении торцевой поверхности пластины. Рисунки 3 и 4 демонстрируют данные случаи соответственно.

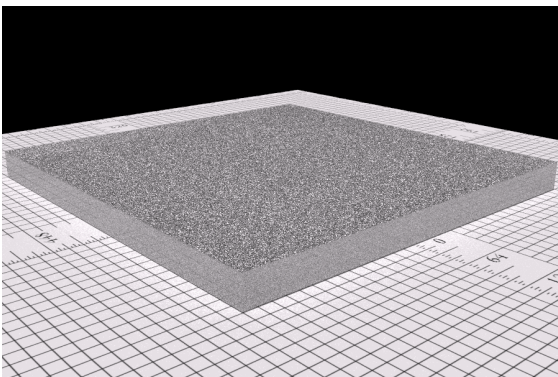


Рис. 3: Изображение, построенное при глубине диффузной трассы луча равной нулю

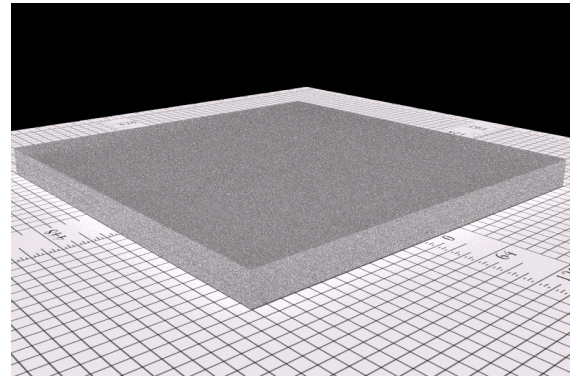


Рис. 4: Изображение, построенное при глубине диффузной трассы луча равной единице

Метод «квази-зеркальной» трассировки обратного луча

Сдвиг точки интегрирования вторичной яркости на трассе обратного луча может идти не только по пути выбора поверхности с максимально однородным рассеиванием, но и посредством применения метода «квази-зеркального» рассеивания света. Основная идея метода аналогична предыдущему решению – расчет вторичной яркости на поверхностях с узконаправленным ДФР является нецелесообразным. Однако реализация метода принципиально отлична от предыдущей и заключается в том, что рассеивание света на поверхностях сцены рассматривается как «квази-зеркальное» событие, т.е. событие, не добавляющее вторичной яркости, если рассеивание света осуществляется в некотором заданном конусе относительно направления «зеркального» луча, т.е. луча, претерпевшего зеркальное отражение или преломление по закону Снеллиуса. В результате трасса луча будет продолжаться до тех пор, пока направление рассеянного луча не выйдет из заданного конуса углов относительно «зеркального» направления. Естественно, если поверхность имеет узконаправленный ДФР, то большинство лучей, рассеянных на поверхности, будет иметь «квази-зеркальный» характер распространения и точка интегрирования вторичной яркости будет сдвинута. Если же поверхность имеет диффузный характер рассеивания, то при незначительной величине допустимого конуса рассеивания (до 10°) доля «квази-зеркальных» лучей будет крайне мала и не окажет значительного влияния на качество изображения за счет неоптимального переноса точки интегрирования.

Корректное решение уравнения рендеринга (2, 4), использующее метод «квази-зеркальной» трассировки лучей, требует дополнительной модификации алгоритма расчета первичной и вторичной яркостей. Во-первых, для поверхности, наблюдаемой после диффузного рассеивания, первичная яркость

не должна приходиться от источников света внутри допустимого конуса рассеивания лучей, а, во-вторых, при расчете вторичной яркости диффузное освещение так же должно приходиться на поверхность вне допустимого конуса рассеивания лучей, т.е. все освещение, приходящее внутри допустимого конуса рассеивания, должно обнулено. Рисунок 5 иллюстрирует метод «квази-зеркальной» трассировки обратного луча и соответствующие методы вычисления яркостей первичного и вторичного освещения.

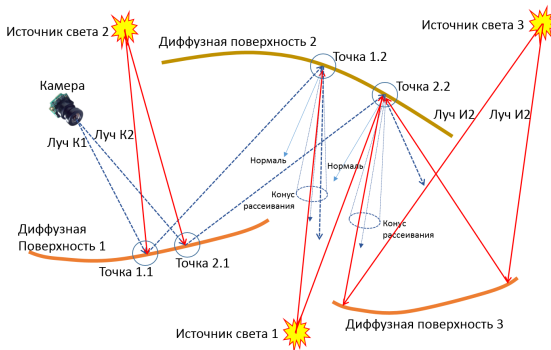


Рис. 5: Метод «квази-зеркальной» трассировки обратного луча

В качестве примера рассматривается длина трассы обратного луча, содержащая одно диффузное рассеивание. Первый луч, испущенный камерой, рассеивается на второй диффузной поверхности внутри конуса и продолжает распространение, сдвигая точку интегрирования яркости вторичного освещения на следующие поверхности сцены. Второй луч, испущенный камерой, рассеивается на второй диффузной поверхности вне заданного конуса, и поэтому точка рассеивания является центром области потенциального расчета вторичной яркости для данного направления наблюдения.

Рассмотрим методику расчета яркости первичного освещения. Если точки наблюдаются непосредственно (или после ряда «зеркальных» или «квази-зеркальных» преобразований), то осуществляется расчет яркости первичного освещения. Так второй источник света формирует первичную яркость в точках 1.1 и 2.1. В случае наблюдения после диффузного рассеивания яркость первичного освещения вычисляется только, если освещение происходит вне конуса рассеивания. Так первый источник света формирует первичную яркость в точке 2.2 и не добавляет яркость в точке 1.2.

Яркость вторичного освещения вычисляется в конце пути трассы обратного луча (окончание трассы определяется ее глубиной и рассеиванием луча вне допустимого конуса). Шанс на исключение яркости вторичного освещения появляется в случае,

когда на трассе обратного луча происходили события диффузного рассеивания или на трассе луча от источника света происходили события диффузного или «квази-зеркального» рассеивания. Достаточным условием того, чтобы исключить компоненту яркости вторичного освещения, является тот факт, что освещение происходит вне конуса рассеивания обратного луча и добавляет вторичную яркость, в то время как второй луч, прошедший по аналогичной траектории, не добавляет вторичной яркости, поскольку освещает поверхность внутри конуса рассеивания.

Метод «квази-зеркальной» трассировки обратного луча может быть совмещен с методом адаптивного выбора глубины обратного луча. Совмещение двух методов позволяет не только понизить шум изображения, но и повысить качество моделирования ряда «тонких» визуальных эффектов, например, вторичных бликов, т.е. бликов, наблюдаемых от бликующих поверхностей. На рисунке 6 показано изображение элемента подсветки заднего фонаря автомобиля. Элемент растрового световода имеет узконаправленную ДФР, что позволяет сдвинуть точку расчета вторичной яркости в область, близкую к источнику света, и тем самым повысить качество изображения.

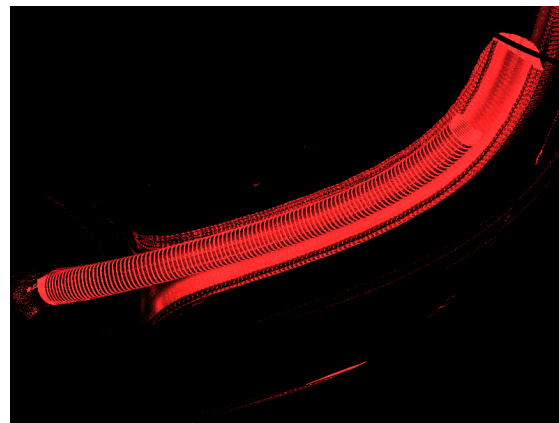


Рис. 6: Изображение элемента подсветки заднего фонаря автомобиля

Заключение

В рамках программного комплекса Lumiscept [6] были реализованы два физически корректных программных подхода, позволяющих ускорить сходимость расчета вторичной яркости методом двунаправленной стохастической трассировки лучей. Разработанные программные решения позволяют ускорить процесс получения изображения с допустимым уровнем шума, а в ряде случаев, возникаю-

щих при проектировании оптически сложных осветительных систем, содержащих поверхности, имеющие узконаправленный характер ДФР, являются единственным способом получения фотореалистичного изображения за ограниченное время вычислений.

Анализ неоднородности свойств оптических материалов был реализован в спектральном пространстве и учитывал спектральные свойства наблюдателя до поверхности наблюдения и спектральные свойства самой наблюдаемой поверхности.

Литература

- [1] J. T. Kajiya. The rendering equation // Computer Graphics (SIGGRAPH '86 Proceedings), 1986, vol. 20, pp.143-150.
- [2] Jensen Henrik Wann, Christensen Per. High quality rendering using ray tracing and photon mapping // ACM SIGGRAPH 2007 courses. SIGGRAPH'07, New York, NY, USA, ACM, URL: <http://doi.acm.org/10.1145/1281500.1281593>, 2007.
- [3] Pharr Matt, Humphreys Greg. Physically Based Rendering, Second Edition: From Theory To Implementation. 2nd edition. // San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2010, 1200p.
- [4] Hachisuka Toshiya, Jensen Henrik Wann. Stochastic progressive photon mapping // ACM Trans. Graph. 2010. – dec. Vol. 28, no. 5. p. 141:1–141:8. URL: <http://doi.acm.org/10.1145/1618452.1618487>, 2009.
- [5] Eric Veach. A dissertation: Robust Monte-Carlo methods for light transport simulation, URL: http://graphics.stanford.edu/papers/veach_thesis/thesis.pdf, 1997.
- [6] Intgera Inc., Lumicept, URL: <http://www.integra.jp/en>, 2015.

Мульти-серверная динамическая балансировка нагрузки для сетевых виртуальных окружений*

Джен-Руй Янг, Фу-Сианг Чанг

jrjiang@csie.ncu.edu.tw | majuschang@gmail.com

Факультет информатики и информационных технологий

Национальный Центральный Университет, Джонгли, Тайвань

Эта статья предлагает схему динамического распределения нагрузки (*load balancing* – *LB*), построенную на клеточной основе, а именно Управляемое Распределение Нагрузки (*Directed Load Diffusion* – *DLD*) для достижения баланса нагрузки между серверами в сетевом виртуальном окружении (*NVE*). Виртуальное окружение состоит из мелких шестиугольных элементов, называемых клетками, и сервер отвечает за управление областями, состоящими из нескольких соседних клеток. Разные серверы имеют разные возможности, и степень нагрузки сервера строится на использовании его потенциала. Схема *DLD* сохраняет степень нагрузки на сервер как можно равномернее, без ущерба для производительности. Мы провели обширные имитационные эксперименты для схемы *DLD* и сравнили её с двумя релевантными схемами балансировки нагрузки на клеточной основе, а именно, схемы *Ahmed* и *ProGReGA*, чтобы показать преимущества нашего подхода.

Ключевые слова: балансировка нагрузки, сетевое виртуальное окружение, горячая точка, массивная многопользовательская Интернет-игра.

Multi-Server Dynamic Load Balancing for Networked Virtual Environments*

Jehn-Ruey Jiang, Fu-Hsiang Chang

Department of Computer Science and Information Engineering

National Central University, Jhongli City, Taiwan

This paper proposes a cell-based dynamic load balancing (*LB*) scheme, namely *Directed Load Diffusion (DLD)*, to achieve load balance among servers in a networked virtual environment (*NVE*). The virtual environment is divided into small hexagonal areas called cells, and a server is responsible for managing a region consisting of many adjacent cells. Different servers have different capacities, and the load degree of a server is the utilization of its capacity. The *DLD* scheme is to keep server load degrees as even as possible without compromising performance. We perform extensive simulation experiments for the *DLD* scheme and compared it with two relevant cell-based *LB* schemes, namely *Ahmed* and *ProGReGA*, to show its advantages.

Keywords: Load Balancing, Networked Virtual Environment, Hotspot, Massively Multi-player Online Game.

1. Introduction

In a *networked virtual environment (NVE)*, players or users can assume roles of *avatars* to navigate in a computer generated realistic virtual world or *virtual environment (VE)* to interact with one another through networked links. *NVEs* are widely applied to a variety of areas like military simulation, education, training and network games. The *massively multi-player online game (MMOG)*, such as *World of Warcraft (WoW)* [7], which is nowadays a billion-dollar business, is a typical example of *NVEs*. Thousands or tens of thousands of *NVE* users may log on and navigate throughout the *VE* concurrently. Numerous servers are provided to cooperatively manage the users by receiving action messages from them, simulating the game, and sending game state update messages to relevant users. However, the dynamic actions of users may lead to load unbalance among servers. Several

load balancing (LB) schemes were thus proposed to flexibly balance *NVE* server loads [1-6].

LB schemes can be classified into four classes: shard-based, zone-based, partition-based, and cell-based. The shard-based *LB* scheme, as adopted by *WoW* [7], makes multiple *VE* duplicates (called shards or realms) run concurrently, with each duplicate being supported by a separate unified sever cluster. The overloaded situation of servers can be avoided by explicitly limiting the maximum number of users in a shard. When the number of users grows and exceeds a pre-specified threshold, a new duplicate is produced and supported by a new server cluster. The shard-based *LB* schemes have low computation overheads; however, they have one limitation that avatars in different shards cannot interact with one another.

The zone-based *LB* scheme divides the whole *VE* into fixed-sized zones or regions, each representing a country or city and managed by a server. Users have the chance to interact with one another since there is only a single shard. However, the passage from one

Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347

zone to another requires special portals (e.g., special tunnels) or teleporting. Since users tend to gather around hotspots where certain events occur and/or NPCs (non-player characters) appear, the server managing many hotspots may easily become overloaded.

The partition-based LB scheme, such as the kd-tree scheme and the schemes adopted by QuON [5] and VSO [6], divides the whole VE into various-sized partitions, each managed by a server. The VE partitioning is dynamical. For example, the kd-tree scheme, QuON and VSO uses the kd-tree, Quad-tree, and Voronoi-diagram structures, respectively, to grow or shrink partitions. Partition-based LB schemes have the advantage that the load of servers can be adjusted dynamically and flexibly to balance the server loads evenly; they have the disadvantage of high computation overheads, though.

The cell-based LB scheme, such as Ahmed [2] and ProGReGA [3], divides the whole VE into small *cells* (or *microcells* mentioned in some papers), a group of which is managed by a server. Typical cell shapes include triangles, squares, and hexagons. Different heuristic strategies are developed for assigning servers to manage different groups of cells. The cells managed by a server may or may not be contiguous and may be of any numbers; therefore, the load of an overloaded server can dynamically and flexibly be transferred to arbitrary other servers. The computation overheads of cell-based LB schemes are mediate.

This paper proposes a cell-based LB scheme, called *DLD (Directed Load Diffusion)*, which partitions the whole VE into hexagonal cells to assign a region of adjacent cells to be managed by a server. Different servers have different *capacities*, and the *load degree* of a server is the utilization of its *capacity*. The DLD scheme is to keep server load degrees as even as possible, while maintaining low load transfer overheads, low inter-server communications, and low load deviation ratios. We perform extensive simulation experiments for the DLD scheme and compare it with two relevant cell-based LB schemes, namely Ahmed [2] and ProGReGA [3], to show its advantages.

The rest of this paper is organized as follows. In Section 2, we introduce some related work. Section 3 elaborates the proposed scheme, and Section 4 presents its simulation results and comparisons. Finally, some concluding remarks are drawn in Section 5.

2. Related work

In this section, we introduce two cell-based LB schemes, namely Ahmed [2] and ProGReGA [3], which are most related to the proposed scheme. Below, we describe the schemes one by one.

The Ahmed scheme [2] measures the load of a server by the sum of message rates of all players the server must handle. A server whose sum of message rates

exceeds a pre-specified threshold is regarded as overloaded. To balance the load, the Ahmed scheme finds all clusters of adjacent cells managed by the overloaded server. The smallest cluster is selected and, from this cluster, the cell which has the least interaction with other cells of the same server is first selected to be transferred to the least loaded server. The selection process continues until the server is no longer overloaded or there is no server that can take over the transferred loads.

The ProGReGA (proportional greedy region growing algorithm) scheme [3] assumes that the whole VE is divided into regions, each of which consists of several cells and is managed by a server. Given a list of the regions to be rebalanced, all servers managing these regions share proportionally the loads of all regions according to the server power. Initially, the heaviest-loaded cell is selected to be the first cell of a region to be managed by the most powerful server. Afterwards, a neighbor cell adjacent to the heaviest-loaded cell which has the largest inter-cell interaction overheads is added into the region. The cell-addition continues until the most powerful server has shared the proportional loads. Then, the process continues to form the second region to be managed by the second powerful server by repeatedly executing the cell-addition process. Iteratively, the third region, the fourth region, ... are formed until every cell is added into one region to be managed by one server. One undesirable effect of the ProGReGA scheme is that after rebalancing, one or more regions may completely change the servers originally managing them, causing high load transfer overheads.

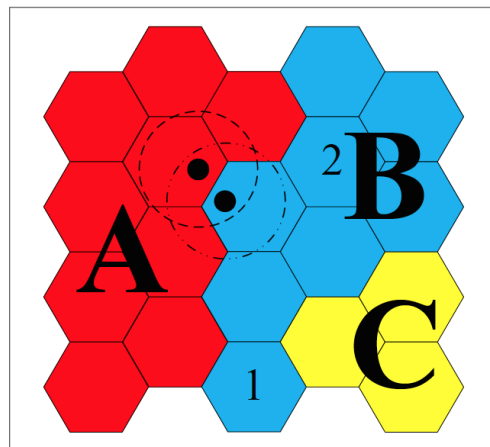


Figure 1: The hexagonal partitioning of a virtual environment.

3. PROPOSED SCHEME

3.1 Assumptions and Definitions

The whole SE is assumed to be divided into hexagonal cells, and a region containing some cells, not necessary

adjacent, is managed by a server. An avatar (i.e., player) navigating in the VE performs some actions and sends associated *action messages* to the region sever. The server then updates the VE state according to the actions and sends *update messages* to every avatar of which AOI (Area of Interest) includes the updates, where an avatar's AOI stands for the circular area of a fixed radius or range centered at the avatar. For example, black dots in Figure 1 represent avatars and dashed circles represent their AOIs. Two regions are adjacent if they have adjacent cells; two servers are adjacent if they manage adjacent regions. For example, regions A and B in Figure 1 are adjacent, but regions A and C are not adjacent. Two servers need to exchange messages if one server manages some boundary cells with avatars whose actions cause the updates included in the AOI of avatars in the cells managed by the other server. For example, in Figure 1, the server managing region A and the server managing region B need to exchange messages since they have avatars that are managed by different servers and influence each other.

It is assumed that the overall load of a sever comes from three aspects: (a) the *Action Load (AL)*, the load to process action messages sent by avatars, (b) the *Computation Load (CL)*, the load to compute the updates caused by action messages, and (c) the *Update Load (UL)*, the load to send update messages to relevant avatars. A server has a pre-specified *Capacity (CAP)* to take loads. In this paper, we define the *Server Load (SL)* to be the load of a server and the *Load Degree (LD)* to be the utilization of the server capacity, i.e., the ratio of the server load to the server capacity. Specifically, $LD = SL/CAP$.

To judge the load condition of a server, we have three thresholds for the load degree, as explained one by one in the following.

- (a) *Overload Threshold (OLT)*: A server is *overloaded* if its load degree exceeds OLT. An overloaded server can be a *giver server* to transfer loads to other servers.
- (b) *Safety-Load Threshold (SLT)*: A server is *normally loaded* if its load degree is between SLT and OLT. By SLT, we can derive two useful server attributes, the *Safety Capacity (SC)* and the *Giving Load (GL)*, according to the following calculations: $SC = (SLT - LD) \times CAP$ and $GL = (LD - SLT) \times CAP$. Note that when LD is larger than SLT , SC is defined to be 0. The SC of a server indicates the maximum extras loads that can be transferred from other servers into the server.
- (c) *Underload Threshold (ULT)*: A server is *lightly loaded* if its load degree is between ULT and SLT; it is *underloaded* if its load degree is beneath ULT. A normally loaded or a lightly loaded server can be a *taker server* to take over loads from other

servers. The ULT threshold is very useful for performing LB at the presence of hotspots, which will be described later.

Note that the server load degree and the above-mentioned threshold values are between 0 (no load) and 1 (full load) and $OLT > SLT > ULT$. For a specific server, we below define the *Local Load Degree (LLD)*, to estimate the server load conditions for its neighborhood of itself, indexed by 0, and its adjacent n neighbor servers, indexed by $1, \dots, n$.

$$LLD = \frac{\sum_{i=0}^n SL_i}{\sum_{i=0}^n CAP_i}$$

3.2 The Proposed Algorithm

In this subsection, we describe the proposed *Directed Load Diffusion (DLLD)* scheme, which is a greedy load balancing algorithm trying to keep every server *normally loaded*. A server can be a *taker server* to take over other servers' loads if it is lightly loaded or underloaded (i.e., its load degree is beneath SLT). A server X checks its load degree every *Check Load Period (CLP)*. If X is overloaded, it then performs *General Load Balancing (GLB)*, described below. X first calculates LLD to evaluate the load conditions for its neighborhood of itself and all its neighbor servers. If the calculated LLD is smaller than SLT , then X 's neighborhood is assumed to be lightly loaded (or underloaded) and X can transfer its load to the neighbor server Y with the lowest server load (i.e., SLY). The load transferred from X to Y is $\min(GLX, SCY)$. Otherwise, X 's neighborhood is assumed to be heavily loaded. It is probably that every neighbor server of X has a load degree exceeding or approximating SLT . In that case, X transfers its load to the neighbor server Z with the largest safety capacity (i.e., SCZ). The load transferred from X to Z is $\min(GLX, SCZ)$. Note that X transfers its load to only one server for GLB within a CLP; if X is still overloaded after the load transfer, X will transfer its load within next CLP.

It is likely that all servers in the neighborhood of a hotspot have load degrees exceeding SLT so that an overloaded server X may not transfer its load to any other server for GLB. In that case, X performs *Forced Load Balancing (FLB)* by sending an FLB Request (FLB-Req) message to all neighbor servers. When server Y receives the FLB-Req message sent by X , Y is forced to transfer its load to its neighbors so that Y is lightly loaded to take over X 's load. To prevent Y from transferring its load to a server that is X 's and Y 's common neighbor, the FLB-Req message sent by X contains a *Forbidden Transfer List (FT-List)* including X and all neighbors of X . Y transfers its load to its every neighbor server not in FT-List according to the "*high safety capacity first*" order. The load transferred from Y to Z caused by X 's FLB-Req

message is $\min((LD_Y - ULT) \times CAP_Y, SC_Z)$ so that Y becomes lightly-loaded and Z becomes normally-loaded after the load transfer. If Y cannot make itself lightly-loaded, Y forwards X 's FLB-Req message to neighbors not in the FT-List to further make them underloaded. However, if all Y 's neighbors are in the FT-List, Y just stops FLB. Note that Y 's neighbors are added into the FT-List before Y forwards X 's FLB-Req message.

We have above described when and how much to transfer load between servers. Below, we describe how to select cells for load transfer. In practice, a server transfers its load by reducing the number of cells managed by it. For better performance, the server tries to keep all the cells adjacent but not to be of the shape of narrow strips, which may cause more inter-server communications. We use the *Cell Weight (CW)* to help cell selection. The CW of a cell c is defined to be the ratio of the number of c 's adjacent cells managed by different servers to the total number of c 's adjacent cells. To take Figure 1 for example, cell 1 managed by server B has the CW of $2/3$, while cell 2, $1/6$. The larger the CW of a cell is, the higher the priority of the cell to be selected as a cell candidate (CC) to be transferred to another server.

There are two options of the DLD scheme: DLD with Load Constrains (DLD-wLC) and DLD without Load Constrains (DLD-woLC). DLD-wLC has the limitation that the load involved in a CC should not exceed the load to be transferred in GLB or FLB, while DLD-woLC does not have the limitation. The former (resp., the latter) can (resp., cannot) prevent the load of the taker server from exceeding SLT too much but makes LB more inflexible (resp., flexible).

4. SIMULATION

4.1 Simulation Settings

We limit a server to manage at least one cell but not limit the maximum number of cells that a server can manage. We assume there are 8 servers and 800 avatars. Furthermore, there may be 3 hotspots or no hotspot in the VE. OLT, SLT, and ULT are set as 0.9, 0.8, and 0.6, respectively. We set the server capacity to be 1600 units and the load caused by avatar actions to be arbitrarily 3, 5, or 10 units. The AOI radius and the cell side length are both set to 4 units. The avatar speed is assumed to be 0 to 2 units per second, and the avatar follows the random waypoint mobility model with 50% of choosing the direction toward the closest hotspot. The CLP is set as 32 seconds. Table 1 shows all the simulation settings.

Table 1: Simulation Settings

The number of cells	224 (14×16)
The number of servers	8
The number of avatars	800
Server capacity	1600
Action load	1
Update load	2
Computation load	3, 5, or 10 (random)
AOI radius	4 units
Cell side length	4 units
Avatar Speed	0~2 units (random)
Check load period	32 sec
Simulation duration	30 min

4.2 Comparisons

We simulate the proposed DLD scheme and compare its two options (i.e., DLD-wLC and DLD-woLC) with two related schemes, namely the Ahmed [2] and ProGreGA [3]. The comparisons are shown below in terms of the following overheads: avatar migration, inter-server communication, and load deviation ratio, each in a separate subsection.

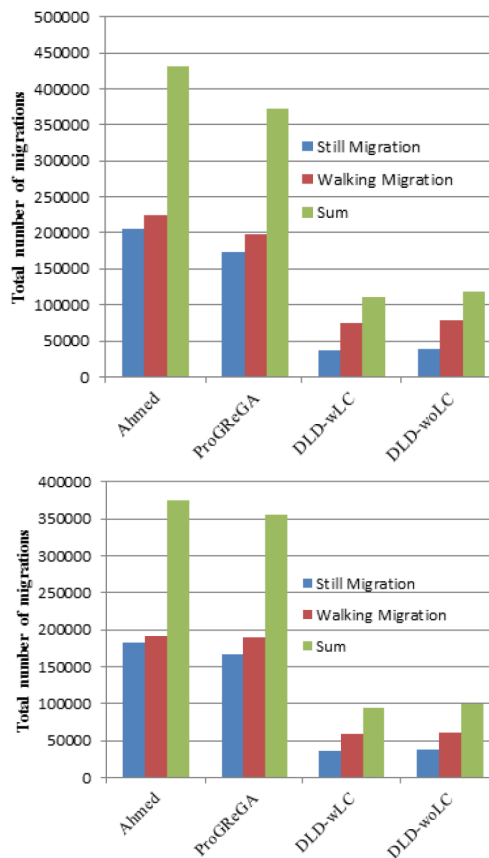


Figure 2: Avatar Migration Comparisons of LB Schemes with hotspots (up) or without hotspots (down).

4.2.1 Avatar Migration

Avatar migration has two portions: walking migration and still migration. When an avatar navigates in

the VE and walks from one cell managed by a server to another cell managed by another region, the *walking migration* occurs and the avatar's data should be transferred from the former server to the latter. When servers perform LB to transfer loads of some cells from one server to another server, the *still migration* occurs and all data of the avatars in the cells should be transferred from the former server to the latter. By Figure 2, we can observe that both DLD-wLC and DLD-woLC outperforms the other two schemes in terms of walking migration, still migration, and total avatar migration whether there are hotspots or not.

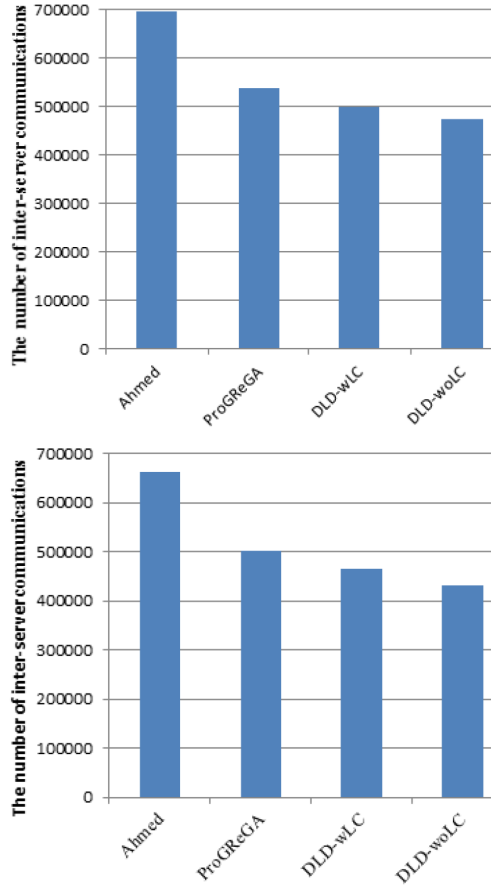


Figure 3: Inter-Server Communication Comparisons of LB Schemes with hotspots (up) or without hotspots (down).

4.2.2 Inter-server Communication

As we have mentioned, two servers need to exchange messages if one server manages some boundary cells with avatars whose actions cause the updates included in the AOI of avatars in the cells managed by the other server. Such a case causes the overheads of inter-server communication. By Figure 3, we can observe that both DLD-wLC and DLD-woLC outperforms the other two schemes in terms of inter-server communication whether there are hotspots or not.

4.2.3 Load Deviation Ratio

The Load Deviation Ratio (LDR) of a server X is defined to be the ratio of the difference between X 's load and the average load to the average load. For example, if the average load is 0.8 and X 's load is 0.85, then X 's LDR is $|0.85 - 0.8| / 0.8 = 0.0625$. The LDR of a scheme is the average LDR of all servers. By Figure 4, we can observe that DLD (precisely DLD-woLC) outperforms the Ahmed and ProGrEgA schemes in terms of LDR.

5. CONCLUSION

This paper proposes DLD, a cell-based dynamic load balancing scheme to achieve load balance among multiple NVE servers. DLD tries to keep servers normally-loaded and makes an overloaded server transfer its load to lightly-loaded or underloaded servers. Extensive simulation experiments are performed to compare the DLD scheme with the Ahmed and ProGrEgA schemes. The proposed scheme outperforms others in terms of overheads of avatar migration, inter-server communication and load deviation ratio.

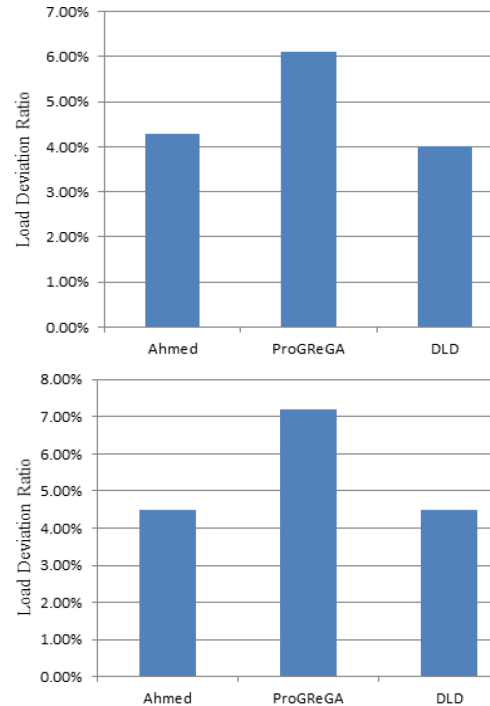


Figure 4: Load Deviation Ratio Comparisons of LB Schemes with hotspots (up) or without hotspots (down).

Bibliography

- [1] S. A. Abdulazeez, A. El Rhalibi, M. Merabti and D. Al-Jumeily, "Multi-Server Dynamic Load Balancing for Networked Virtual Environments", in Proc. of The 15th Post Graduate Network Symposium (PGNet), 2014.

- [2] D. T. Ahmed, and S. Shirmohammadi, "A Microcell Oriented Load Balancing Model for Collaborative Virtual Environments", in Proc. of 2008 IEEE Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems, pp.86 – 91, 2008.
- [3] C. E. Bezerra and C. F. R. Geyer, "A load balancing scheme for massively multiplayer online games", Journal of Multimedia Tools and Applications, Vol. 45, Issue 1-3, pp.263-289, 2009.
- [4] C. E. Bezerra, J. L. D. Comba, and C. F. R. Geyer, "A fine granularity load balancing technique for MMOG servers using a kd-tree to partition the space", in Proc. of 2009 IEEE VIII Brazilian Symposium on Games and Digital Entertainment (SBGAMES), 2009.
- [5] H. Backhaus and S. Krause, "QuON: a quad-tree-based overlay protocol for distributed virtual worlds", Int. Journal of Advanced Media and Communication, vol. 4, no. 2, pp.126-139, 2010.
- [6] S.-Y. Hu and K.-T. Chen, "VSO: Self-Organizing Spatial Publish Subscribe", 2011 IEEE 5th Int. Conf. on Self-Adaptive and Self-Organizing Systems, pp.21–30, Oct. 2011.
- [7] World of Warcraft, <http://us.battle.net/wow/en>.

Сегментация изображений магнитно-резонансной томографии головного мозга с помощью сопоставления с несколькими атласами*

А. Ю. Зубов, О. В. Сениюкова

a31ex33@yandex.ru | osenyukova@graphics.cs.msu.ru

Факультет вычислительной математики и кибернетики,

Московский государственный университет имени М.В. Ломоносова, Москва, Россия

Рассматривается задача автоматической сегментации изображений магнитно-резонансной томографии (МРТ) головного мозга человека на анатомические структуры. В настоящее время наиболее широко применяются алгоритмы сегментации, основанные на регистрации (сопоставлении) с атласом — изображением, для которого известна экспертная разметка. Сегментация на основе регистрации с несколькими атласами позволяет в некоторой степени компенсировать ошибки сопоставления с каждым отдельным атласом. В данной работе предлагается модификация лучшего на данный момент алгоритма слияния разметок нескольких атласов с целью получения разметки входного изображения, которая обеспечивает более эффективную реализацию. Алгоритм был применен к задаче разбиения изображений МРТ мозга на 43 анатомических региона с использованием открытой базы IBSR, в отличие от исходной работы авторов, где приведены результаты тестирования для задачи сегментации только одной анатомической структуры (гиппокампа).

Ключевые слова: сегментация изображений, регистрация с атласом, разметка на анатомические структуры, магнитно-резонансная томография.

Segmentation of magnetic resonance images of human brain by registration with multiple atlases*

A. Y. Zubov, O. V. Senyukova

Faculty of Computational Mathematics and Cybernetics,

Lomonosov Moscow State University, Moscow, Russian Federation

This paper is devoted to the problem of automatic segmentation of magnetic resonance images of human brain into anatomical structures. Currently, the most popular segmentation algorithms are based on registration (matching) of an input image with atlas, an image with manual expert labelling. Segmentation by registration with multiple atlases allows compensating errors caused by registration with each of these atlases, to some extent. In this work we propose a more effective implementation of the state-of-the-art segmentation algorithm that uses label fusion from multiple atlases. The algorithm was applied to the problem of segmentation of brain MRI into 43 anatomical regions using IBSR database, unlike the original work, where the authors provide the evaluation results only for segmentation of hippocampus.

Keywords: image segmentation, registration with atlas, anatomical labelling, magnetic resonance imaging.

Введение

Метод магнитно-резонансной томографии (МРТ) широко применяется для диагностирования и отслеживания динамики заболеваний мозга [12, 6], а также для исследования его функционирования [9]. Метод позволяет получать трехмерные изображения высокого качества и разрешения, которые строятся по набору последовательных двумерных «срезов».

Разметка изображений МРТ мозга на анатомические структуры является важным шагом для дальнейшего анализа во многих задачах в этой области. Полная разметка трехмерного изображения предполагает разбиение (сегментацию) объема мозга на несколько десятков регионов, соответствующих основным анатомическим структурам. Каждой точ-

ке (вокселю) ставится в соответствие метка анатомической структуры. Таким образом, ручная разметка в данном случае становится длительным и трудоемким процессом. Поэтому необходимы алгоритмы, автоматизирующие процесс анатомической разметки. В настоящей работе предлагается модификация лучшего на данный момент алгоритма автоматической анатомической разметки, который ранее применялся к задаче выделения одной из основных анатомических структур — гиппокампа, а также разбиения его на подструктуры.

Статья организована следующим образом. Краткий обзор существующих методов приведен в Разделе 2. Раздел 3 посвящен описанию базового алгоритма и предложенных авторами настоящей работы улучшений. Результаты тестирования авторского алгоритма и модифицированного алгоритма на реальных данных из открытой базы, а также де-

Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347

тали реализации приведены в Разделе 4. Выводы и обсуждение результатов — в Разделе 5.

Обзор существующих методов

Задача автоматической сегментации изображений МРТ головного мозга осложняется такими факторами, как неравномерность освещенности изображений, артефакты интерполяции при построении объемного изображения по последовательности плоских срезов, одинаковые значения интенсивности у различных анатомических структур. Поэтому задача становится практически неразрешимой при отсутствии *априорной* анатомической информации. Подобная информация может быть представлена в виде набора правил, сформулированных экспертами. Но наиболее эффективным подходом оказалось сопоставление с *атласом* [4]. Атлас в данном случае представляет собой набор из двух трехмерных изображений мозга, одно из которых является изображением МРТ (шаблон), а второе хранит в каждом вокселе значение метки соответствующей анатомической структуры (разметка). При наличии атласа задача сегментации сводится к задаче *регистрации* (сопоставления) изображения МРТ из атласа и входного изображения, которое необходимо разметить. К изображению из атласа применяется преобразование, которое делает его максимально похожим на входное изображение. Обычно регистрация производится в два этапа. Сначала применяется линейное преобразование (аффинное или жесткое), которое является глобальным и позволяет быстро компенсировать основные различия в местоположении и масштабе. Затем применяется локальное преобразование, чтобы более точно сопоставить анатомические структуры. После того, как изображение из атласа было полностью сопоставлено с входным изображением, можно перенести преобразованную таким же образом разметку из атласа на входное изображение для получения его сегментации на анатомические структуры.

В работах [10, 16, 1] показано, что использование более одного атласа позволяет повысить точность автоматической анатомической разметки. В этом случае несколько атласов регистрируются с входным изображением, и производится слияние их разметок для получения разметки входного изображения.

Большинство существующих алгоритмов используют схему взвешенного голосования [11, 13, 7, 8, 2], где каждый атлас вносит свой вклад в принятие решения о присвоении каждой точке входного изображения той или иной метки пропорционально своему весу. Вес атласа определяется схожестью с входным изображением. Наиболее успешными из этих алгоритмов на практике оказались те, которые вычисляют веса не для изображения целиком,

а в каждой точке по отдельности, в зависимости от локальной схожести атласа и входного изображения [13, 8, 2]. Общей проблемой данной группы алгоритмов является то, что они не учитывают тот факт, что ошибки, привносимые разными атласами, могут коррелировать. Если в голосовании участвуют несколько атласов, предлагающих некорректную метку в одной и той же точке, точность алгоритма падает.

Существует ряд работ, где предлагается проводить голосование с использованием только некоторых атласов из всего набора, которые наиболее похожи на входное изображение [2, 5]. Данный подход позволяет минимизировать влияние указанной выше проблемы, но, в то же время, теряется преимущество использования голосования по большому количеству атласов.

Наиболее перспективный на данный момент подход описан в работе [15]. Алгоритм позволяет минимизировать коррелирующую ошибку разных атласов, при этом, не отказываясь от голосования по всем атласам. Ниже приведено описание алгоритма и предложенной авторами настоящей работы модификации.

Алгоритм слияния разметок

В работе [15] описан новый способ слияния разметок нескольких атласов для получения разметки входного изображения. Предполагается, что перед началом работы алгоритма изображения атласов уже сопоставлены с входным изображением. Сопоставление производится, как было описано в предыдущем разделе, в два этапа: глобальная линейная регистрация — с помощью инструмента FSL FLIRT [14] и локальная нелинейная регистрация — с помощью алгоритма ANTS Symmetric Normalization (SyN) [3].

Основные идеи алгоритма и понятия

Основная идея алгоритма опирается на тот факт, что разметка изображения сильно коррелирует с его внешним видом. Определим входное изображение $I(x)$ как $I : x \in R^3 \rightarrow I(x) \in R$, где x задает координаты точки в трехмерном пространстве (вокселя), а I — значение интенсивности в этой точке. Анатомическую разметку изображения $C(x)$ определим как $C : x \in R^3 \rightarrow C(x) \in L$, где $L = \{1 \dots K\}$ — набор меток K анатомических структур. Тогда атлас представляет собой пару $\{I(x), C(x)\}$.

Как и многие другие алгоритмы с использованием нескольких атласов, данный алгоритм проводит слияние разметок путем взвешенного голосования. Делается предположение, что ошибки сегментации, привносимые разными атласами, могут коррелировать друг с другом. Учитывая это предположение, для минимизации ошибок сегментации, веса атласов при взвешенном голосовании рассчитываются с

учетом похожести входного изображения на пары атласов, а не на каждый из атласов по отдельности.

Процесс сегментации и слияния разметок

Предположим, что для каждой точки входного изображения получены веса, определяющие меру схожести с каждым из атласов. Тогда, для общего случая взвешенного голосования с n атласами, влияние метки l в точке x можно определить как

$$\sum_{i=1}^n w_i(x) * S_l^i(x) \quad (1)$$

где $w_i(x)$ – значение веса i -го атласа в точке x ; $S_l^i(x) = 1$, если метка i -го атласа в точке x равна l , и $S_l^i(x) = 0$ иначе. Для каждой точки происходит подсчет влияния каждой из меток, и алгоритм выбирает метку с наибольшим влиянием.

Существует множество стратегий выбора весов $\mathbf{w}(\mathbf{x}) = [w_1(x), \dots, w_n(x)]^t$. Как уже говорилось ранее, данный алгоритм пытается учесть попарное соответствие атласов входному изображению, рассчитывая веса специальным образом для каждой из точек входного изображения. Авторы алгоритма [15] предлагают использовать матрицы попарной зависимости атласов, которые позволяют оценить вероятность попарных ошибок атласов в каждой из точек. Для упрощения рассуждений, предположим, что существует только две метки – 0 и 1. Тогда решается задача бинарной сегментации, и отличие полученной сегментации от истинной в каждой из точек изображения равно либо 1 – значит, произошла ошибка, либо 0 – ошибки нет. Последующие рассуждения очевидным способом переносятся на случай нескольких меток, например, отделяя каждую из меток от всех остальных. В таком случае, в каждой из точек изображения матрица попарной зависимости n атласов $\mathbf{M}(\mathbf{x})$ определяется как матрица размера $n \times n$, элемент которой с индексами (i, j) определяет вероятность того, что оба атласа с индексами i и j предложили неверные метки для входного изображения в данной точке. Предположим, что в каждой точке известна матрица попарной зависимости атласов. С помощью данной матрицы можно сформулировать и решить задачу минимизации ожидаемой разницы между истинной сегментацией и полученной следующим образом:

$$w^*(x) = \arg \min_{w(x)} w^t(x) \mathbf{M}(\mathbf{x}) \mathbf{w}(x), \sum_{i=1}^n w_i(x) = 1 \quad (2)$$

Таким образом, в этом случае атласы, дающие коррелирующую ошибку в данной точке, получат меньшие веса, чем в случае независимого расчета весов для каждого атласа.

Если матрица $\mathbf{M}(\mathbf{x})$ не полного ранга, то задача (2) имеет не одно решение. Для получения единственного решения авторы алгоритма предлагают

добавить к матрице $\mathbf{M}(\mathbf{x})$ единичную матрицу, домноженную на некоторый коэффициент α , и решать задачу минимизации для полученной матрицы. Для корректности расчета весов $\mathbf{w}(\mathbf{x})$ элементы матрицы $\mathbf{M}(\mathbf{x})$ приводятся к некоторому диапазону значений, а величина α берется как некоторая процентная составляющая от максимального элемента.

Получение приближенной матрицы попарной зависимости. Вышеописанный подход основывается на том, что известна матрица попарной зависимости атласов. Данную матрицу можно получить приближенно, если ввести некоторые упрощения. Как уже говорилось ранее, в биомедицинских изображениях разметка коррелирует с внешним видом изображения. Далее можно сделать два предположения, которые подтверждаются на практике. Во-первых, вероятность того, что два атласа неверно предложат метку, не зависит от остальных атласов. Во-вторых, для определения данной вероятности в каждой из точек изображения достаточно учитывать некоторую локальную окрестность данной точки. Существует несколько метрик, использующих данные предположения для получения матрицы попарной зависимости. Авторы алгоритма предложили использовать метрику обратного расстояния в локальной окрестности радиуса r :

$$M_{i,j}(x) = \left(\sum_{y \in N_r(x)} |I_t(y) - I_i(y)| \cdot |I_t(y) - I_j(y)| \right)^\beta \quad (3)$$

где $M_{i,j}(x)$ – элемент матрицы $\mathbf{M}(\mathbf{x})$ с индексами (i, j) , I_t – входное изображение, а I_i – изображение из i -го атласа.

Предложенная модификация алгоритма. Алгоритм был протестирован на изображениях магнитно-резонансной томографии головного мозга для задачи сегментации всех размеченных анатомических структур, в то время как авторы [15] тестировали алгоритм только для сегментации гиппокампа. Задача сегментации мозга целиком требует в несколько раз больше памяти и временных затрат, поэтому необходима оптимизация работы алгоритма как по памяти, так и по скорости. Предложенная авторами [15] реализация алгоритма предлагает ускорение алгоритма с помощью различных оптимизаций и векторной обработки данных, но при этом требуется хранение всех используемых атласов и исходного изображения в оперативной памяти. В настоящей работе предложена реализация алгоритма с иным ускорением, которое позволяет работать в почти любых условиях по объему оперативной памяти и числу атласов, а по времени работы практически не отличается от авторской реализации.

Для совершения предложенных оптимизаций изображение разделяется на равные части. Работа ве-

дятся для каждой из выделенных частей изображения, что позволяет хранить промежуточную информацию только для данной части, а не для всего изображения, уменьшая затраты. Для каждой точки выделенной части вычисляется вектор объемов отличия атласов от исходного изображения в некоторой локальной окрестности. Объем каждого элемента вектора указывает разницу между исходным изображением и соответствующим атласом. В дальнейшем, данный вектор перемножается на свою транспонированную версию для создания нормализованной матрицы попарных объемов отличий атласов от исходного изображения, из которой можно получить матрицы попарной зависимости в каждой из точек изображения путем применения трехмерной свертки по координатам изображения с единичным ядром радиуса локальной окрестности, что эквивалентно по результату операции суммированию (3) во всех точках выделенной части изображения. Дальнейшие шаги алгоритма аналогичны исходному. Применение многомерной свертки увеличивает время работы алгоритма в зависимости от размера ядра свертки, которое зависит от радиуса локальной окрестности, в которой вычисляется матрица попарной зависимости. Однако, в общем случае время работы алгоритма также снижается после применения свертки за счет более эффективного суммирования по объему при вычислении матриц попарной зависимости и оптимизаций операции свертки. В общем случае, время работы зависит логарифмически от размера входного изображения, что в случае больших входных изображений покрывает предыдущие расходы. Также алгоритм сокращает затраты памяти в степенное число раз. Данные затраты становятся более существенными по приближению размера выделенной части к размеру ядра свертки. Пользователь может контролировать степень уменьшения затрат оперативной памяти.

Результаты

Модифицированный алгоритм был реализован на языке C++ и сравнивался с авторской реализацией, которая написана также на языке C++. В Таблицах 1 и 2 представлены результаты работы алгоритма, полученные после его применения к каждому из изображений базы IBSR [17], где в качестве атласов были использованы все остальные изображения вместе с их разметками. Были выбраны параметры алгоритма, показывающие наилучшие результаты в общем случае: $\alpha = 0,1$ от максимального значения матрицы попарных зависимостей атласов, $\beta = 2$, радиус локальной окрестности $r = 3$.

Как видно из представленных результатов, точность работы предложенной реализации алгоритма практически совпадает с точностью работы авторской реализации, включающей в себя иные улуч-

Таблица 1: Результаты работы алгоритма. DSC — мера сходства Дайса, усредненная по всем анатомическим структурам.

№	DSC Авторская реализация	DSC Предложенная реализация
1	0,7450	0,7402
2	0,77	0,7603
3	0,7804	0,7814
4	0,7946	0,7406
5	0,7634	0,7632
6	0,795	0,7913
7	0,713	0,7097
8	0,75	0,7546
9	0,7894	0,7857
10	0,7163	0,7157
11	0,80	0,7970
12	0,8321	0,8237
13	0,74	0,7421
14	0,7231	0,7141
15	0,7902	0,78
16	0,7427	0,7393
17	0,7384	0,7278
18	0,792	0,7957

Таблица 2: Точность выделения некоторых структур.

Анатомическая структура	DSC
Мозжечок	0,885 ± 0,05
Паллидум	0,7442 ± 0,009
Желудочек	0,9 ± 0,02
Кровеносные сосуды	0,2 ± 0,001
Средний мозг	0,8474 ± 0,0073

шения алгоритма. Время работы авторской и предложенной реализации алгоритма также практически совпадает и находится в пределах 190 секунд для одного трехмерного изображения. При этом, в предыдущем разделе показано, что модифицированный алгоритм значительно эффективнее по объему используемой памяти. Пример работы алгоритма представлен на Рисунке 1.

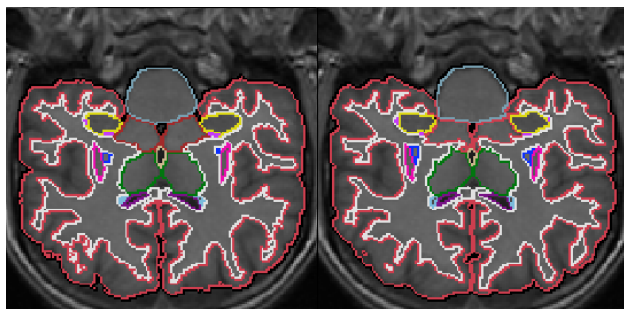


Рис. 1: Пример работы алгоритма. Слева — истинная разметка; справа — полученная разметка.

Выводы

Проведен анализ существующих алгоритмов сегментации изображений МРТ головного мозга на анатомические структуры и выявлено, что алгоритмы сегментации на основе регистрации с несколькими атласами обеспечивают наибольшую точность. Выбран лучший на данный момент алгоритм, который учитывает корреляцию ошибок разных атласов при проведении взвешенного голосования. Предложена модификация данного алгоритма, которая позволяет сделать его более эффективным по используемой памяти. Проведено тестирование авторской реализации и предложенной реализации на всей базе IBSR методом кросс-валидации и приведены результаты тестирования с учетом всех анатомических структур.

Проанализировав полученную сегментацию более подробно, можно выделить следующие результаты. Большинство структур выделяются с одинаковой точностью — порядка 80-90% корректно выделенного объема от общего объема структуры. Исключение составляют структуры малых размеров, занимающие порядка 0,01% от общего размера изображения. На некоторых из этих структур точность работы алгоритма существенно снижается: алгоритм не способен определить центр структуры и оценить ее размеры. Но в большинстве случаев алгоритм практически верно (с ошибкой до трех вокселей) определяет центр основных структур и их размеры.

Литература

- [1] P. Aljabar, R.A. Heckemann, A. Hammers et al. Multi-atlas based segmentation of brain images: atlas selection and its effect on accuracy // *NeuroImage*, vol. 46, no. 3, pp. 726–738, 2009.
- [2] X. Artaechevarria, A. Munoz-Barrutia, C. Ortiz de Solorzano Combination strategies in multi-atlas image segmentation: Application to brain MR data // *IEEE TMI*, vol. 28, no. 8, pp. 1266–1277, 2009.
- [3] B. Avants, C. Epstein, M. Grossman, J. Gee Symmetric diffeomorphic image registration with crosscorrelation: Evaluating automated labeling of

elderly and neurodegenerative brain // *Medical Image Analysis*, vol.12, no. 1, pp. 26–41, 2008.

- [4] M. Cabezas, A. Oliver, X. Lladó et al. A review of atlas-based segmentation for magnetic resonance brain images // *Computer methods and programs in biomedicine*, vol. 104, pp. 158–177, 2011.

- [5] *D. Collins, J. Pruessner* Towards accurate, automatic segmentation of the hippocampus and amygdala from MRI by augmenting ANIMAL with a template library and label fusion // *NeuroImage*, vol. 52, no. 4, pp. 1355–1366, 2010.
- [6] *Y. Ge* Multiple sclerosis: the role of MR imaging // *Am. J. Neuroradiol.*, vol. 27, no. 6, pp. 1165–1176, 2006.
- [7] *R. Heckemann, J. Hajnal, P. Aljabar et al.* Automatic anatomical brain MRI segmentation combining label propagation and decision fusion // *NeuroImage*, vol. 33, pp. 115–126, 2006.
- [8] *I. Isgum, M. Staring, A. Rutten et al.* Multi-atlas-based segmentation with local decision fusion application to cardiac and aortic segmentation in CT scans // *IEEE Trans. on MI*, vol. 28, no. 7, pp. 1000–1010, 2009.
- [9] *R.E. Jung, J.M. Segall, H.J. Bockholt et al.* Neuroanatomy of creativity // *Hum. Brain Mapp.*, vol. 31, no. 33, pp. 398–409, 2010.
- [10] *A. Klein, B. Mensh, S. Ghosh et al.* Mindboggle: automated brain labeling with multiple atlases // *BMC Med. Imag.*, vol. 5, no. 7, 2005.
- [11] *T. Rohlfing, R. Brandt, R. Menzel, C. Maurer* Evaluation of atlas selection strategies for atlas-based image segmentation with application to confocal microscopy images of bee brains // *NeuroImage*, vol. 21, no. 4, pp. 1428–1442, 2004.
- [12] *A. Rovira, J. Swanton, M. Tintoré et al.* A single, early magnetic resonance imaging study in the diagnosis of multiple sclerosis // *Arch. Neurol.*, vol. 66, no. 5, pp. 587–592, 2009.
- [13] *M. Sabuncu, B.T.T. Yeo, K. Van Leemput et al.* A generative model for image segmentation based on label fusion // *IEEE TMI*, vol. 29, no. 10, pp. 1714–1720, 2010.
- [14] *S.M. Smith, M. Jenkinson, M.W. Woolric et al.* Advances in functional and structural MR image analysis and implementation as FSL // *Neuroimage*, vol. 23, no. 1, pp. 208–219.
- [15] *H. Wang, J.W. Suh, S.R. Das et al.* Multi-Atlas Segmentation with Joint Label Fusion // *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 35, no. 3 pp. 611–623, 2013.
- [16] *M. Wu, C. Rosano, P. Lopez-Garcia et al.* Optimum template selection for atlas-based segmentation // *NeuroImage*, vol. 34, no. 4, pp. 1612–1618, 2007.
- [17] The Internet Brain Segmentation Repository (IBSR) <http://www.nitrc.org/projects/ibsr>.

Алгоритмы построения и распознавания навигационных описаний маршрутов для картографических компьютерных систем

В.А. Галактионов, М.В. Пестун

vlgal@gin.keldysh.ru|max.pestun@gmail.com

Институт прикладной математики им. М.В. Келдыша Российской академии наук, Москва, Россия

Компьютерные навигационно-картографические системы, доступные на текущий момент, предлагают возможности по нахождению оптимального пути из точки А в точку В. Однако большинство из них не позволяет ввести тот путь, который желает пользователь, и узнать, например, время движения конкретно по нему с учетом текущих заторов на дорогах. Те системы, которые позволяют это сделать, используют графические интерфейсы пользователя, предлагая мышкой разместить опорные точки маршрута, между которыми система построит маршрут в автоматическом режиме. Такой способ крайне плохо применим для мобильных платформ, где размер экрана сильно ограничен, а ввод данных пальцами имеет низкую точность позиционирования. Помимо этого, любой графический ввод информации неприменим с антропоморфными роботами, взаимодействующими при помощи синтезированной человеческой речи и алгоритмов ее распознавания. К аналогичным роботизированным системам можно отнести существующие на сегодняшний день и широко распространенные программы “Siri” и “Google Now”, представляющие из себя персональных цифровых ассистентов.

Обратная задача по описанию маршрута решается в основном двумя способами: изображение траектории пути на карте и последовательность действий к перемещению, требующая четкого выполнения. Оба варианта требуют наличия графического интерфейса и не могут быть восприняты человеком на слух. Также обладают рядом других недостатков, рассмотренных далее в тексте.

В данной работе рассмотрены алгоритмы построения и распознавания текстового описания маршрута в удобном для человека виде с учетом его когнитивных способностей и персональных знаний об окружающих объектах.

Ключевые слова: маршрут, карта, навигация, когнитивная навигация, описание маршрута.

Algorithms of construction and recognition of navigational route descriptions for computer mapping systems

V.A. Galaktionov, M.V. Pestun

Keldysh Institute of Applied Mathematics (Russian Academy of Sciences), Moscow, Russia

Computer navigation and mapping systems available at the moment, offer the possibility of finding optimal path from point A to point B. However, most of them do not allow you to enter the path that the user wants, and learn, for example, time for driving on specific route considering current traffic congestion. Those systems that allow it use graphical user interfaces which require to place anchor points of the route. Between them the system will build the path in an automatic mode. Such method is not suitable for mobile platforms because screen size is very limited and touch data entry by fingers has a low positioning accuracy. In addition, any graphical information input inapplicable with anthropomorphic robots and existing today and widespread programs like “Siri” and “Google Now”, which represent the personal digital assistants.

The inverse problem of the description of the route is commonly solved by two main ways: line of path on the map image and sequence of actions to move. Both options require a graphical user interface and can not be perceived by human ear. Also have a number of other disadvantages discussed hereinafter.

This paper describes the algorithms of construction and recognition of text description of route in human readable form in a view of its cognitive abilities and personal knowledge of the surrounding objects.

Keywords: route, map, navigation, cognitive navigation, route description.

Введение

Навигационно-картографические системы (такие как Яндекс Карты, Google Maps, автомобильные навигаторы и другие) решают следующие задачи:

- предоставление актуальных статических и динамических картографических данных;
- локализация положения пользователя;
- работа с маршрутами.

Задача по работе с маршрутами имеет следующие подзадачи:

- автоматический расчет оптимального маршрута между заданными точками начала и конца;
- построение описания маршрута (передача информации от системы к пользователю);
- распознавание описания маршрута (передача информации от пользователя к системе);
- отслеживание перемещения по заданному маршруту.

Интерес для данной работы представляют подзадачи построения и распознавания описания маршрута.

рута. Описание маршрута, построенное при помощи существующих на сегодняшний день систем, как, например, в автомобильном навигаторе, обладает слабой выразительностью и не имеет персонализации. Это осложняет его восприятие человеком. Одновременно, ввод конкретного маршрута человеком в компьютер ограничен существующими инструментами, обладающими малой функциональностью и строгими требованиями к интерфейсу пользователя. Таким образом, проблема существует в обоих направлениях передачи информации о маршруте: от компьютера к человеку и, наоборот, от человека к компьютеру. Помимо этого, современные картографические системы не подстраиваются под знания пользователя и не выделяют важную именно для него информацию, что уменьшает степень их удобства для повседневного использования.

Актуальность

Описание маршрута от точки А до Б в удобном для человека виде актуально в следующих задачах [1, 2, 3].

Навигация внутри зданий со сложной планировкой. В современном мире здания стали иметь очень большие размеры, постоянно расширяются и достраиваются. В связи с этим часто бывает осложнена навигация по ним. Система таблиц и указателей упрощает навигацию, но не всегда бывает достаточной: указатели на все интересующие человека места физически невозможно разместить повсеместно. На сегодняшний день наибольшие трудности вызывают следующие здания: торговые центры, аэропорты, многоэтажные парковки на тысячи автомобилей и другие крупные сооружения. Иногда встречаются здания с планировкой, при которой, например, на второй этаж можно попасть только с третьего, но не с первого этажа. Решением, упрощающим навигацию, является расстановка небольших терминалов, на которых пользователь может выбрать интересующий его объект (конкретный магазин, парковочное место, кабинет), и система расскажет ему, как до него добраться. Частично описание может быть текстовым, оформленным в виде прямой речи человека, объясняющего “как пройти”. Такой текст должен быть легко запоминаемым и однозначно понятным.

Навигация в городском окружении. Использование навигационных систем (например, автомобильного навигатора) частично решает проблему поиска нужного места в сложной системе дорог, однако их рекомендации не всегда бывают удобны для водителя, из-за чего ему требуется отвлекаться от дороги на монитор устройства с изображением карты и нарисованной поверх нее траектории движения. Инstrukция “поверните налево через 712

метров” может быть оформлена в более удобном виде: “поверните налево на втором светофоре”.

Указание компьютеру о следовании по специальному маршруту. На текущий момент навигационные системы для широкой аудитории не предлагают возможности легко и удобно задать конкретный маршрут, они могут лишь сами предлагать варианты для выбора. Однако порой необходимо узнать время в пути по конкретному маршруту. Для этого было бы удобно описать маршрут в приблизительном виде и попросить компьютер предложить наиболее подходящие под описание варианты. В случае необходимости можно уточнить маршрут. Существующие навигационные системы для решения этой задачи используют интерфейс, в котором пользователь может поставить на карте точки, через которые должен проходить маршрут. Однако такой ввод данных становится неактуальным в связи с развитием голосового ввода информации (“OK Google”, “Siri”), поэтому требуется новое универсальное решение.

Взаимодействие с роботизированными системами в области описания маршрута. На текущий момент идет стремительное развитие антропоморфных роботов, лишенных графических экранов и умеющих взаимодействовать с пользователями только при помощи голоса. Вопрос описания маршрута в словесной форме самим роботом и понимания того, какой путь имел в виду человек, становится актуальным.

Существующие решения

Было проведено широкое исследование существующих навигационно-картографических систем. Исследовались:

- специализированные автомобильные навигаторы: Navitel, Garmin, Mio;
- онлайн-сервисы: Яндекс Карты, Google Maps, Bing Maps;
- мобильные приложения: Яндекс Карты, Яндекс Навигатор, Google Maps, Apple Maps;
- носимые персональные устройства: Google Wear, Google Glass, Apple Watch;
- стационарные терминалы в крупных зданиях и туристических центрах.

Результаты исследования:

- существующие на сегодняшний день в навигационно-картографических системах методы построения описания маршрута не обладают одновременно удобством для восприятия и запоминания, универсальностью использования на разных устройствах (обязательно требуют экран) и подстройкой под знания конкретного пользователя, не предоставляя тем самым возможность осуществления когнитивной навигации;

— существующие на сегодняшний день в навигационно картографических системах методы распознавания описания маршрута не универсальны (не могут использоваться на разных платформах, требовательны к оборудованию пользователя) и не обладают персонализацией, тем самым не позволяют использовать в описании знакомые конкретному пользователю объекты и интервалы пути.

Проведенное исследование систем позволяет выделить два существующих метода для описания маршрута и два для распознавания:

- методы построения описания маршрута:
 - пошаговая инструкция (рис. 1);
 - изображение траектории (рис. 2);
- методы распознавания описания маршрута:
 - ввод опорных точек маршрута путем их установки мышкой на стационарном компьютере, между ними маршрут прокладывается автоматически (данный способ позволяет максимально точно задать траекторию, при этом по-прежнему учитывает правила дорожного движения и никогда не прокладывает маршрут по дороге с односторонним движением в обратном направлении, что, несомненно, является плюсом);
 - ввод точек транзита путем задания их конкретных координат или имени POI (менее удобный с точки зрения уточнения сложных маршрутов; часть автомобильных навигаторов, где преимущественно используется метод добавления транзитных точек, поддерживают их ограниченное количество и не позволяют полностью описать желаемый маршрут).

Предлагаемое решение

Предполагается, что универсальным форматом ввода и вывода маршрута является его текстовое словесное описание, исходя из фактов:

- текстовое представление может быть синтезировано в речь и использоваться в роботизированных системах, у которых отсутствует экран;
- текстовое представление может быть отображено на любом экране — черно-белом, цветном и даже только текстовом;
- текстовое представление легко человеку для ввода с учетом развития технологий распознавания речи и персональных ассистентов, таких как “Siri” и “Google Now”, работающих исключительно на голосовых командах;
- текстовое представление описания, имеющее вид прямой речи человека (т.е. не формальный строгий формат, а свободный разговорный стиль), является легким для восприятия и запоминания, так как использует привычные для

пользователя из повседневной жизни выражения и обороты.

Выводы об удобстве использования для человека текстового описания маршрута были получены на основе проведенных исследований совместно с факультетом психологии МГУ им. М.В. Ломоносова с использованием специальной системы построения виртуальной реальности CAVE [4, 5, 6].

рядом с 2-я Хуторская улица 22, Москва, Россия, 127287		A-B: 32,6 км 42 мин
1	Выезжайте по 2-я Хуторская улица в направлении Башкиловская улица	36 м
2	Поверните налево на Башкиловская улица, а затем резко поверните налево на 2-я Хуторская улица	0,3 км
3	Поверните налево на улица Костякова	0,3 км
4	Поверните направо на Дмитровский проезд	0,3 км
5	Название дороги изменяется на улица Руставели	0,1 км
6	Воспользуйтесь съездом направо на улица Бутырская улица	0,3 км
7	Название дороги изменяется на Дмитровское шоссе	11,8 км 13 мин
8	Продолжайте движение прямо по А104 / Дмитровское шоссе	9,6 км
9	Поверните налево на Рогачевское шоссе / R113	4,6 км
10	Название дороги изменяется на Батаревная улица	1,2 км
11	Название дороги изменяется на улица Чайковского	1,0 км
12	Поверните налево на Краснопольская улица	2,1 км
13	Поверните налево на Текстильная улица	0,3 км
14	Поверните направо на улица Булычьева	0,4 км
15	Поверните налево на Аэропортовская улица	0,1 км
16	Поверните налево на дороге	56 м
B		Выезжайте в пункт рядом с Лобня, Россия, 141730 Последний перекресток — Аэропортовская улица

Рис. 1: Пример описания маршрута в виде пошаговой инструкции (Bing Maps).

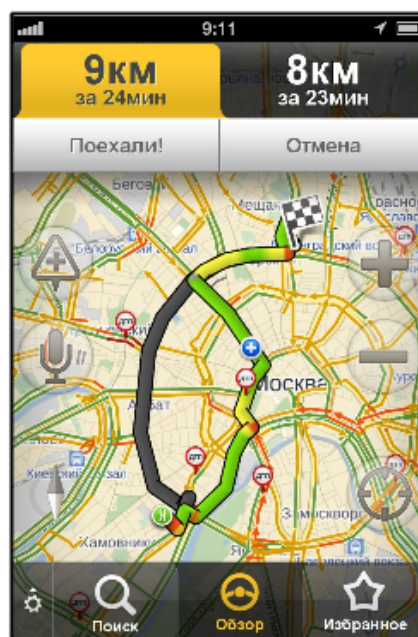


Рис. 2: Пример описания маршрута в виде изображения траектории (Яндекс Навигатор).

Алгоритм построения описания

Высокоуровневое представление шагов работы алгоритма построения текстового описания маршрута выглядит следующим образом [1, 4]:

1. выделение из полигона маршрута, (внутреннего представления пути в виде ломаной линии с географическими координатами в вершинах), полученного от навигационно-картографической системы, знакомых пользователю интервалов (тех маршрутов, по которым он часто перемещается и может идентифицировать по заданному названию);
2. фиксация близлежащих к маршруту знакомых пользователю POI (от англ. Point Of Interest – точка интереса) в качестве ориентиров (известность POI определяется из частоты его посещения пользователем и информации из социальных сетей);
3. пировка интервалов по однотипным навигационным директивам (навигационные директивы – это команды к перемещению вида “повернуть налево”, “двигаться прямо”, “развернуться”);
4. формирование каркаса (рис. 3) результата из лексем (под каркасом понимается представление итогового текстового описания в абстрагированном от естественного языка виде);
5. склонение названий и интеграция их в каркас;
6. замена оставшихся лексем по словарю;
7. дозагрузка изображений POI (получение от навигационно-картографической системы снимков объектов конкретно с той позиции, которой их увидит пользователь в процессе перемещения по маршруту);
8. формирование результирующего текстового описания в формате HTML с изображениями и визуальным форматированием (рис. 5).

```
{ACTION} · {DIRECTION} · {DISTANCE}, · {JUNK} · {TURN_LEFT}, · {JUNK} · {ACTION} ·
{GROUP_DIRECTION} · {GROUP_DISTANCE} · {TILL} · {POI_NAME}, · {JUNK} ·
{KNOWN_ACTION} · {KNOWN_NAME}, · {JUNK} · {ACTION} · {GROUP_DIRECTION} ·
{GROUP_DISTANCE} · {TILL} · {POI_NAME}. ¶
```

Рис. 3: Пример построенного каркаса (промежуточного представления) будущего текстового описания маршрута.

Приведенный выше шаг 4 (формирование каркаса результата из лексем) является одним из самых важных в алгоритме. На этом этапе строится последовательность лексем, которая в итоге будет преобразована в текст (рис. 4). При построении каркаса (в основе алгоритма лежит формальная грамматика – способ описания формального языка) используются следующие формы, представимые в виде одной или нескольких лексем:

- полные и упрощенные названия POI, используемые в прямой речи;

- словесное описание внешнего вида POI;
- склонения названий POI (POI_NAME);
- знакомые пользователю интервалы пути (KNOWN_NAME);
- директивы к перемещению (ACTION, DIRECTION);
- упрощенные округленные расстояния (GROUP_DISTANCE);
- группы однотипных действий (GROUP_DIRECTION);
- предлоги, союзы, знаки препинания (TILL, “;”, “.”);
- “мусорные” слова, придающие тексту “человечности” (JUNK).



Рис. 4: Составные части текстового описания пути, используемые алгоритмом.

Большую сложность составляет подзадача придания тексту “человечности”. Для ее решения был разработан следующий механизм формирования каркаса лексем, основанный на случайных величинах и частотах использования слов и оборотов. Все описание разбивается на блоки $PB = \{ACTION_TYPE, DIRECTION, RELATIVITY_NEAR, POI_NAME_NEAR, DISTANCE, RELATIVITY_TILL, POI_NAME_TILL\}$ (важно сразу отметить, что не все поля пятерок PB будут использованы в итоговом сформированном текстовом описании):

- ACTION_TYPE – вид действия, это может быть “проходите”, “проезжайте”, “двигайтесь”, “поверните”, “развернитесь” и так далее;
- DIRECTION – направление движения, это может быть “прямо”, “мимо”, “налево”, “направо” и так далее;
- RELATIVITY_NEAR – относительность действия к POI, это может быть “у”, “мимо”, “рядом” и так далее;
- POI_NAME_NEAR – собственно название POI, рядом с которым происходит действие, связанное с RELATIVITY_NEAR;
- DISTANCE – расстояние, которое необходимо преодолеть, это может быть “полкилометра”, “четверть километра”, “пара километров”, “сразу” и так далее;

- RELATIVITY_TILL – относительность действия к POI, это может быть “до”, “вплоть” и так далее;
- POI_NAME_TILL – собственно название POI, рядом с которым происходит действие, связанное с RELATIVITY_TILL.

Далее алгоритм разбивает РВ на предложения, для этого используются следующие правила:

- в одном предложении может быть минимум один и максимум три блока РВ:
 - один блок РВ может быть только в конечном предложении и только в том случае, если к нему не может быть добавлен другой блок РВ;
 - в остальных случаях используется два или три блока РВ;
 - вероятность появления трех блоков РВ определяется значением наперед заданной константы Ψ_1 .
- в случаях, когда два последовательных блока РВ содержат одинаковые значения ACTION_TYPE, они могут быть сгруппированы в один, вероятность определяется наперед заданной константой Ψ_2 .

Предложения, в которых повторяется одно и то же слово и словесный оборот, воспринимаются человеком как некорректные, поэтому важно избавиться от подобных ситуаций. Для этого заведены следующие константы, упомянутые выше:

- Ψ_1 – вероятность предложения из трех блоков РВ;
- Ψ_2 – вероятность группировки двух блоков РВ в один;

Мусорные слова используются для придания тексту большей “человечности”. Под мусорными словами понимаются наречия, используемые людьми при описании пути, например, “затем”, “после чего”, “далее” и другие. Для каждой лексемы определена таблица с возможными вероятностями ее использования. Помимо этого, четко задано правило, что подряд не может быть использовано две одинаковые словесные формы лексемы одного типа. В случаях, когда это неразрешимо, например, при необходимости указать одинаковое расстояние на последовательных интервалах пути, используются лексемы заменители. Для каждого типа лексем определены свои лексемы заменители, где это возможно:

- для ACTION_TYPE: “снова”, “и опять”, “еще раз” и другие;
- для DIRECTION: “в ту же сторону” и другие;
- для DISTANCE: “столько же”, “такое же расстояние” и другие.

Алгоритм распознавания текстового описания маршрута представляет собой обратную последо-

вательность действий с несколькими обобщениями, позволяющими сократить объем вводимой пользователем информации.



Рис. 5: Пример перевода изображения траектории в текстовое представление с использованием знаковых пользователю маршрутов и POI.

Практическое применение

Разработанный на основе предложенных алгоритмов программный комплекс уже используется:

- в созданной автором навигационной системе Интранета высотного офисного здания SkyLight (Ленинградский проспект, д. 39) — любой сотрудник, работающий в данном здании, имеет возможность получить подробную инструкцию, как добраться до рабочего места интересующего его человека, просмотрев информацию о нем в персональном профиле на корпоративном портале; обычные методы навигации (указатели, надписи) работали плохо ввиду сложного зеркального по четырем направлениям расположения рабочих мест (используется методика рассадки open space);
- в картографической системе Карты Mail.Ru для предоставления описания маршрута пользователю в удобном персонализированном виде;
- в летающей автономной роботизированной системе (квадрокоптер), способной обходить препятствия на пути согласно заданной инструкции, описывающей маршрут передвижения в близком к естественному языку виде (использовалась заранее подготовленная карта местности с размеченными объектами). Робот был разработан, собран и запрограммирован командой (автор и еще три участника) для конкурса летающих роботов от компании КРОК;
- в автономном комнатном роботе на гусеницах, способном проезжать по описанной пользователем траектории (в разработке) [1].

Заключение

В статье были рассмотрены алгоритмы, позволяющие строить и распознавать текстовое описание

маршрута, обладающего персонализацией, близостью к прямой речи человека и лаконичностью, стремящиеся максимально задействовать когнитивные процессы пользователя с целью упрощения восприятия информации. Благодаря текстовому формату представления они могут быть реализованы и использованы на максимально широком списке различных платформ: начиная от мобильных и стационарных компьютеров и заканчивая антропоморфными роботами. Текст всегда может быть синтезирован в голосовую речь, распечатан или выведен на экран. Появление персональных ассистентов, таких как “Siri” и “Google Now”, подталкивает к использованию именно голосового взаимодействия с устройствами.

Литература

- [1] *О.И. Давыдов, А.К. Платонов* Сеть Пассфреймов - комбинированная модель операционной среды мобильного робота // Препринт ИПМ № 15. – М.: ИПМ им. М.В. Келдыша РАН, 2015.
- [2] *Loomis J.M., Klatzky R.L., and Colledge R.G.* Human navigation by path integration // *Wayfinding: Cognitive mapping and spatial behavior*, 1999.
- [3] *М.В. Пестун* Когнитивная навигация и алгоритм построения текстового описания маршрута в удобном для человека виде // Программные продукты и системы. – Тверь: “Центрпрограммсистем”, 2015. С. 28-33.
- [4] *Lakhtionova I., Menshikova G.* The method of testing the ability of allocentric cognitive maps acquisition // *Proc. of 36-th European Conference on Visual Perception, Bremen, Germany, Aug. 25-29, 2013, Perception, v.42, ECVP Abstract supplement*, P. 53.
- [5] *Lakhtionova I., Menshikova G.* Testing the ability of allocentric cognitive map acquisition using the CAVE technique // *Cognitive Modeling: Collection of Papers of the First International Forum on Cognitive Modeling (14-21 September, 2013, Italy, Milano-Marittima)*.
- [6] *Menshikova G., Bayakovski Yu., Luniakova E., Pestun M., Zakharkin D.* Virtual Reality Technology for the Visual Perception Study // *Springer, Transactions on Computational Science XIX, Lecture Notes in Computer Science Volume 7870, 2013, P. 107-116.*

Повышение эффективности технологических процессов обработки и анализа графических данных в ГИС*

Ю.Г. Васин, Т.Ш. Утешева

ya.vasinyuri@yandex.ru | uts13@yandex.ru

Нижегородский Государственный Университет имени Н.И. Лобачевского

Рассматриваются эффективные методы вычислительной геометрии, расширяющие функциональные возможности автоматизированной составляющей технологии создания цифровых карт и функционирования геоинформационных систем при массовой обработке больших объемов графических данных.

Ключевые слова: вычислительная геометрия, геоинформационные системы, обработка графической информации

Improving the Efficiency of Technological Processes of Processing and Analysis of Image Data into GIS*

Yu.G. Vasin, T.Sh. Utesheva

Nizhny Novgorod State University named after N.I. Lobachevskiy

We consider effective methods of computational geometry, extends the functionality of the automated part of the technology of digital mapping and geographic information systems functioning in the mass processing of large volumes of image data.

Keywords: computational geometry, geographic information systems, processing of graphical information

1. Введение

Временная и емкостная эффективность массовой обработки больших объемов графических данных в значительной мере определяется используемыми методами вычислительной геометрии, лежащими в основе геометрических и логических операций над графическими объектами. Схема функционирования современных ГИС имеет несколько наиболее ресурсоемких блоков, временные затраты которых определяются именно вычислительной сложностью реализуемых ими алгоритмов. В большинстве таких «узких» мест производится анализ геометрии взаимного расположения большого числа графических объектов и потому применение методов вычислительной геометрии очень эффективно. Практический опыт разработки методов, алгоритмов и программ обработки комплексной (метрической и атрибутивной) картографической информации, как на этапе создания цифровых карт, так и на этапах ее анализа в различных геоинформационных тематических приложениях, показал, что для данного вида информации наиболее эффективен метод от общего к частному на базе иерархического представления данных. Кроме того, на начальных этапах создания цифровых карт и в приложениях, ориентированных на задачи принятия решений на базе цифровых планов городов, когда высок процент объектов малой метрической протяженности, эффективен метод охватывающих прямоугольников.

Эффективное иерархическое представление метрики картографической векторной информации (планарной и трехмерной) [1-5], метод от общего к частному решения геометрических задач [6,7] на основе такого структурированного представления, различные варианты повышения эффективности метода от общего к частному [8,9] и алгоритмы реализации различных технологических этапов ГИС были разработаны и апробированы на уровне промышленного внедрения в НИИ ПМК ННГУ.

2. Метод от общего к частному

В работах [1-6] были предложены методы и алгоритмы обработки графической информации на базе локальных однородных хорошо приспособленных базисных функций (ЛОХПБФ) и предопределяемых ими иерархических регулярных структур представления данных в виде усеченных разностных бинарных деревьев. Отсчеты каждого уровня иерархии указанных структур отбираются с использованием ЛОХПБФ и представляют исходные данные с некоторой контролируемой погрешностью. Чем ниже рассматриваемый уровень иерархии представления данных, тем точнее описывается исходное поле. В совокупности иерархическая структура содержит не избыточный набор существенных отсчетов и ошибок описания на предыдущих уровнях иерархии, позволяющий провести восстановление исходной информации с заданной точностью. Указанная структура может быть представлена в виде усеченных бинарных разностных деревьев для различного типа графических данных.

Работа выполнена и опубликована при финансовой поддержке РФФИ, гранты 13-07-00521, 13-07-12211, 15-07-20347

Иерархическое представление графической информации позволяет эффективно использовать метод от общего к частному в задачах анализа взаимного расположения картографических объектов. Геометрические задачи решаются путем последовательных приближений: поиск решения начинается с корня иерархии и завершается при возможно более грубом представлении данных. Информация уточняется там, где это необходимо, то есть на участках, определяющих решение задачи, используется более точное описание данных. При этом на верхних уровнях дерева формируются промежуточные решения, которые, во-первых, могут использоваться как некоторое приближение к искомому решению, во-вторых, они применяются для оценок перспективных направлений дальнейшего поиска окончательного решения.

При реализации такого подхода важен корректный выбор критериев отбора перспективных участков аппроксимации, уточнение которых заведомо приведет к искомому решению. Синтез таких критериев при решении широкого класса задач требует в каждой вершине дерева на каждом уровне иметь ограничивающие оценки, такие, что исходная информация, соответствующая данным существенным отсчетам не может отклониться за их пределы. Было предложено при решении задач анализа графических данных использовать усеченные бинарные разностные деревья с D -окрестностями, где D -окрестность уровня k – D_i^k это область, за пределы которой исходная информация не может отклониться от i -го аппроксимирующего участка k -го уровня. Это позволило для каждой геометрической задачи сформулировать критерий отбора только тех участков рассматриваемых графических объектов, которые могут содержать решение, а заведомо бесперспективные (и только их) исключать из рассмотрения. Такой подход оказался высокоэффективным при массовой обработке больших объемов графических данных и был применен при разработке алгоритмов комплекса программ решения задач вычислительной геометрии, связанных с анализом взаимного положения объектов на плоскости. Программы комплекса оформлены в виде функций и организованы в библиотеку, которая может быть использована для создания любых приложений, требующих быстрой обработки больших объемов графической информации.

Библиотека содержит функции для решения следующих задач:

Для определения взаимного положения точек и кривых: определение расстояния от точки до кривой; поиск кривой из множества, ближайшей к заданной точке; поиск точки из множества, ближайшей к заданной кривой.

Для определения взаимного положения точек и областей: определение положения точки относительно области; определение положения точек множества относительно области; определение точек из множества, лежащих в ϵ -окрестности области вне ее; определения положения части заданных точек из некоторого множества с отнесением точек, лежащих на границе, в соответствии с заданным режимом.

Для определения взаимного положения кривых: определение расстояния между двумя кривыми; определение точек пересечения и участков совпадения для множества кривых с подмножеством и для двух множеств кривых; поиск близких кривых в заданном множестве.

Для определения взаимного положения кривых и областей: выделение части кривой, лежащей внутри многоугольника; выделение части кривой, лежащей внутри области, граница которой задана структурированной кривой.

Для определения взаимного положения областей: определение положения множества контуров относительно многосвязной области; построение пересечения заданной области с множеством областей.

Это – базовые функции, суперпозиция которых позволяет решать широкий класс задач определения геометрических и логических отношений между графическими объектами.

Разработанные алгоритмы обладают следующими важными достоинствами: высокая вычислительная производительность; общий методологический подход к решению всех геометрических задач; использование единой предварительной обработки данных, которая проводится однократно и используется для решения всех необходимых задач, являясь при этом частью общей технологии обработки графической информации.

3. Обработка и анализ графических данных в ГИС

Результатом первого этапа оцифровки картографических данных является метрическая составляющая объектов карты с простейшим семантическим наполнением. Последующие этапы предполагают формирование более сложных характеристик, пространственных семантических и логических связей, а также создание новых площадных и линейных объектов из автоматически введенных дискретных и линейных. Этот процесс является интерактивным и уровень его автоматизации во многом зависит от качества исходного материала и успешности всех предварительных процедур. Тем не менее, ресурсы повышения эффективности автоматизированной составляющей в настоящее время не исчерпаны, и с этой целью были разработаны следующие процедуры:

- сшивки смежных картографических листов с учетом всех атрибутивных данных, а так же пространственно-логических и топологических связей;
- программный комплекс построения цепочно – узловой (сегментной) модели описания метрической информации;
- процедура формирования информативной семантической (атрибутивной) информации исходя из анализа взаимного расположения картографических объектов;
- автоматического формирования новых объектов – производных от нескольких ранее существующих;
- семантического структурирования объектов посредством механизма прерываний и характеристик на основе анализа геометрического отношения объектов;
- процедура распознавания связанных групп элементов заполнения площадных объектов карты;
- сборки площадных и линейных объектов из набора дискретных;
- идентификации кодами автоматически введенных объектов по различным критериям: примыкания, направления оцифровки, одновременно примыкания нескольких объектов с различными кодами, отношения вне – внутри и др.
- процедура формирования новых площадных и линейных объектов из автоматически введенных линейных объектов по условию выполнения большого набора геометрических отношений.

Задача всех перечисленные процедур состоит в автоматическом формировании новых объектов, присвоении новых кодов, формировании характеристик (информативной семантической информации), прерываний или пространственно – топологических связей в зависимости от наличия и типа геометрического взаимодействия с другими картографическими объектами. При этом возможен анализ логики взаимодействия (конъюнкция, дизъюнкция и др.) рассматриваемого объекта в одной точке с несколькими различными типами картографических объектов. Решаемые задачи являются ресурсоемкими, причем основную временную нагрузку алгоритмов определяет блок геометрических и логических операций над графическими единицами карты.

Программная реализация этих процедур предусматривает универсальность по отношению к наполнению информационной базы (классификатора) и возможность использования при создании электронных топографических карт, топопланов, а так же морских навигационных карт всего масштабного ряда.

4. ЗАКЛЮЧЕНИЕ

Представленные выше программные комплексы были интегрированы в состав созданной в НИИ ПМК ННГУ объектно-ориентированной интеллектуальной геоинформационной системы «ГИС-Терра».

Комплекс реализован на языке программирования С++. В качестве инструментальных средств для создания и отладки ПО использовалась интегрированная среда разработки Borland С++ Builder.

Эксплуатация в реальных производственных условиях программных комплексов и систем в различных организациях и предприятиях Российской Федерации подтвердила эффективность и результативность принятых решений при разработке методов, алгоритмов и создании программного обеспечения.

Литература

- [1] Васин Ю.Г. Нерегулярные выборки отсчетов исходной информации и задача кодирования электрокардиографических данных // Кибернетика и вычислит. техника, 1978. Вып.42. Киев, 1978. С.98–104.
- [2] Васин Ю.Г. “Хорошо приспособленные” базисы и задачи обработки экспериментальной информации: Учебное пособие // Горьк. гос. ун-т, 1979. С.1-129.
- [3] Васин Ю.Г. Оптимизация описания исходных данных в диалоговых системах решения задач классификации // Современное состояние теории исследования операций. М.: Наука, 1979. С.424-446.
- [4] Васин Ю.Г. “Хорошо приспособленные” локальные однородные методы обработки графической информации // Автоматическая обработка сложной графической информации: Межвуз. тематич. сб. науч. тр. Горьк. гос. ун-т, 1984. С.131-158.
- [5] Васин Ю.Г. Эффективность различных стратегий обработки видеoinформации на базе локальных однородных рекуррентно - рекурсивных функций // Методы и средства обработки графической информации : Межвуз. сб. науч. тр. Горьк. гос. ун-т - Горький, 1986. С.4-45.
- [6] Васин Ю.Г., А.Д. Крахнов Метод от общего к частному в задачах дискретной геометрии // Методы и средства обработки графической информации: Межвуз. сб. Горьк. гос. ун-т., 1986. С.67-80.
- [7] Васин Ю.Г., А.Д. Крахнов, Т.Ш. Утешева Метод от общего к частному при решении пространственных задач дискретной геометрии // Автоматическая обработка сложной графической информации: Межвуз. сборник. Горьк. гос. ун-т, 1988. С.73-83.
- [8] Васин Ю.Г., А.Д. Крахнов, Т.Ш. Утешева Методы дискретной геометрии в задачах обработки сложной графической информации // 7 Международная конференция по распознаванию образов и анализу изображений: Новые информационные тех-

нологии PRIA-7-2004. Труды конференции, СПб. 2004. Т.3. С.954-956.

- [9] Vasin Yu.G., Utesheva T.Sh. Raising the efficiency of technological processes of digitizing and preparation of graphical information for 3d modeling and rendering // The 11-th International Conference "Pattern recognition and image analysis: new information technologies"(PRIA-11-2013). September 23-28, 2013. Conference proceedings, Samara: IPSI RAS, 2013. Vol. II. pp.763-765.

Об авторах

Юрий Васин – профессор ННГУ им. Н.И. Лобачевского.

Его адрес: ya.vasinyuri@yandex.ru.

Тамара Утешева – доцент ННГУ им. Н.И. Лобачевского.

Ее адрес: uts13@yandex.ru.

Итерационный метод восстановления спектра по цвету RGB

В.В. Афанасьев¹, А.В. Игнатенко², А.Г. Волобой¹

vafanasjev@graphics.cs.msu.ru | ignatenko@graphics.cs.msu.ru | voloboy@gin.keldysh.ru

¹Институт прикладной математики им. М.В.Келдыша, Москва, Россия

²Факультет вычислительной математики и кибернетики

Московский государственный университет им.М.В.Ломоносова

Во многих задачах синтеза изображений необходимо проводить расчёты с учётом спектра светового излучения, а не только его цвета. Это позволяет корректно моделировать такие явления как дисперсия, спектральное поглощение. К таким задачам относится, например, фотореалистичная визуализация драгоценных камней. Однако в большинстве случаев получение исходных данных в спектральном представлении затруднено или невозможно. Как правило, цифровые камеры записывают цвет в виде трёх чисел, соответствующих отклику чувствительных элементов. Данная статья описывает итерационный метод получения спектра из цвета RGB, который позволяет строить результат с учётом заданных ограничений.

Ключевые слова: восстановление спектра, синтез изображений, фотореалистичный рендеринг

Iterative spectrum reconstruction using RGB color

V. V. Afanasyev¹, A. V. Ignatenko², A. G. Voloboy¹

¹The Keldysh Institute of Applied Mathematics RAS

¹Faculty of Computational Mathematics and Cybernetics

Lomonosov Moscow State University, Moscow, Russia

Many image synthesis problems require spectrum-aware calculations instead of simple color calculations. It allows correct modelling of wavelength-dependent phenomena such as dispersion or spectral absorption. For example, photorealistic rendering of gemstones refers to such problems. In most cases initial spectral data acquisition is hard or impossible. Usually digital cameras store the color represented by 3 numbers corresponding to sensitive elements' response. This article describes an iterative method of spectrum reconstruction using RGB color that allows getting the result matching given constrains.

Keywords: spectrum reconstruction, image synthesis, photorealistic rendering

Введение

Свет представляет собой электромагнитную волну, одной из характеристик которой является спектр. Человеческий глаз имеет 3 типа цветных рецепторов, которые воспринимают свой диапазон длин волн из видимого спектра. В ходе экспериментов CIE были получены усреднённые кривые спектральной чувствительности человеческого глаза. Отклики чувствительных клеток глаза задают одно из линейных цветовых пространств. Преобразования между линейными цветовыми пространствами описаны во многих источниках, например, [4]. Здесь и далее будем работать с кривыми для условного линейного цветового пространства RGB.

Спектральные кривые для пространства CIE RGB приведены на рис. 1.

Для регистрации световых волн используются камеры, которые имеют несколько видов чувствительных элементов (обычно тоже 3), каждый со своей кривой отклика. Как правило, они соответствуют красному, зелёному и синему цветам: $r(\lambda)$, $g(\lambda)$, $b(\lambda)$.

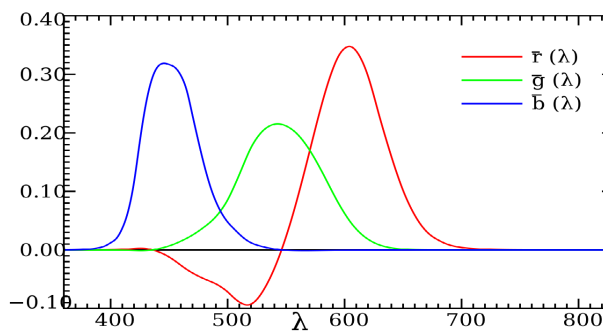


Рис. 1: Спектральные кривые CIE RGB.

Отклик камеры на спектр $S(\lambda)$ выражается следующим образом:

$$\begin{aligned} R &= \int_{380}^{780} S(\lambda)r(\lambda)d\lambda \\ G &= \int_{380}^{780} S(\lambda)g(\lambda)d\lambda \\ B &= \int_{380}^{780} S(\lambda)b(\lambda)d\lambda \end{aligned} \quad (1)$$

Устройства отображения цвета, такие как мониторы, имеют набор излучателей, как правило, 3 типов, каждый со своим спектром излучения. Цвет пикселя, представленный тремя числами RGB , задаёт мощность этих источников излучения, для наблюдателя спектры излучателей одного пикселя складываются, и его глаз воспринимает итоговый спектр в соответствии со своими кривыми отклика. Выше была описана общая схема записи и воспроизведения цвета. В случае синтеза изображений отсутствует этап записи цвета реального объекта, вместо этого его цвет рассчитывается на основе некоторых данных. В большей части задач можно пренебречь спектральной природой света и проводить расчёты с цветами RGB . Однако для определённого класса задач спектр важен: существуют такие явления как дисперсия, спектральное поглощение и отражение, учёт которых существенно влияет на результат. К таким задачам относится синтез изображений драгоценных камней, где дисперсия света в материале определяет итоговый цвет, который видит наблюдатель.

Для синтеза изображений с учётом спектра необходимо иметь спектральные исходные данные. В частности, панораму окружения бриллианта, или спектральные текстуры в других задачах. Однако со стандартного оборудования получить исходные данные можно в цвете RGB , и напрямую использовать их для расчётов в спектре нельзя. Нужно по имеющемуся цвету воспроизвести максимально приближенно исходный спектр. Из прямого преобразования спектра в RGB очевидно, что существует бесконечно много спектров, дающих одинаковый набор чисел RGB . Такие спектры называются метамерами. С математической точки зрения, подходит любая функция, дающая после преобразования (1) заданные числа RGB , однако с точки зрения физики, бессмысленны спектры с отрицательными значениями. По мере наличия априорной информации об исходном спектре, на результат могут быть наложены дополнительные ограничения, например, непрерывность или гладкость.

Обзор существующих методов

Существующие решения основаны на представлении спектра в виде кусочно-постоянной функции, дискретизованной с определённым шагом по длине волны.

В статье [1] спектр реконструируется как линейная комбинация трёх базисных функций, соответствующих излучающим фосфорам, с коэффициентами, вычисленными из исходного цвета RGB путём решения системы линейных уравнений.

В [2] предлагается взять такую же линейную комбинацию от заданных автором функций: константной, синусоидальной и косинусоидальной, с коэффициентами, полученными аналогичным способом.

Данные два подхода позволяют получить спектры, которые после преобразования имеют в точности исходный цвет RGB при заданных кривых отклика. Проблема этих подходов состоит в том, что они не учитывают реальных физических свойств результирующего спектра, также результатом может быть функция, имеющая отрицательные значения. Автор работы [3] ввёл дополнительные ограничения на гладкость спектра для того, чтобы результат лучше соответствовал спектрам отражения известных материалов. На основе этого ограничения был построен набор базисных функций, и итоговый спектр является некоторой их линейной комбинацией.

Проблемой последнего метода является невозможность построить спектр для некоторых цветов.

Предложенный подход

Предлагается итерационный метод восстановления спектра. На вход алгоритма подаётся:

- Набор коэффициентов линейного цветового пространства, например, 3 числа RGB .
- Набор кривых отклика $r(\lambda)$, $g(\lambda)$, $b(\lambda)$, на основе которых из спектра можно получить цвет.
- Ограничения на результат, представленные в виде процедуры коррекции “неправильного” спектра в сторону “правильного” $Constrain(S(\lambda))$. Например, процедура, приравнивающая нулю все отрицательные значения.
- Точность приближения цвета ε .

Результатом применения алгоритма является спектр, цвет которого приближает заданный с указанной точностью ε . Спектр и кривые отклика могут быть заданы множеством дискретных способов или аналитически.

Начальным приближением S_0 является нулевой спектр. Далее последовательно циклически выбираются компоненты цвета R, G, B, R, \dots и производятся следующие действия (пример для компоненты R):

1. Расчёт текущего значения R_i .
2. Вычисление разности с соответствующим эталоном $diff = R_{photo} - R_i$.
3. Коррекция существующего спектра S_{i-1} на функцию $k * BaseR(\lambda)$. Коэффициент k вычисляется таким образом, чтобы $diff \rightarrow 0$.
4. Применение коррекции $Constrain(S_i(\lambda))$ для приведения спектра в соответствие с ограничениями.

Процесс повторяется до тех пор, пока не будет достигнута заданная точность по цвету: $\|RGB_{photo} - RGB_i\| < \varepsilon$. Базовые функции $BaseR(\lambda)$, $BaseG(\lambda)$, $BaseB(\lambda)$ можно выбрать равными соответствующим кривым отклика камеры или другими, исходя из некоторой информации о спектре. В простейшем варианте, без ограничений на

спектр, алгоритм сойдётся к решению из работы [1] с данными базовыми функциями, т.к. является покомпонентным спуском. Обратим внимание, что в общем случае, при наличии ограничений полученное решение не будет являться линейной комбинацией базовых функций. Также введение ограничений может повлиять на сходимость. Сходимость для всевозможных ограничений на спектр не исследовалась.

Результаты

Предложенный алгоритм был применён для перевода панорам из *RGB* в спектральное представление в задаче синтеза изображений бриллиантов в этой панораме. Алгоритм применяется к изображению попиксельно. Алгоритм реализован для кусочно-постоянных спектров в среде Matlab и опробован на реальных данных. Использовано ограничение неотрицательности спектра. Опытным путём не была выявлена расходимость алгоритма на каких-либо данных при заданных условиях.

Ниже приведён пример разложения тестового цветного изображения по длинам волн с шагом 5 нм. На рис. 2 показано исходное изображение и монохроматические составляющие для некоторых длин волн.

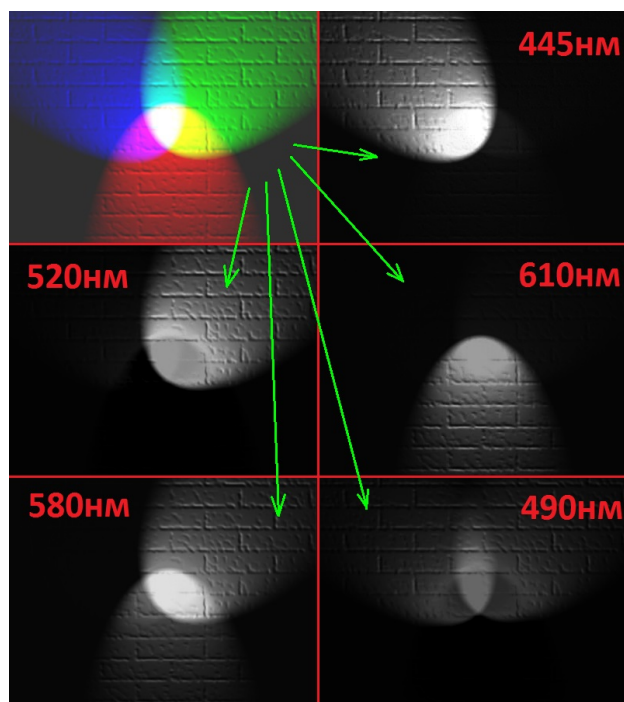


Рис. 2: Пример восстановления спектра.

Метод описан и применялся для трихроматических данных, однако использование 3 цветов не является его принципиальным ограничением. Также возможна произвольная форма задания использу-

емых функций при условии соответствующих изменений в процедурах суммирования, интегрирования и коррекции.

Пример визуализации бриллианта в работе [5] с использованием спектральной панорамы окружения, полученной из *RGB* панорамы описанным методом, приведен на рис. 3.



Рис. 3: Визуализация бриллианта в спектральной панораме.

Выводы

Разработан метод восстановления спектра излучения по цвету, позволяющий итерационно строить итоговый спектр с учётом заданных ограничений. Данный метод был проверен в практических приложениях. В то же время, остаётся открытым вопрос сходимости для различных ограничений, накладываемых на спектр.

Литература

- [1] *Glassner A.S.* How to derive a spectrum from an RGB triplet // *IEEE Computer Graphics and Applications*, July 1989 – Vol.9, No.4. pp.95–99.
- [2] *Glassner A.S.* *Principles of Digital Image Synthesis.* – San Francisco: Morgan-Kaufman, 1995. – pp.104-105.
- [3] *Smits B.* An RGB-to-spectrum conversion for reflectances // *Graphics tools: The jgt editors' choice*, 2005 – p.291.
- [4] *Rogers D.F.* *Procedural Elements for Computer Graphics.* – McGraw-Hill, 1985.
- [5] *Лобанов И.В.* Спектральные представления и аппаратно-ускоренная трассировка лучей в задаче визуализации драгоценных камней // Тезисы конференции «Ломоносов — 2014». – Москва, МАКС Пресс, 2014.

Анализ совместной работы методов сопровождения объектов в видеопотоке, получаемом с летательного аппарата*

В.А. Павлов, С.В. Завьялов, С.В. Волвенко

pavit@bk.ru | zavyalov_sv@cee.spbstu.ru | volk@cee.spbstu.ru

Санкт-Петербургский государственный политехнический университет Петра Великого

Рассмотрены варианты решения задачи сопровождения объекта в видеопотоке, получаемом с камеры беспилотного летательного аппарата. Проведён анализ современных алгоритмов сопровождения объектов и принципов их работы. Рассмотрена совместная реализация алгоритмов сопровождения.

Ключевые слова: TLD, CMT, Struck, сопровождение объекта, летательный аппарат

Analysis of collaboration methods for tracking objects in a video stream received from an aircraft*

V.A. Pavlov, S.V. Zavialov, S.V. Volvenko

Peter the Great St.Petersburg Polytechnic University, St.Petersburg, Russia

The variants of solving the problem of tracking an object in the video stream received from the camera of the unmanned aerial vehicle are considered. The analysis of modern algorithms for tracking objects performed. Joint implementation of the tracking algorithm is considered.

Keywords: TLD, CMT, Struck, object's tracking, aerial vehicle

1. Введение

Одной из наиболее сложных и актуальных задач машинного зрения является проблема обнаружения и сопровождения движущихся объектов в видеопоследовательности при наличии различного вида помех [1]. Представляет интерес два основных случая движения [2]: неподвижная камера и постоянный фон, подвижная камера и изменяющийся фон. В первом случае движение объектов характеризуется изменением яркости пикселей в определённых областях изображения, поскольку фон не изменяется. Второй случай является более сложным, из-за обобщённого движения объекта и камеры.

2. Проблемы, возникающие при работе с видеопотоком, получаемым с летательного аппарата

В данной статье рассматривается расположение камеры на борту летательного аппарата, соответствующее второму случаю движения. Работа с видеопотоком, получаемым с летательного аппарата, накладывает специфические требования на алгоритмы сопровождения, из-за следующих особенностей видеопоследовательностей: потеря одного или нескольких кадров; частично или полностью повреждённые кадры; артефакты сжатия с потерями; резкие рывки камеры; выход объекта за пределы кадра и возврат в кадр через несколько секунд; размытие (motion blur) всего кадра при резком движении камеры из-за низкой скорости затвора; размер сопровождаемого объекта может меняться от минимального (около 16x16 пикселей) до максималь-

ного (704x576), что соответствует размеру кадра; движение камеры характеризуется частым сдвигом, приближением/удалением в очень широких пределах, вращением камеры; наличие данных телеметрии поверх изображения. Также следует отметить, что часто возникает ситуация, когда в кадре присутствуют нескольких одновременно движущихся объектов с близкими характерными признаками и пересекающимися траекториями.

Целью настоящего исследования является анализ, выбор и тестирование совместной работы алгоритмов сопровождения за объектами.

3. Описание выбранных алгоритмов

В ходе анализа публикаций, освещающих решения поставленной задачи, были выбраны следующие подходы: TLD [3], CMT [4], Struck [5]. Эти подходы частично удовлетворяют следующим критериям: устойчивая работа при полном или частичном исчезновении объекта из поля зрения камеры; требуемая производительность; точность сопровождения. Далее рассмотрим более подробно выбранные алгоритмы.

3.1 Принцип работы TLD

Подход, который использован в алгоритме TLD (рис. 1) называется: Сопровождение-Обучение-Обнаружение (Tracking-Learning-Detection) и состоит из трёх компонентов: краткосрочного сопровождения, обучения и обнаружения. Модуль сопровождения осуществляет сопровождение объекта от кадра к кадру. Модуль обнаружения локализует все появления объекта, которые наблюдались в ближайшее время и корректирует модуль сопровождения, если это необходимо. Модуль обучения оце-

Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347.

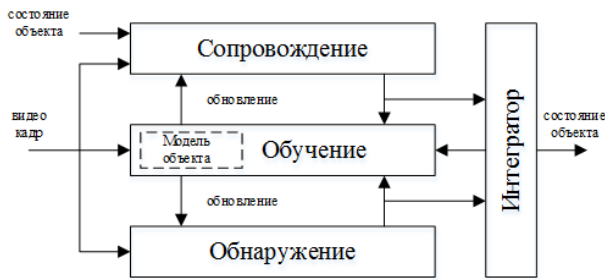


Рис. 1: Принцип работы TLD

нивает ошибки модуля обнаружения и обновляет его, чтобы избежать их в будущем. Интегратор усредняет выходные ограничительные прямоугольные рамки модулей сопровождения и обнаружения в одну выходную ограничительную прямоугольную рамку. Если на выходе ни модуля сопровождения, ни модуля обнаружения нет ограничительной рамки, объявляется что сопровождаемого объекта не видно.

Достоинства: высокая производительность; устойчивость к частичному и полному исчезновению, а также к различным изменениям внешнего вида объекта.

3.2 Принцип работы СМТ

Основная идея алгоритма СМТ (рис. 2) заключается в обнаружении ключевых точек, принадлежащих сопровождаемому объекту и их поиске в каждом кадре. Для этого используются два подхода:

- отслеживание ключевых точек от кадра к кадру с помощью оценки их оптического потока;
- сопоставление ключевых точек, путем сравнения их дескрипторов.

Поскольку оба этих метода подвержены ошибкам, осуществляется поиск согласованного решения по обоим подходам в найденных ключевых точках. По каждой ключевой точке определяется предполагаемый центр объекта. Вычисленные предполагаемые центры объекта затем интегрируются и побочные максимумы удаляются. На основе оставшихся ключевых точек вычисляется новая ограничивающая прямоугольная рамка и процесс продолжается.



Рис. 2: Принцип работы СМТ

Достоинства: высокая производительность; точное сопровождение при смещении, изменении масштаба и вращении различных классов объектов; устойчивость к частичному и полному исчезновению объекта, а также к различным изменениям внешнего вида.

3.3 Принцип работы Struck

В подходе Struck для сопровождения объектов используется метод опорных векторов с использованием функции ядра [5], который позволяет проводить обучение в режиме реального времени. Проблемой метода опорных векторов с использованием функции ядра является то, что число векторов неограниченно растет с ростом количества обучающих данных.

Для решения этой проблемы в данном подходе применяется механизм бюджетирования (или ограничения). Если бюджет уже полный и необходимо добавить новый опорный вектор, то выполняется процедура определения и удаления подходящего опорного вектора с наименьшим весовым коэффициентом и коррекция коэффициентов остальных опорных векторов. Достоинства: высокая производительность, устойчивость к шумам, устойчивость к частичному и полному исчезновению объекта.

4. Система тестирования

Для анализа выбранных алгоритмов сопровождения была разработана система тестирования (рис. 3). Реализация алгоритмов сопровождения основывалась на библиотеке OpenCV 2.4.9 [6]. В качестве тестовых видеопоследовательностей использовались записи с камер летательного аппарата. Для каждой тестовой видеопоследовательности предварительно создавалась аннотация (координаты и размеры объекта на каждом кадре) вручную в формате «XML». Результат работы алгоритмов сопровождения фиксировался, затем выполнялось сравнение аннотаций и результатов работы алгоритмов сопровождения. После сравнения формировался отчет в формате «HTML» с подробной статистикой.

Для оценки качества работы алгоритмов сопровождения на всей видеопоследовательности использовалась мера «точность сопровождения», которая для одной видеопоследовательности определяется по формуле:

$$p = \sum_{i=1}^{N(n)} (S_i / N(n)),$$

$$S_i = \begin{cases} 1, & \left[\begin{array}{l} A(R_1 \cap R) / A(R) > e \\ A(R_1 \cap R) / A(R_1) > e \end{array} \right] \\ 0, & \text{иначе} \end{cases},$$

где $N(n)$ — число кадров в видеопоследовательности, на которых присутствует объект; n — число кадров, на которых было произведено выделение сопровождаемого объекта; S_i — мера перекрытия для покадровой оценки производительности алгоритмов сопровождения (результат обнаружения на очередном кадре); R_1 — прямоугольник найденного алгоритмом на кадре объекта; R — прямоугольник объекта из файла аннотаций; $A(R)$, $A(R_1)$, $A(R_1 \cap R)$ — площади соответствующих прямоугольников и площадь пересечения этих прямоугольников; e — точность, устанавливается равной 0.3 (значение обусловлено плохим качеством видеопоследовательностей).

То есть объект считается найденным верно в кадре, если площадь пересечения найденного прямоугольника и прямоугольника из файла аннотаций не меньше, чем 30% площади каждого из них соответственно. Если объект сопровождения на очередном кадре вышел за его пределы, а алгоритм сопровождения данный объект на кадре нашёл, то S_i считается равным 0.

5. Результаты работы TLD, CMT, Struck на тестовых видеопоследовательностях, полученных с камеры летательного аппарата

Тестирование проводилось на 77 видеопоследовательностях, с различными сценариями поведения объекта, освещением, временем года, фоном, типом объекта, типом камеры и качеством съёмки. Следующие факторы осложняли работу алгоритмов: наличие посторонних объектов (данные телеметрии); плохое качество съёмки (объект трудно различить и отследить даже человеческому глазу); на некоторых видеопоследовательностях удавалось определить объект лишь в динамике; на видеопоследовательностях, снятых на тепловизор, зачастую, трудно различить очертания объектов и выделить их из общей сцены; наличие объектов близких по характеристикам с наблюдаемым объектом; составная структура некоторых объектов; значительное увеличение и уменьшение размеров сопровождаемого объекта; частичный и полный поворот камеры или объекта.

В таблицах 1, 2, 3 представлены результаты работы алгоритмов сопровождения для произвольных видеопоследовательностей.

Был проведён анализ целесообразности применения предобработки кадров перед подачей на вход алгоритма. Путём тестирования было показано, что предобработка кадров имеет смысл только для CMT, поскольку средняя точность сопровождения по всем тестовым последовательностям возросла примерно на 10%. Причем для CMT ключевые точки будут детектироваться с большей вероятностью

Таблица 1: Результаты тестирования TLD для шести произвольных видеопоследовательностей.

Номер видеопоследовательности	TLD	
	Точность, %	Ср. время, с
1	12.6	0.02
2	81.2	0.03
3	86.3	0.02
4	71	0.02
5	56.4	0.05
6	100	0.02
Среднее по 77 видео	51.7	0.04

Таблица 2: Результаты тестирования CMT для шести произвольных видеопоследовательностей.

Номер видеопоследовательности	CMT	
	Точность, %	Ср. время, с
1	97.9	0.04
2	*	-
3	98.6	0.03
4	27.1	0.03
5	20.8	0.04
6	0	0.01
Среднее по 77 видео	21.5	0.03

Таблица 3: Результаты тестирования Struck для шести произвольных видеопоследовательностей.

Номер видеопоследовательности	Struck	
	Точность, %	Ср. время, с
1	21.8	0.04
2	99.1	0.05
3	100	0.02
4	12.6	0.02
5	100	0.04
6	100	0.02
Среднее по 77 видео	32.8	0.05

Таблица 4: Результаты тестирования для двух произвольных видеопоследовательностей с применением выделения границ оператором Scharf

Номер видеопоследовательности	CMT	
	Без обработки	Выделение границ
7	48%	76.9%(+28.9%)
8	*	60.3%
Среднее по 77 видео	21.5%	31.5%(+10%)

за счёт повышения детализации объекта сопровождения. В таблице 4 приведены результаты тестирования для двух произвольных видео с применением выделения границ оператором Scharf.

В таблице 5 представлены результаты тестирова-

Таблица 5: Результаты тестирования для двух произвольных видеопоследовательностей с применением фильтра увеличивающего резкость

Номер видеопоследовательности	СМТ	
	Без обработки	Резкость (Sharpen filter)
9	20.5%	99.2% (+78.7%)
10	*	60.3%
Среднее по 77 видео	21.5%	0.05

ния для двух произвольных видеопоследовательностей с применением фильтра увеличивающего резкость.

Тестирование показало, что применять к кадрам выделение границ оператором Scharr перед подачей на вход СМТ нецелесообразно, из-за значительного повышения времени работы СМТ.

6. Реализация совместной работы алгоритмов

Для повышения точности сопровождения целесообразно использовать совместную работу алгоритмов TLD, СМТ и Struck с учётом коэффициента уверенности и весовых коэффициентов для каждого алгоритма. Весовой коэффициент определяет вклад выходного результата алгоритма при формировании итогового результата. Коэффициент уверенности необходим для определения момента потери объекта при сопровождении или для переключения с одного алгоритма на другой при совместной работе алгоритмов. В оригинальной реализации TLD — это «conf», для СМТ в качестве коэффициента уверенности можно использовать отношение найденных ключевых точек при инициализации к количеству ключевых точек на текущем кадре, а в реализации Struck — параметр «bestScore».

На рис. 3, 4, 5 представлены примеры зависимости коэффициентов уверенности TLD, СМТ и Struck от номера кадра, а также процент перекрытия, нормированный к единице от номера кадра для произвольной видеопоследовательности. Анализ коэффициента корреляции между зависимостями коэффициента уверенности от номера кадра и коэффициентом точности сопровождения от номера кадра позволяет определить целесообразность использования коэффициентов уверенности алгоритмов сопровождения при совместной реализации.

Средний коэффициент корреляции между коэффициентом уверенности и коэффициентом точности сопровождения для 77 видеопоследовательностей для TLD составляет 0.84, для СМТ 0.83, для Struck — 0.08. Таким образом для совместной работы алгоритмов можно использовать коэффициент

* не удалось детектировать ключевые точки



Рис. 3: Зависимости коэффициента уверенности и точности сопровождения от кадра для произвольной видеопоследовательности для TLD (коэфф. корреляции 0.99)



Рис. 4: Зависимости коэффициента уверенности и точности сопровождения от кадра для произвольной видеопоследовательности для СМТ (коэфф. корреляции 0.98)



Рис. 5: Зависимости коэффициента уверенности и точности сопровождения от кадра для произвольной видеопоследовательности для Struck (коэфф. корреляции 0.84)

ты уверенности TLD и СМТ, которые в должной мере описывают момент исчезновения объекта из поля зрения наблюдателя.

Результаты экспериментов на 77 последовательностях показали, что при исчезновении объекта из

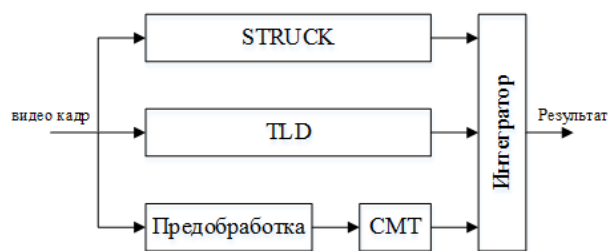


Рис. 6: Принцип совместной работы TLD, CMT, Struck

поля зрения, значение коэффициента «bestScore» алгоритма Struck некорректно по отношению к выходному результату и его использование при сопровождении нецелесообразно. Поэтому при совместной реализации при значении «bestScore» меньшим или равном 0 на более чем N кадрах (N подбирается опытным путём и равно 3-6), алгоритм Struck переинициализировался выходной ограничительной рамкой алгоритма TLD, которая подавалась на вход функции инициализации Struck при значении коэффициента уверенности TLD больше или равным 0.5. При неудачной инициализации CMT, на последующих кадрах осуществляется инициализация алгоритма выходной ограничительной рамкой алгоритма TLD, аналогично описанному выше случаю переинициализации Struck.

На рис. 6 представлена блок-схема совместной работы TLD, CMT и Struck. При инициализации каждому алгоритму присваивается свой весовой коэффициент, лежащий в диапазоне $[0, 1]$, сумма всех весовых коэффициентов не должна превышать 1. На вход каждого алгоритма подаётся кадр видеопоследовательности, перед подачей кадра на вход CMT осуществляется предобработка фильтром, повышающим резкость. Все алгоритмы работают параллельно и независимо друг от друга. В интеграторе осуществляется усреднение выходных ограничительных рамок алгоритмов с учётом весовых коэффициентов и формирование итоговой выходной ограничительной рамки с предполагаемым расположением объекта.

При равных весовых коэффициентах средняя точность сопровождения по 77 тестовым видеопоследовательностям составила 54.8%, что всего лишь на 3% выше средней точности TLD, но при оптимальных (по критерию максимизации результирующей точности сопровождения) значениях весовых коэффициентов средняя точность сопровождения возросла до 69.7%. Количество видеопоследовательностей, на которых алгоритмы показывают точность более 70% при оптимальных значениях весовых коэффициентов возросло до 44. Предложенная схема переинициализации позволила повысить точность сопровождения Struck с 32.8% до 50.5%. Точность

CMT при использовании данной схемы переинициализации и предобработки (повышение резкости) возросла с 21.5% до 57%.

Выводы

Совместное использование TLD, CMT и Struck позволяет незначительно повысить точность сопровождения при равных значениях весовых коэффициентов до 54.8%, количество видеопоследовательностей, на которых алгоритмы показывают точность сопровождения более 70% при оптимальных значениях весовых коэффициентов возросло с 33 до 44 (из 77).

Для повышения точности сопровождения целесообразно применять: предобработку изображений (повышение резкости) перед подачей кадра на вход CMT, проверку корректности полученных кадров, а также использовать данные телеметрии для уточнения положения объекта.

Предложенная схема переинициализации позволила повысить точность сопровождения Struck с 32.8% до 50.5%. Точность CMT при использовании предложенной схемы переинициализации и предобработки (повышение резкости) возросла с 21.5% до 57%.

Целью дальнейших исследований является разработка адаптивной схемы переключения и присваивания оптимальных весовых коэффициентов алгоритмам сопровождения; повышение точности сопровождения и производительности; рассмотрение новых алгоритмов принятия решения при совместной реализации; использование модели движения камеры для дополнительного уточнения местоположения объекта на текущем кадре.

Литература

- [1] А.А. Лукьяница, А.Г. Шихин Цифровая обработка видеоизображений. – Москва: «Ай-Эс-Эс Пресс», 2009. – 518с.
- [2] Л. Шапиро, Дж. Стокман Компьютерное зрение. – Москва: БИНОМ. Лаборатория знаний, 2006. – 752с.
- [3] Zdenek Kalal, Krystian Mikolajczyk, Jiri Matas Tracking-Learning-Detection // IEEE Transactions on pattern analysis and machine intelligence, 2010. — Vol. 6, no. 1.
- [4] Georg Nebehay, Roman Pflugfelder Consensus-based Matching and Tracking of Keypoints for Object Tracking // Safety and Security Department AIT Austrian Institute of Technology 1220 Vienna, Austria, 2014.
- [5] Sam Hare, Amir Saffari, Philip H. S. Torr Struck: Structured Output Tracking with Kernels // Oxford Brookes University, Oxford, UK Sony Computer Entertainment Europe, London, UK 2012.
- [6] opencv.org – OpenCV – 2015.

Зашивка дыр в объединённом наборе триангуляционных сеток*

А.П. Кудряшов, В.А. Бобков
kudryashovA@dvo.ru | bobkov@dvo.ru

Владивосток, Россия, Институт автоматизации и процессов управления, ДВО РАН

Предложен алгоритм зашивки дыр при построении 3D модели по дальностным данным. Алгоритм основан на диффузном размытии воксельного пространства с построением неявной весовой функции, как вдоль нормалей, так и вдоль линии наблюдения в октантном воксельном пространстве. Описан алгоритм текстурирования модели, учитывающий взаимное перекрытие исходных сеток.

Ключевые слова: воксельное пространство, трехмерная реконструкция, триангуляционная оболочка, зашивка отверстий, октодереву, диффузное размытие, текстурирование.

Patching holes in the joint set of triangulation grids*

A.P. Kudryashov, V.A. Bobkov

Vladivostok, Russia, Institute of Automation and Control Processes, FEB RAS

We present a method for building and texturing a continuous watertight mesh using a hole filling algorithm based on volume diffusion. A multilevel octree-based technique with the weight function being built along both the normal and line of view is proposed. A parallelized computational procedure for a multicore processor is presented which enables significant reduction in computation time.

Keywords: voxels, 3D reconstruction, meshing, filling holes, octree, volume diffusion, texturing.

1. Введение

Дальностные данные, получаемые с помощью стереокамер, лазерных дальномеров или методами компьютерного зрения по серии изображений часто используются для построения трехмерных моделей реальных сцен и объектов. Эти данные могут быть представлены в виде облака точек в трехмерном пространстве или в виде триангуляционной поверхности. Наряду с задачей объединения этих триангуляционных сеток, подразумевающей построение единой связной оболочки без дублирования участков поверхности, видимых с нескольких видов, возникают задачи зашивки дыр и нанесения текстуры на финальную модель. Под дырами подразумеваются участки, где должна быть поверхность, но по каким либо причинам она не была построена ни на одном из видов.

Важным требованием к алгоритмическому решению задачи является высокая скорость обработки данных и экономное использование оперативной памяти, поскольку для насыщенных сцен со сложными объектами может потребоваться много видов с большим количеством описывающих треугольников (порядка $10^6 - 10^7$).

Как показывает анализ известных работ, в основном, применяются два подхода для решения задачи объединения триангуляционных сеток и зашивки дыр на этих сетках. В первом подходе объединение и зашивка ведется непосредственно на самой

триангуляционной сетке [1]. Его основной недостаток – генерация взаимопересекающихся поверхностей при работе над сетками, имеющими сложную, составную топологию.

Другой подход основан на использовании воксельной структуры пространства сцены. Эффективная реализация такого подхода была предложена в работе [2]. Основным преимуществом предложенного решения является сведение исходной задачи к известной задаче построения изоповерхности в скалярном поле. Этим же коллективом разработчиков был предложен и метод зашивки дыр [3], использующий диффузное размытие построенной весовой функции внутри воксельной структуры. Недостатками данного метода можно его считать высокую ресурсоемкость, а также артефакты на результирующей модели [3, fig.10]. В работах других авторов предлагается использовать октодереву [5], чтобы сократить время работы, а также альтернативный метод расчета весовой функции [6], направленный на зашивку дыр. Однако проблема артефактов, по-прежнему, остается, а время обработки даже небольших объемов (ок. 40 млн. вокселей) может занимать несколько часов.

Предлагаемый в настоящей статье метод построения единой целостной трехмерной модели основывается на подходе, предложенном в [2] и [3], а также на более ранней работе авторов [4], и направлен на преодоление указанных недостатков. Вклад авторов состоит в разработке оригинального алгоритма, в котором: а) построение гибридной весовой функции осуществляется в октантном воксельном пространстве, как вдоль нормалей, так и вдоль линии наблюдения; б) обеспечивается опти-

Работа выполнена при частичной финансовой поддержке РФФИ, грант 15-07-00341, Программы «Дальний Восток» (проект 15-I-4-025) и опубликована при финансовой поддержке РФФИ, грант 15-07-20347

мизация вычислений за счет эффективной структурной организации данных; в) достигается высокая скорость обработки данных за счет реализации параллельных вычислений на многоядерных процессорах.

2. Описание метода

В начальной версии программы был реализован метод, предложенный в [3] с использованием октодерева и оптимизацией из [4], поскольку реальное заполнение воксельного пространства значениями весовой функции составило только 5-12% от всего объема. Предполагалось, что такая реализация позволила бы сократить затраты памяти на порядок, что в свою очередь дало бы возможность повысить размерность воксельного пространства, и таким образом повысить детализацию финальной модели при использовании того же объема оперативной памяти. Однако повышение детализации привело к невозможности зашивать большие дыры на модели за разумное количество итераций. Наряду с этим появились артефакты, связанные с ростом поверхности в неправильном направлении (т.н. «выросты»). Данный недостаток был описан в оригинальной статье [3]. На рис.1 показаны оба типа дефектов.

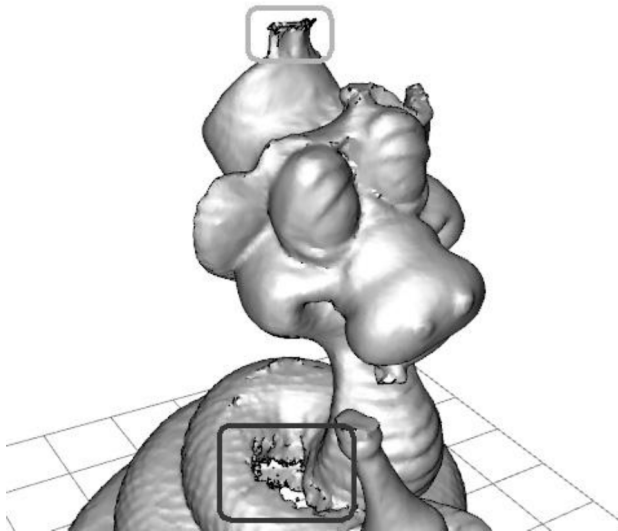


Рис. 1: Дефекты оригинального метода.

Для решения этой проблемы был предложен метод построения весовой функции не вдоль линии взгляда, а вдоль нормали к каждому треугольнику. Такой метод позволяет избавиться от части артефактов, но порождает другие, возникающие, в основном, на больших перегибах поверхности (рис.3). Объединение значений весовых функций обоих методов в одном пространстве значений также не дало желаемого улучшения.

Одновременно с проблемой «выростов» решалась и задача зашивки больших дыр. При небольшой размерности воксельного пространства дыры зашиваются, но при увеличении размерности, даже при пропорциональном увеличении количества итераций размывтия, зашивка не выполняется. Поэтому был рассмотрен вариант реализации диффузии с использованием воксельного пространства меньшей размерности с последующим пересчетом его на исходную размерность. Анализ и эксперименты показали, что оптимальным является переход на 4 верхних уровня октодерева (с уменьшением исходной детализации в 4096 раз). Именно на верхнем уровне выполняется основное количество операций размывтия с вычислительной производительностью на 3 порядка выше, чем на самом нижнем уровне октодерева. Затем осуществляется перенос зашитой поверхности с верхнего на нижний уровень. Это позволяет решить проблему зашивки больших дыр, но остается проблема «выростов». Для ее решения был предложен гибридный метод, согласно которому на нижнем уровне поверхность строится методом «по взгляду», на втором уровне поверхность строится «по нормальям», а затем производится диффузия на верхних уровнях с последующим поэтапным переносом данных на более детальные уровни. Предложенный подход позволил избавиться от дефектов обоих типов. Текстурирование конечной модели выполняется с использованием фотоизображений исходных видов. Задача нанесения текстуры сводится к поиску для каждого треугольника участка на одном из видов, который был бы наблюдаем соответствующей камерой с наименьшим углом к его нормали. В общем случае, для нахождения такого треугольника требуется перебор всех треугольников и проверка их на перекрытие (сложность n^2), но используя индексирование треугольников [4], можно понизить сложность поиска до n .

3. Алгоритмическая реализация

Используемые сокращения: $L1$ - самый детальный уровень октодерева (нижний), $L4$ - наименее детальный уровень (верхний). $L2, L3$ - промежуточные уровни.

- 1) Строим на уровне $L2$ весовую функцию методом нормалей.
- 2) Перенос данных на уровень $L3$ с применением межуровневой диффузии.
- 3) Фиксируем воксели, образующие поверхность на $L2$, остальные обнуляем.
- 4) Перенос данных на уровень $L4$ с применением межуровневой диффузии.
- 5) Фиксируем воксели, образующие поверхность на $L3$, остальные обнуляем.
- 6) Диффузное размывтие на уровне $L4$.
- 7) Перенос данных с $L4$ на $L3$, за исключением фиксированных вокселей $L3$.

- 8) Диффузное размытие на уровне L3.
- 9) Перенос данных с L3 на L2, за исключением фиксированных вокселей L2.
- 10) Диффузное размытие на уровне L2.
- 11) Очищаем L2 от вокселей, не прилегающих к поверхности.
- 12) Строим на уровне L1 весовую функцию методом по направлению взгляда. Все воксели перед поверхностью являются фиксированными, поскольку перед наблюдаемой поверхностью заведомо нет другой поверхности.
- 13) Переносим данные с L2 на L1, за исключением фиксированных вокселей L1.
- 14) Диффузия на уровне L1 - устраняем артефакты и сглаживаем переход. 2 итерации по всему октодереву с разными весовыми коэффициентами для существующей и для построенной поверхности.
- 15) Чистим L1 от вокселей, не прилегающих к поверхности.
- 17) Строим поверхность методом марширующих кубиков.
- 18) Наносим текстуру на финальную модель.

3.1 Метод объединения сеток, использующий весовую функцию по линии наблюдения

Данный метод следует оригинальному воксельному методу [1], в котором для каждого вокселя строится непрерывная неявная функция $D(V_i)$. Отличием является повышение быстродействия за счет использования индексированного массива индексов треугольников (подробнее см. в [3]). Весовая функция строится как взвешенная сумма получаемых для n видов расстояний $d_1(V_i), d_2(V_i), \dots, d_n(V_i)$ от вокселя V_i до ближайшего треугольника j -ой поверхности. Треугольник находится проецированием координат вокселя в массив индексов треугольников I . Расстояние d_j берется на луче, направленном из центра проекций O_i в точку V_i , и является величиной со знаком (отрицательное для точек, находящихся за поверхностью, и положительное для точек перед поверхностью (рис.2 а)). 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 $W(V_i)$ линейно убывает до нуля в пределах окрестности, причем для вокселей перед поверхностью размер этой окрестности выбран в данной реализации 5 вокселям, а за поверхностью - 2. Такой выбор области определения весовой функции направлен на предотвращение возникновения ложных поверхностей в областях с большими пере-

гибами поверхности. Поскольку весовая функция для каждого вокселя для каждого вида считается независимо, проход по вокселям легко параллелится на многопроцессорной архитектуре.

3.2. Метод объединения сеток с весовой функцией по нормальям

В отличие от вышеописанного метода непрерывная неявная функция $D(V_i)$ строится вдоль нормали каждого треугольника на 3 вокселя в каждую сторону от поверхности треугольника (рис.2 б). Весовая функция строится как взвешенная сумма получаемых для n видов расстояний $d_1(V_i), d_2(V_i), \dots, d_n(V_i)$ от вокселя V_i до каждого треугольника j -ой поверхности. Весовая функция $W(V_i)$ линейно убывает до нуля в пределах окрестности.

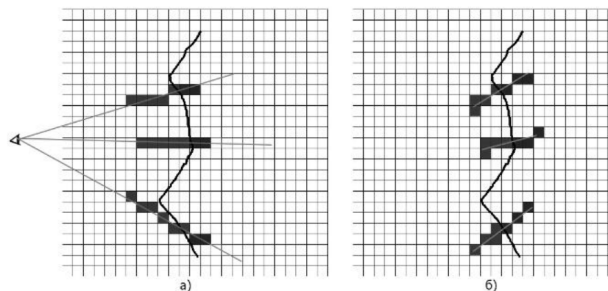


Рис. 2: Заполнение вокселей значениями весовой функции а) по линии наблюдения, б) по нормальям.

3.3 Диффузное размытие

Диффузное размытие необходимо для заполнения пустых вокселей значениями с соседних (непустых) вокселей, по этим значениям и будет строиться новая поверхность. Поскольку вся работа происходит в октодереве, то пустых вокселей в памяти не содержится и их необходимо задать. Для этого итеративно проходим по занятым вокселям и добавляем все соседние (рис.3). Количество итераций будет равно максимальному радиусу зашивки дыр. Далее, для каждого значения нефиксированного вокселя $D(V_i)$ находим новое значение равное значению $D(V_i) \times 0.25$ и сумме значений соседних 6 вокселей с коэффициентом 0.125 (рис.4). Поскольку, весовая функция для каждого вокселя рассчитывается независимо, то проход по вокселям легко параллелится на многопроцессорной архитектуре.

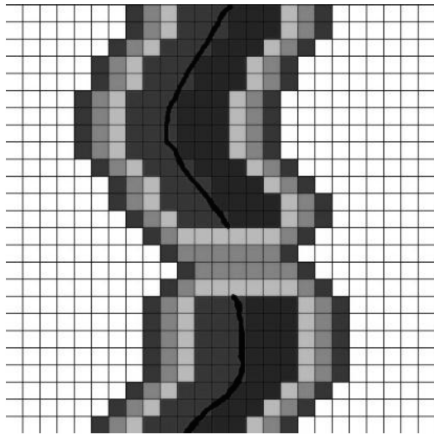


Рис. 3: Добавление пустых вокселей в октодерево. Разрыв должен быть заполнен вокселями, в которых будет происходить диффузное размытие.

3.4 Межуровневое диффузное размытие при переходе на верхние уровни октодерева

При переходе сразу на 4 уровня вверх по октодереву, т.е. при уменьшении детализации в 16 раз по каждому измерению, значения весовой функции, которые распространялись от поверхности на 3 вокселя, будут занимать всего один воксель, что недостаточно для операции диффузного размытия. Поэтому был разработан алгоритм перехода с уровня на уровень для сохранения необходимых данных. При переходе на один уровень вверх значения в вокселях на верхнем уровне должны браться не только как усредненные по значениям восьми вокселей, занимающих тот же объем на нижнем уровне, но и значениями 24-х прилегающих вокселей, т.е. по 4 с каждой из 6 сторон. Эта операция позволяет заполнить часть пустых вокселей на каждом верхнем уровне и не допускает «схлопывания» данных.

3.5 Переход на нижние уровни октодерева

Очевидно, что при диффузном размытии на верхних уровнях октодерева получаемая поверхность будет иметь низкую детализацию. Поэтому для построения финальной поверхности необходимо перенести данные с верхних уровней октодерева на нижние, не потеряв зашитую поверхность и не внося искажений в существующую. Для этого, перед переносом данных, необходимо очистить нижний уровень (кроме самого детального) от всех значений вокселей, не прилегающих к поверхности, остальные воксели зафиксировать. Данные с верхнего уровня записываем только в пустые воксели нижнего уровня. Таким образом, мы сохраняем уже построенную поверхность, а переносим с верхних уровней только достроенную поверхность.

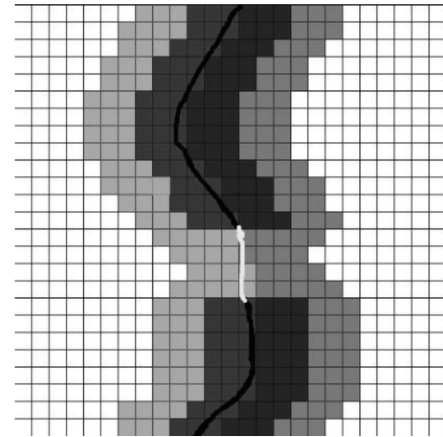


Рис. 4: Диффузное размытие. Белым цветом показана новая поверхность, построенная на месте перехода от отрицательных значений весовой функции к положительным.

4. Результаты

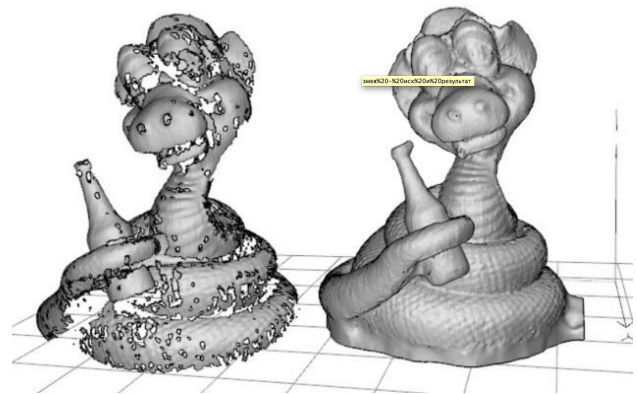


Рис. 5: Простое объединение и объединение с зашивкой дыр. Модель «Змея».

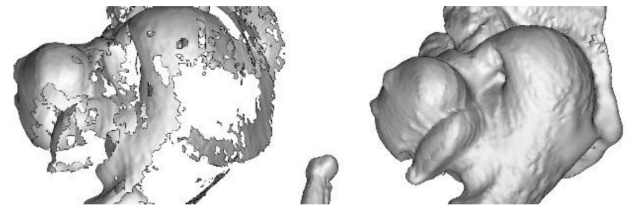


Рис. 6: Те же модели более детально.

Для получения сравнительных оценок эффективности предложенного метода были проведены вычислительные эксперименты на реальных сценах (рис. 5-8), полученных с помощью трехмерного сканера RangeVision 3D. Параметры используемого вычислительного оборудования: процессор IntelCorei5 3,0 ГГц, 4 Гб оперативной памяти. Использование многоядерности позволило увеличить

скорость работы в среднем в 2,5 раза, по сравнению с однопроцессорным вариантом.

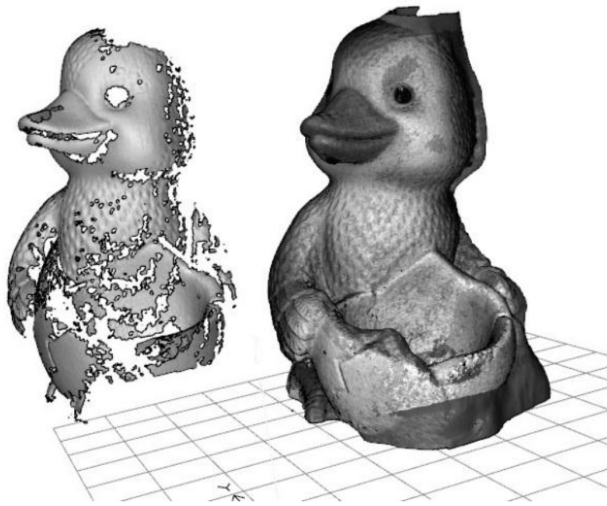


Рис. 7: Простое объединение и объединение с зашивкой дыр и текстурированием. Модель «Утка».

5. Заключение

В работе представлен метод построения связной триангуляционной трехмерной модели сцены с зашивкой дыр по множественному набору пространственных данных, основанный на воксельном подходе. Тестирование на реальных данных показало, что предложенные алгоритмические решения и структура данных с использованием октодерева, индексации треугольников, гибридного вычисления весовой функции и диффузного размытия позволили существенно повысить качество визуализации и производительность метода в сравнении с аналогами [3,5,6]. Дополнительное ускорение получено за счет использования многоядерности центрального процессора и применения параллельных вычислений.

Литература

- [1] Wei Zhao, Shuming Gao, Hongwei Lin. A robust hole-filling algorithm for triangular mesh // *International Journal of Computer Graphics*. Volume 23 Issue 12, November 2007. – pp.987–997.
- [2] Brian Curless, Marc Levoy. A Volumetric Method for Building Complex Models from Range Images // *Computer Graphics (SIGGRAPH '96 Proceedings)*. 1996. – pp.303-312.
- [3] James Davis, Stephen R. Marschner, Matt Garr, Marc Levoy. Filling holes in complex surfaces using volumetric diffusion // *First International Symposium on 3D Data Processing, Visualization, and Transmission*, Padua, Italy, June 19-21, 2002.
- [4] Бобков В.А., Кудряшов А.П. Воксельный метод построения триангуляционной поверхности по множеству видов // *Информатика и системы управления*, №2, 2012, С.31-38.
- [5] Joshua Podolak, Szymon Rusinkiewicz. Atomic Volumes for Mesh Completion // *Symposium on Geometry Processing*, July 2005.
- [6] Amitesh Kumar, Alan M. Shih. Hole Patching in Unstructured Mesh Using Volumetric Diffusion // *19th International Meshing Roundtable*, Springer-Verlag, Research Note, October 3-6 2010.

Нахождение областей, содержащих здания, на аэрофотоснимках

Л.В. Новоторцев, А.Г. Волобой

torets13@gmail.com | voloboy@in.keldysh.ru

Институт Прикладной Математики им. М.В. Келдыша РАН, Москва, Россия

Одной из основных задач фотограмметрии является обнаружение трёхмерных объектов на аэрофотоснимках. Особенно это касается задачи обнаружения строений. Алгоритмы, решающие данную задачу с наибольшей точностью, обрабатывают данные довольно долго (в основном из-за того, что необходимо выполнять сопоставление по нескольким снимкам). При анализе аэроснимков это особенно критично из-за их больших размеров. В данной статье рассматривается метод нахождения участков изображения, где имеет смысл искать строения, тем самым отбраковывая области, заведомо не представляющие интереса. Таким образом, значительно уменьшается количество данных, обрабатываемых точными алгоритмами.

Ключевые слова: компьютерное зрение, обнаружение прямых, обнаружение зданий, аэрофотоснимки.

Detection of areas containing buildings on aeroimages

L. V. Novotortsev, A. G. Voloboy

Keldysh Institute of Applied Mathematic RAS, Moscow, Russia

One of the challenging problems in photogrammetry is extracting of three-dimensional objects from aero images, especially different kinds of buildings. Methods which provide satisfactory results are rather time consuming and process data quite long (basically because there is need to match objects in different images). It is even more crucial for processing aeroimages, because of their large sizes. In the paper we propose a method that detects areas on aero images, which may contain buildings. This way we filter regions of image that possess no interest for our task and lessen amount of data to be processed by complex precise method.

Keywords: computer vision, line detection, building detection, aeroimages

Введение

Фотограмметрия занимается обработкой различного рода снимков, а именно определением формы, размера положения и иных характеристик объектов. Одним из основных направлений этой дисциплины на данный момент является анализ снимков, сделанных с воздуха (аэрофотосъёмка, съёмка с беспилотников) и с орбиты (космосъёмка). Наиболее распространённым на данный момент является аэрофотосъёмка за счёт высокого качества и доступности. Данное направление имеет широкий спектр применения: создание топографических карт, геологические изыскания, проектирование и строительство различных сооружений, автоматизированное построение пространственных моделей объекта по снимкам. В рамках этого направления часто встречается задача обнаружения и реконструкции (получение размеров и других параметров) различных строений. На данный момент существует несколько подходов решения данной задачи.

Первый подход ориентирован на построение плодотворной модели ЦМР (Цифровая Модель Рельефа) и последующую обработку этой модели с целью обнаружения и трёхмерной реконструкции зданий. Обобщённые методы выделения изображений [4, 6, 9] дают довольно неточные результаты. По этой причине часто используются методы, использующие априорную информацию об искомым объектах.

К примеру, существует метод, ориентированный на обнаружение только плоских крыш [3].

Второй подход основан на использовании структурной, контекстной и спектральной информации об изображении [8, 11, 14, 17]. В данном подходе предполагается использование для обнаружения зданий только одно изображение, и накладываются ограничения на форму объектов. Помимо этого для обнаружения используются тени, отбрасываемые зданиями, что также накладывает ограничения на входные данные.

Третий подход связан с обработкой данных, полученных при помощи специальных методов, таких как LiDAR (световое обнаружение и определение дальности). Алгоритмы, основанные на таком подходе [16, 19] дают высокий процент обнаружения объектов и низкую вероятность ложного срабатывания, однако во многих случаях подобные специальные данные недоступны.

Последний подход, представляющий для нас наибольший интерес, связан с обнаружением линейных черт на различных изображениях, последующем сопоставлением этих черт и построении трёхмерной модели на основе результатов сопоставления [1]. Такой подход обеспечивает хорошие результаты за счёт сбора информации с нескольких изображений, но довольно долго обрабатывает данные за счёт того, что необходимо выполнить сопоставление. Данный недостаток можно компенсировать,

ограничив область каждого изображения, которую обрабатывает алгоритм.

Основная идея алгоритма представленного в данной статье заключается в обнаружении отрезков на изображении и их последующем объединении в замкнутые контуры.

Основная идея

В данной статье мы предлагаем использовать тот факт, что большинство строений имеют линейные черты. По этой причине мы выбрали метод выделения отрезков на изображении в качестве основы предложенного метода. Использование этой особенности позволит увеличить точность определения необходимых объектов и облегчит вычисления. Весь алгоритм может быть разделён на следующие подзадачи:

- 1) Обнаружение границ
- 2) Выделение прямых линий и отрезков
- 3) Объединение отрезков в замкнутые контуры

Обнаружение границ

Первым шагом представленного алгоритма является обнаружение границ на изображении. Основным подходом к решению этой задачи является построение карты градиентов яркости изображения в каждой точке и использования значений вычисленных градиентов в качестве показателя границы. На этом подходе основывается фильтр Собеля [18]. В тех случаях, когда задача обнаружения границ подразумевает двоичное решение (есть граница в точке или нет границы в точке), то используется фильтр Кэнни [5]. Этот метод помимо того, что производит бинаризацию границы ещё и такие операции, как сужение границы. В нашей работе предлагается использовать модифицированный вариант метода Кэнни, дающий ещё более точные результаты [2].

Выделение отрезков

Далее по полученной карте границ производится векторизация и обнаружение отрезков. Наиболее распространённым методом решением этой задачи является алгоритм Хафа [10]. Основная идея заключается в том, чтобы построить дискретную матрицу, в которой каждая ячейка соответствует прямой: строка и колонка соответствуют длине нормали и углу её наклона. Далее для каждой точки, помеченной, как граница на предыдущем этапе, производятся следующие вычисления:

- 1) Для каждого из значений угла наклона из таблицы вычисляется длина нормали.
- 2) Для всех пар «угол-длина» в соответствующей ячейке матрицы инкрементируется её значение

Из заполненной матрицы выделяются ячейки, содержащие значения, являющиеся локальными максимумами. Значения угла и длины нормали берутся как параметры найденной прямой. Из-за дискретизации параметров в процессе поиска прямых методом Хафа может возникнуть погрешность. Изучению этой проблемы и методов её решения посвящено немало научных работ [7, 12, 13, 15].

В данной работе для решения этой задачи предлагается использовать следующий подход. Вначале мы разделяем изображение на равные квадратные области (за исключением границ изображения). Для каждой области применяется модификация метода Хафа, оперирующая с окрестностями отрезков [7]. Отрезки, найденные на соседних сегментах, объединяются, если они лежат на одной прямой.

После применения такого подхода мы получим набор отрезков, являющихся частью линейных объектов на изображении. Эти отрезки необходимо объединить для получения единых контуров.

Объединение отрезков

Для решения этой задачи было решено использовать два входных параметра для алгоритма слияния отрезков. Первый параметр – «ожидаемое расстояние между строениями» (ОРМС). Если расстояние между отрезками меньше этого значения, то считается, что они принадлежат одному объекту сцены. Вторым параметром является «ожидаемый размер строений» (ОРС). Если расстояние больше этого значения, то предполагается, что несвязанные отрезки принадлежат разным объектам. Возможность вычислять данные параметры автоматически будет исследована в дальнейших работах.

Далее поочередно рассматриваются пары ближайших друг другу отрезков, и решается надо ли их соединять. Существует три варианта взаимного расположения отрезков, каждый из которых обрабатывается отдельно:

- 1) Отрезки лежат на одной прямой
- 2) Отрезки лежат на пересекающихся прямых
- 3) Отрезки лежат на параллельных прямых

После того, как все отрезки рассмотрены, то проводится замыкание получившихся ломаных линий. Рассмотрим эти операции подробнее.

Отрезки, лежащие на одной прямой Иногда отрезок, соответствующий одной из сторон здания прерывается. Это может быть связано с плохим качеством изображения, тенями, перекрывающим объектом или малым контрастом. Необходимо иметь возможность восстанавливать такие разрывы.

Использование для этих целей порогового значения недостаточно эффективно, поскольку величина

на разрывов может сильно варьироваться. В данной статье предлагается использовать подход, основанный на использовании двух пороговых значений (ОРМС и ОРС).

Если при рассмотрении двух отрезков расстояние между ними меньше, чем ОРМС, то эти отрезки сливаются в один. Если расстояние больше, чем ОРС и больше каждого из отрезков, то отрезки не соединяются. Если не выполнено ни одно из этих условий, то производится более детальное области между отрезками.

Вначале на эту область применяется локальное растяжение яркости. Потом используется алгоритм Кэнни. После этого считается количество граничных пикселей, попавших на отрезок между рассматриваемыми отрезками. Если полученное число незначительно отличается от предполагаемой длины дополнительного отрезка, то рассматриваемые отрезки сливаются в один.

Отрезки, лежащие на пересекающихся прямых Если два отрезка находятся на пересекающихся прямых, то для каждого из них возможны два варианта положения точки пересечения.

Первый вариант – точка пересечения находится на самом отрезке. В этом случае от резков разделяется на два точкой пересечения.

Второй вариант – точка пересечения находится на продолжении отрезка. Рассмотрение этого варианта идентично рассмотрению случая с отрезками на одной прямой за тем исключением, что анализу подлежит не расстояние между отрезками, а расстояние от конца отрезка до точки пересечения.

Отрезки, лежащие на параллельных прямых При таком расположении отрезков рассмотрению подлежат следующие варианты: отрезки сдвинуты относительно друг друга, отрезки мало отличаются по длине и их концы на небольшом расстоянии друг от друга. В остальных случаях отрезки остаются как есть. В первом варианте предлагается провести проверку, описанную в разделе 5.1 для отрезка, соединяющего ближайшие концы рассматриваемых отрезков. Если проверка прошла, то они соединяются.

Во втором варианте рассматривается расстояние между отрезками. Если оно незначительно (не больше, чем 10% от ОРС), и если один из отрезков является частью контура, то тогда контур расширяется до второго отрезка. В противном случае всё остаётся как есть.

Объединение контуров

На предыдущем этапе мы получили набор ломаных линий, которые могут и не быть замкнутыми. Поэтому их необходимо замкнуть, чтобы получить замкнутый контур. Отрезок, построенный для замыкания ломаной, будем называть «замыкающим».

На данном этапе ломаные линии могут получиться разветвлёнными и иметь больше двух концов. Поскольку в данной статье описывается метод предобработки данных, то гораздо важнее не допустить пропуск объектов, чем выделение лишнего. Поэтому соединяться будут все концы, соединение которых не приведёт к пересечению замыкающего отрезка с остальной ломаной. После этого проводится повторный анализ замыкающих отрезков.

Если некоторые замыкающие отрезки имеют пересечения друг с другом (согласно условию построения они не могут пересекаться с другими частями ломаной), то они разбиваются точкой пересечения на разные отрезки (каждый из них тоже считается «замыкающим»). После этого удаляются те замыкающие отрезки, что оказались внутри контуров.

В результате данной операции мы получаем набор контуров, которые ограничивают зоны, в которых могут находиться здания. В дальнейшем более сложный алгоритм будет обрабатывать только эти зоны.

Результаты

На изображении 1 представлено исходное изображение, полученное в ходе аэросъёмки. К этому изображению мы применили предложенный метод. Результаты выделения областей представлены на изображении 2.



Рис. 1: Исходное изображение.

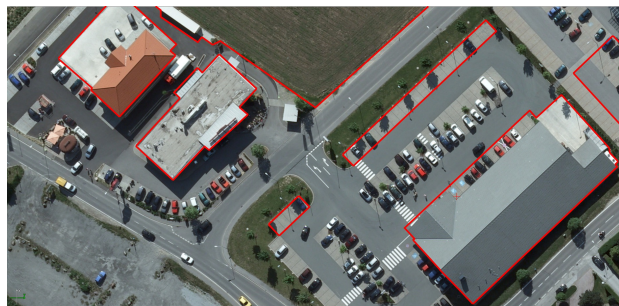


Рис. 2: Результат работы алгоритма.

Средняя точность обнаружения зданий представленным алгоритмом достаточно мала: около 71%.

Но поскольку, этот метод разработан как предобработка снимков для алгоритма, обрабатывающего снимки снятых с разных ракурсов, то это не так важно. Дело в том, что если на одном снимке строение не обнаружено, то оно будет обнаружено на другом, снятым с более удобного ракурса. Таким образом, точность обнаружения будет примерно такой же, как и в сложном методе (к примеру [1]). При этом количество данных, обрабатываемых этим алгоритмом, будет значительно меньше. На аэрофотоснимках данный алгоритм выделял 20%-60% от общей площади (в зависимости от типа местности). Это позволяет рассчитывать на ускорение порядка 30%-60%, поскольку данные, полученные на всех этапах, кроме анализа полученных отрезков и соединения их в контуры используются в алгоритме реконструкции зданий по набору аэроснимков. Дополнительные вычислительные затраты малы, поскольку обработка векторных данных вычислительно проще, чем анализ растров.

Выводы

Алгоритм, приведённый в данной статье, представляет из себя метод выделения зон, где могут располагаться строения, на основе поиска и анализа замкнутых полилиний. Этот алгоритм может быть использован, как самостоятельный метод поиска строений. Однако, результаты такого применения заметно уступают современным методам поиска изображений [8, 14, 17]. С другой стороны, рассмотренный алгоритм хорошо показывает себя в качестве предобработки изображений для последующей передачи полученных результатов алгоритму, обрабатывающему группу изображений сцены [1].

Литература

- [1] *Baillard C., Zisserman A.* A plane-sweep strategy for the 3D reconstruction of buildings from multiple image // International Archives of Photogrammetry and Remote Sensing, 2000. – Vol.33(B2), Part 2. – pp. 56-62.
- [2] *Bao P., Zhang D., Wu X.* Canny edge detection enhancement by scale multiplication // IEEE transactions on pattern analysis and machine intelligence, 2005. – Vol. 27(9). – pp. 1485-1490.
- [3] *Berthod M. et al.* High-resolution stereo for the detection of buildings. Automatic Extraction of Man-Made Objects from Aerial and Space Images // Birkhäuser Basel, 1995. – pp. 135-144.
- [4] *Brunn A., Weidner U.* Extracting buildings from digital surface models // International Archives of Photogrammetry and Remote Sensing, 1997. – Vol. 32(3), sect. 4W2. – pp. 27-34.
- [5] *Canny J.* A computational approach to edge detection // IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986. – Vol. 6. – pp. 679-698.
- [6] *Cord M., Paparoditis N., Jordan M.* Dense, reliable and depth discontinuity preserving DEM computation from HRV urban stereopairs // International Archives of Photogrammetry and Remote Sensing, 1997. – Vol. 32. – pp. 49-56.
- [7] *Du S., van Wyk B. J., Tu C., Zhang X.* An improved Hough transform neighborhood map for straight line segmentss // IEEE Transactions on Image Processing, 2010. – Vol. 19(3). – pp. 573-585.
- [8] *Ghaffarian S.* Automatic building detection based on supervised classification using high resolution Google Earth images // The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2014. – Vol. 40(3). – pp. 101-106.
- [9] *Girard S. et al.* Building detection from high-resolution color images // Remote Sensing – International Society for Optics and Photonics, 1998. – pp. 278-289.
- [10] *Hough P.* Method and Means for Recognizing Complex Patterns // U.S. Patent 3069654, Dec. 18, 1962
- [11] *Jim X., Davis C. H.* Automated building extraction from high-resolution satellite imagery in urban areas using structural, contextual, and spectral information // EURASIP Journal on Applied Signal Processing, 2005. – pp. 2196-2206.
- [12] *Li H., Lavin M. A., Le Master R. J.* Fast Hough transform: A hierarchical approach // Computer Vision, Graphics, and Image Processing, 1986. – Vol. 36(2). – pp. 139-161.
- [13] *Nguyen T. T., Pham X. D., Jeon J.* An improvement of the Standard Hough Transform to detect line segments // IEEE International Conference on Industrial Technology, 2008. – pp. 1-6.
- [14] *Ok A. O.* Automated Extraction of Buildings and Roads in a Graph Partitioning Framework // ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2013. – Vol. II-3/W3 – pp. 79-84.
- [15] *O'Rourke J.* Dynamically Quantized Spaces for Focusing the Hough Transform // IJCAI, 1981. – pp. 24-28.
- [16] *Rottensteiner F., Briese C.* A new method for building extraction in urban areas from high-resolution LIDAR data // International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, 2002. – Vol. 34(3/A). – pp. 295-301.
- [17] *Singhal S., Radhika S.* Automatic Detection of Buildings from Aerial Images Using Color Invariant Features and Canny Edge Detection // International Journal of Engineering Trends and Technology, 2014. – Vol. 11(8). – pp. 393-396.
- [18] *Sobel I., Feldman G.* A 3x3 isotropic Gradient Operator for Image Processing // Pattern Classification and Scene Analysis, 1973. – pp. 271-272.
- [19] *Sohn G., Dowman I.* Data fusion of high-resolution satellite imagery and LIDAR data for automatic building extraction // ISPRS Journal of Photogrammetry and Remote Sensing, 2007. – Vol. 62(1). – pp. 43-63.

О методе подсчета посетителей при наклонном расположении камеры*

С.С. Волкова

malysheva.svetlana.s@gmail.com

Вологодский государственный университет, Вологда, Россия

В работе рассматривается метод автоматического подсчета проходящих персон при наклонно расположенной камере. Метод основан на выделении лиц и контрольных точек и дальнейшим анализом траектории их движения. Показаны границы применимости метода и представлены экспериментальные результаты. Метод показывает высокую эффективность даже при большой плотности потока.

Ключевые слова: подсчет посетителей, обнаружение лиц, трекинг

About method of counting of visitors from the camera located slantwise*

S.S. Volkova

Vologda state unievrstity, Vologda, Russia

In this paper deals with the method of automatic counting visitors when analysing videoflow from the camera located slantwise. The method is based on faces and facial points detection in the flow and tracking the trajectory. The limits of applicability of the method are shown. The experimental results are presented. The method shows a high efficiency even at a hight flux density.

Keywords: counting of visitors, face detection, tracking

Введение

Задача подсчета посетителей является актуальной. Местом потенциального использования включают в себя торговый центр, метро, стадионы. Использование подобной системы позволяет увеличить безопасность объекта, а также проводить эффективный мониторинг движения потоков людей.

Методы подсчета посетителей можно разделить на две категории: механические и интеллектуальные [1]. Механические подразумевают использование специального оборудования, основывающегося на принципе отражения лучей от тела человека, либо по принципу прерывания луча. Подобные системы просты в установке и независимы от условий освещения, однако их эффективность падает при увеличении потока людей. Интеллектуальные системы используют анализ видеопотока, способны работать при достаточно плотном потоке людей и способны отличать человека от прочих предметов. Работа посвящена одному из методов интеллектуального подхода для подсчета проходящих людей.

Существующие подходы

В последнее время предложен ряд интеллектуальных методов, предназначенных для решения задачи подсчета посетителей. Основное направление данных работ — устойчивое моделирование фона для отслеживания перемещения объекта. В работе [3] авторы предлагают метод сегментации на основе двунаправленной гистограммы на разностном

изображении. Недостатком метода является возможность ошибки при подобию проекции другого движущегося объекта человеку. В работе [6] авторы используют нейронные сети для идентификации фона на каждом тестовом изображении. В работе [2] предложена схема подсчета при зенитном расположении камеры. Алгоритм основывается на анализе векторов движения и оценке их принадлежности объекту. В работе [5] предложена схема, основанная на построении ограничивающего прямоугольника вокруг каждой из персон. Все вышеупомянутые методы дают удовлетворительный результат при жестком задании условий, статичности фона и низкой плотности потока.

Другая группа интеллектуальных методов [4], [8] основывается на отслеживании головы и плеч людей как наиболее отличительных частей человеческого тела, что приводит к большей точности подсчета в случае большой плотности групп посетителей.

Предлагаемый подход основывается на отслеживании лиц людей в потоке и дальнейшим сопровождением и предсказанием траектории их движения, а также идентификация персон на соседних кадрах при невозможности адекватного отслеживания.

Требования к системе

Для того, чтобы позволить системе достичь производительности в реальном времени, мы налагаем ряд ограничений на видеопоследовательность:

— Последовательность кадров получается с использованием стационарной камеры с фиксированным фокусным расстоянием;

Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347.

- Люди двигаются по направлению к камере;
- Расположение камеры фиксировано, при этом точка пересечения оптической оси и плоскости земли проецируется на плоскость изображения;
- Лицо человека на кадре видимо и имеет размер не менее 100 пикселей

Графическое представление среды, удовлетворяющей условиям наблюдения, представлена на рис. 1. Следует отметить, что на способы движения и на условия исходного кадра ограничения не накладываются.

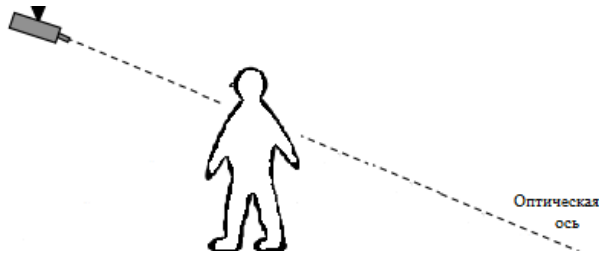


Рис. 1: Схема среды наблюдения, удовлетворяющая условиям эффективной работы метода.

Алгоритм подсчета посетителей

Предлагаемый подход основывается на анализе лица и антропометрических точек на каждом кадре видеопотока. Работа состоит из нескольких этапов: получение кадра; выделение лица и контрольных точек; проверка наличия персоны в очереди уже прошедших персон; обновление очереди персон.

При инициализации системы формируется очередь прошедших персон, которая обновляется при получении каждого кадра. При этом время жизни элемента в очереди фиксируется, что позволяет осуществлять учет персон, прошедших спустя некоторое время повторно.

Основной шаг работы — выделение лица и контрольных точек, который базируется на каскадах классификаторов на основе признаков Хаара [7], [10].

Следующий шаг — определение отношения найденных лиц к уже прошедшим персонам в очереди, который состоит из нескольких этапов: проверка по особой точке; экспоненциальное предсказание траектории; идентификация по корреляции (рис. 2).

Первый этап проверки — проверка по особой точке, включает в себя определение смещения каждой из точек лица на соседних кадрах. Если смещение контрольных точек лица не превышает допустимое отклонение — персона в очереди обнаружена и повторный ее учет не требуется. В случае превышения допустимого смещения производится переход на следующие шаги, которые необходимы в тех случаях, когда персона резко изменила положение

лица или в случае перекрытия персоны в плотном потоке людей.

Второй шаг — экспоненциальное предсказание траектории, который осуществляет предсказание положения точек лица каждого из элемента в очереди в следующий момент времени методом экспоненциального сглаживания по каждой из координат точки.

Экспоненциальное сглаживание временного ряда x_t осуществляется по следующей рекуррентной формуле:

$$S_t = \alpha x_t + (1 - \alpha)S_{t-1}, \quad (1)$$

где S_t — значение экспоненциальной средней в момент t , а S_{t-1} — в момент $t-1$; $0 < \alpha < 1$ — параметр сглаживания.

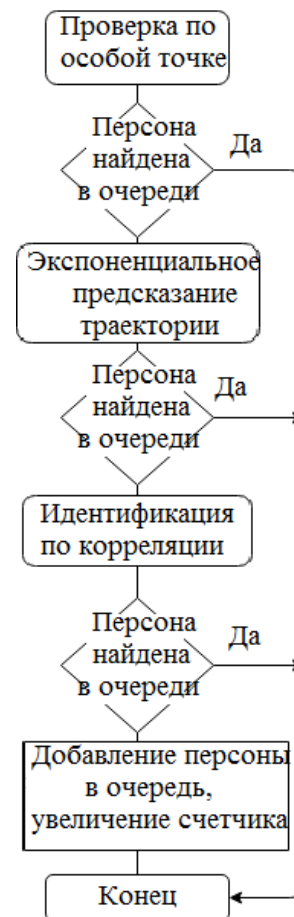


Рис. 2: Схема работы алгоритма поиска персоны в очереди уже прошедших персон.

$$S_t = S_{t-1} + \alpha(x_t - S_{t-1}) \quad (2)$$

Выражение (1) можно преобразовать к выражению (2), где экспоненциальная средняя в момент t выражена как экспоненциальная средняя предшествую-

щего момента S_{t-1} плюс доля α отклонения текущего наблюдения x_t от экспоненциальной средней S_{t-1} момента $t - 1$.

Последовательно используя (1), можно выразить экспоненциальную среднюю S_t через значение временного ряда:

$$S_t = \alpha \sum_{i=0}^{t-1} (1 - \alpha)^t x_{t-i} + (1 - \alpha)^t S_0, \quad (3)$$

где S_0 - начальные условия для первого применения (1) при $t = 1$.

Так как $1 - \alpha < 1$, то при $t \rightarrow 0$ $(1 - \alpha)^t \rightarrow 0$ и, согласно (3), имеем

$$S_t = \alpha \sum_{i=0}^{t-1} (1 - \alpha)^t x_{t-i}, \quad (4)$$

т.е. S_t - взвешенная сумма всех членов ряда. При этом веса падают экспоненциально в зависимости от давности наблюдения [9].

В случае, когда координаты предсказанной точки отклоняются на допустимое значение от точки на кадре, персона считается в очереди найденной.

Для неидентифицированных по первым двум шагам лицам производится третий этап проверки — идентификация по корреляции.

Каждое из найденных лиц масштабируется к малому размеру, положение нормализуется согласно обнаруженных антропометрическим точкам.

Алгоритм корреляции двух лиц состоит из двух шагов — фильтрация изображения лица и подсчет коэффициента схожести.

Фильтрация позволяет частично избавиться от проблемы освещенности, а также устраняет мелкие шумы на изображении. Фильтр основывается на выравнивании гистограммы в скользящем окне, их суммирование и согласно выровненным значениям — выравнивание гистограммы яркости на суммарном изображении.

Коэффициент схожести вычисляется с использованием корреляции Пирсона, которая для изображений X и Y определяется как:

$$r_{XY} = \frac{cov_{XY}}{\sigma_X \sigma_Y}, \quad (5)$$

где $\sigma_X \sigma_Y$ - стандартное отклонение яркости точек, cov_{XY} - ковариация.

Ковариация определяется как

$$cov_{XY} = \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}). \quad (6)$$

В случае если корреляция между двумя лицами превышает допустимый порог, персона считается в очереди обнаруженной.

В случае отсутствия лица в очереди персон на всех трех этапах, счетчик увеличивается, информация о персоне заносится в очередь.

Экспериментальные результаты

Для подсчета эффективности работы системы было проанализировано 5 видеороликов снятых с различных камер, длительность каждого составляла от 5 до 10 минут, число человек на каждом из кадров составляло от 0 до 10 человек. Пересечения траекторий персон в каждом из роликов содержатся. Для каждого ролика отмечалось количество реально прошедших людей человеком-экспертом и количество ответов системы. Эффективность определялась отношением количества верно подсчитанных проходов к количеству реально прошедших. Средняя эффективность по роликам составляет 89%. При этом в случае нечастого появления групп людей эффективность стремится к 100%.

Заключение

В работе представлен метод подсчета посетителей, показывающий высокую эффективность, при этом среднее время обработки одного кадра не превышает 50 мс., что позволяет осуществлять подсчет в режиме реального времени. Метод основывается на выделении лиц и контрольных точек, дальнейшего анализа изменения их положения и отслеживания траектории движения. Введение дополнительных шагов предсказания положения точек, а также подсчета корреляции между лицами позволяет учитывать проходы, независимо от наличия пересечений групп людей в плотном потоке.

Литература

- [1] *Barandiaran J., Murguia B., Boto F.* Real-Time People Counting Using Multiple Lines // Image Analysis for Multimedia Interactive Services, pp. 159 – 162, 2008.
- [2] *Chen C.H., Chang Y.C., Chen T.Y., Wang D.J.* People Counting System for Getting In/Out of a Bus Based on Video Processing // IEEE Computer Society, Eighth International Conference on Intelligent Systems Design and Applications, pp. 565-569, 2008.
- [3] *Honglian M., Huchuan L. Mingxiu Z.* A Real-time Effective System for Tracking Passing People Using a Single Camera // Proceedings of the 7th World Congress on Intelligent Control and Automation, Chongqing, China, pp. 6173-6177, 2008.
- [4] *Huazhong Xu, Pei L., Lei M.* A People Counting System based on Head-shoulder Detection and Tracking in Surveillance Video // International Conference on Computer Design and Applications, pp. VI-394 –VI-398, 2010.
- [5] *Kim J. W., Choi K. S., Choi B. D., Ko S. J.* Real-time Vision-based people counting system for security door // International Technical Conference on Circuits/Systems Computers and Communications, pp. 1416-1419, 2002.

- [6] *Schofield A.J., Mehta P.A., Stonham T.J.* A System for Counting People in Video images using Neural Networks to identify the Background scene // Journal of Pattern Recognition, Vol. 29, Issue no. 8, pp. 1421-1428, 1996.
- [7] *Viola P., Jones M.* Rapid object detection using a boosted cascade of simple features.// Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. December 2001, vol. 1, pp. 1063-6919
- [8] *Xing J., Haizhou Ai, Liwei L., Shihong L.* Robust Crowd Counting Using Detection Flow// Image Processing (ICIP), 2011, 18th IEEE International Conference on, pp. 2061 - 2064 , 2011
- [9] *Назаров М.Г.* Курс социально-экономической статистики: Учебник для вузов// Под ред. проф. М.Г. Назарова. — М.: Финстатинформ, ЮНИТИ-ДАНА, 2000 - 771 с.
- [10] *Мальшова С.С.* Модифицированный алгоритм локализации элементов лица на основе метода Виолы-Джонса // Искусственный интеллект и принятие решений, №1, 2015. — С. 35-44

Применение самоорганизующейся карты Кохонена для сегментации гиперспектральных изображений*

М.Н. Гурьева

mariag221115@gmail.com

Рязанский государственный радиотехнический университет, Рязань, Россия

Настоящая работа посвящена задаче сегментации гиперспектральных изображений. Предложенный алгоритм решения задачи и программное обеспечение, реализующее его, обеспечивают надежную идентификацию однородных по содержанию областей (водная поверхность, антропогенные объекты, растительность и т.д.) на гиперспектральных космических снимках, полученных с космического аппарата дистанционного зондирования Земли «Ресурс-П».

Ключевые слова: дистанционное зондирование Земли, сегментация, гиперспектральные изображения, карта Кохонена, нейронные сети

The Kohonen self-organizing maps application for segmentation of hyperspectral images*

M.N. Guryeva

Ryazan state radio engineering university, Ryazan, Russia

This paper is devoted to the task of hyperspectral images segmentation. The proposed algorithm and software that implements it provides reliable identification of homogeneous regions in the hyperspectral satellite images obtained from the satellite of Earth remote sensing "Resurs-P".

Keywords: hyperspectral imaging, remote sensing, segmentation, Kohonen map

Введение

Гиперспектральные изображения являются относительно новым достижением в дистанционном зондировании Земли и представляют собой трехмерный массив данных, который включает в себя пространственную информацию об объекте, дополненную спектральной информацией по каждой пространственной координате. Для получения таких изображений используется оптоэлектронная система, позволяющая одновременно производить замеры интенсивности солнечного освещения на большом числе длин волн. С использованием спектральной характеристики каждого пикселя, можно дистанционно идентифицировать множество материалов.

При обработке информации, поступающей со спутника, важным этапом является сегментация, предназначенная для выделения областей с однородными объектами, непохожими друг на друга, например, таких как водная поверхность, растительность, песок, снег, антропогенная зона и др. Это облегчает реализацию дальнейшего, более детального анализа изображения.

Алгоритм сегментации

Высокое разрешение космических снимков, а также большая размерность входных данных требует автоматизации процесса сегментации изображе-

ния. В настоящей работе рассмотрено использование самоорганизующейся карты Кохонена, представляющей собой однослойную нейронную сеть с обучением без учителя, которая является методом проецирования многомерного пространства в пространство с более низкой размерностью, в данном случае, двумерное. Преимуществами этого метода являются устойчивость к зашумленным данным, быстрое и неуправляемое обучение, а также возможность упрощения многомерных входных данных с помощью визуализации. В процессе работы сети происходит вычисление общих признаков внутри большой входной выборки. Результатом работы является кластеризация данных. Представляя входную последовательность сети как набор спектральных характеристик каждой пространственной координаты, можно выявить зависимость и оценить принадлежность этого пикселя к тому или иному классу.

Обучение сети

Первым этапом алгоритма является обучение сети. На этом шаге происходит автоматическое формирование усредненных значений спектральных характеристик каждого класса, для последующего использования их в качестве эталонных при построении карты. Количество классов на изображении определяется автоматически, в процессе обучения сети.

Формирование весовых векторов На первом этапе каждая эталонная характеристика представляет собой нормализованный вектор, заполненный

Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347.

случайными числами. После корректировки это будет отражать усредненную спектральную характеристику каждого класса.

Формирование входной последовательности для обучения Входная последовательность выбирается как набор спектральных характеристик пикселей, равномерно распределенных по изображению.

Определение принадлежности к тому или иному классу Определяется мера соответствия, в качестве которой выбрано Евклидово расстояние текущей спектральной характеристики до всех весовых векторов. Вектор, расстояние до которого минимально, признается «победителем». Изначально количество классов выбрано достаточно большим.

Корректировка весовых векторов Весовые векторы корректируются следующим образом: $m(t) = m(t-1) + h_c(t) * (x(t) - m(t-1))$, где $m(t)$ - скорректированный весовой вектор, $m(t-1)$ - нескорректированный весовой вектор, $x(t)$ - спектральная характеристика текущего пикселя $h_c(t) = \alpha(t) * \delta(t)$ - корректировочная функция. Корректировочная функция $h_c(t)$ определяется двумя коэффициентами: скоростью обучения $\alpha(t)$ и функцией соседства $\delta(t)$. Скорость обучения отвечает за точность подстройки и на начальном этапе $\alpha(t) = 0.7$. В процессе обучения сети этот коэффициент падает по экспоненциальной зависимости и к концу обучения достигает минимального значения. Это означает, что на первых итерациях алгоритма происходит жесткая корректировка весовых векторов, а на последних - «подстройка», что приводит к формированию усредненной спектральной характеристики каждого класса. В качестве функции соседства выбрана Гауссовская функция $\delta(t) = e^k, k = -S^2 / (2 * \sigma^2(t))$, где S^2 - расстояние на карте между текущим весовым вектором и «победителем», $\sigma(t)$ - множитель, уменьшающий количество соседей с итерациями, монотонно убывает.

Окончание обучения Обучение считается законченным, когда минимизируется относительная ошибка карты.

Оптимизация количества классов Перед переходом к следующему этапу производится автоматический анализ изначально большого числа классов для объединения наиболее схожих из них.

Построение карты

По окончании обучения каждому пикселю изображения на основе вычисления меры сходства спектральной характеристики с откорректированными весовыми векторами присваивается номер класса. По полученным данным формируется карта и

RGB-модель исходных данных, которые затем совмещаются для лучшего визуального восприятия. Таким образом, в результате карта Кохонена классифицирует входные данные на кластеры и визуально отображает многомерные входные данные в двумерной плоскости, распределяя векторы близких признаков в соседние ячейки и раскрашивая их в зависимости от анализируемых параметров

Экспериментальные исследования

Для проведения экспериментальных исследований было разработано программное обеспечение на языке C++. Исходные данные представляют собой гиперспектральное изображение, состоящее из 130 каналов, снятых в диапазоне длин волн 395–998 нм, полученное с космического аппарата Ресурс-П. На Рис.1 представлены:

RGB-модель исходных данных (а),
карта Кохонена (б),
сегментация алгоритмом k-средних (в).

В результате построения карты было автоматически определено 8 классов объектов на изображении. По результатам обработки можно заметить, что успешно выполнено определение водной поверхности, растительности, антропогенных объектов. По сравнению с алгоритмом k-средних достигнута более высокая производительность и низкая ошибка карты.

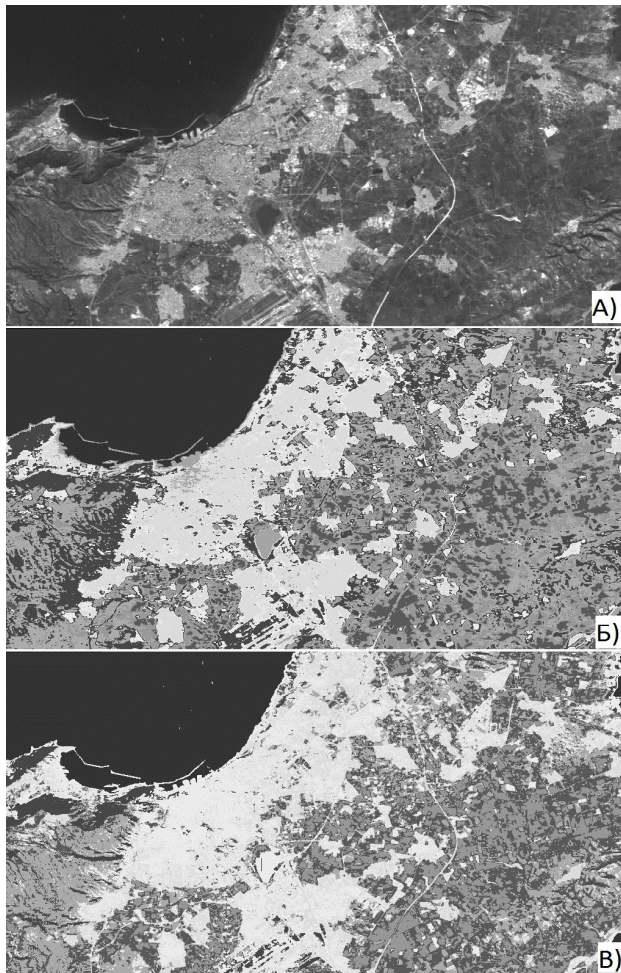


Рис. 1: а) RGB-модель исходных данных; б) карта Кохонена; в) результат обработки алгоритмом k -средних

Выводы

В данной работе был предложен алгоритм сегментации гиперспектрального изображения с использованием самоорганизующейся карты Кохонена. По сравнению с популярным алгоритмом k -средних значительно повышена производительность, а также снижена ошибка обнаружения. Анализ результатов компьютерных экспериментов позволяет говорить об успешной работе созданного алгоритма и программного обеспечения.

Система визуализации «GLView» для имитационно-тренажерных комплексов и систем виртуального окружения*

М.В. Михайлюк, М.А. Торгашев
mix@niisi.ras.ru|mtorg@mail.ru

Москва, Россия, Научно-исследовательский институт системных исследований РАН

В статье описывается архитектура и основные возможности системы визуализации трехмерных виртуальных сцен «GLView» в масштабе реального времени в моно и стерео режимах. Эта система может использоваться в моделирующих и имитационно-тренажерных комплексах управления сложными динамическими процессами, системах виртуального окружения, виртуальных лабораториях, мультимедийных руководствах и других областях.

Ключевые слова: визуализация, тренажеры, виртуальная реальность, реальное время

The System of Visualization «GLView» for Simulators and Virtual Environment Systems*

M. V. Mikhaylyuk, M. A. Torgashev
SRISA RAS, Moscow, Russia

The article describes the architecture and main features of the «GLView» system of virtual scenes' real-time visualization in the mono and stereo modes. This system can be used in simulation-training complexes of dynamic processes modeling, virtual environment systems, virtual labs, multimedia tutorials and other areas.

Keywords: visualization, training systems, virtual reality, real-time mode

Введение

Система «GLView» предназначена для визуализации трехмерных виртуальных сцен в режиме реального времени в моно и стерео режимах. Исходные виртуальные сцены подготавливаются в системе трехмерного моделирования 3D Studio MAX. Система «GLView» поддерживает использование сложных материалов, текстур и источников освещения различного типа, попершинный и попиксельный расчет освещенности, ключевую анимацию с набором различных контроллеров интерполяции, морфинг (деформацию) объектов, скелетную анимацию и т.д. Важной особенностью системы является то, что вид и движение объектов в ней в точности совпадает с тем, как их подготовил дизайнер в системе моделирования 3D Studio MAX. Кроме того, обеспечивается управление объектами, камерами и источниками света как от стандартных средств (клавиатура, мышь, джойстик), так и по сети от моделирующего комплекса с помощью специального протокола. Это позволяет использовать систему как составную часть сложных программно-аппаратных комплексов. В «GLView» реализованы также дополнительные возможности, такие как многопортовая и мультискранный визуализация, эффекты постобработки, объемные динамические эффекты, системы частиц различного типа и т.д. Режим стерео поддерживается для широкого спектра устройств стерео отображения: с использованием анаглифических,

затворных и поляризационных очков, а также шлемов виртуальной реальности и автостереоскопических мониторов. Система может использоваться независимо для просмотра моделей 3D Studio MAX, а также подключаться в виде библиотеки DLL к другим программным комплексам. Области использования системы визуализации могут быть имитационно-тренажерные комплексы управления сложными динамическими системами, системы виртуального окружения, виртуальные лаборатории, мультимедийные руководства и другие области, в которых требуется визуализация трехмерных сцен в реальном режиме времени.

Система «GLView» успешно апробирована и эксплуатируется в ряде организаций, к числу которых относится Российский государственный научно-исследовательский испытательный Центр подготовки космонавтов им. Ю.А.Гагарина, ОАО РКК «Энергия», МГТУ им. Н.Э.Баумана и др. В статье рассматриваются архитектура, основные возможности и особенности реализации системы визуализации «GLView».

Архитектура системы

Трехмерные виртуальные сцены подготавливаются в системе моделирования 3D Studio MAX и затем конвертируются в формат системы «GLView». Конвертирование осуществляется с помощью специально разработанного программного модуля (так называемого «плагины»), подключаемого к 3D Studio MAX. Результат конвертирования записывается в файловый архив, содержащий как сами

Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347

объекты трехмерной сцены, так и все текстуры, используемые для ее материалов.

Программные модули системы написаны на объектно-ориентированном языке Visual C++ в среде Windows с использованием графической библиотеки OpenGL. Система работает на персональных компьютерах с общедоступными графическими картами.

В качестве базовой технологии для вывода трехмерной графики используется графическая библиотека OpenGL. Базовые возможности ориентированы на версию 1.2, расширенные возможности визуализации (реализация сложных материалов и дополнительных эффектов визуализации) требуют версии OpenGL 4.2 с поддержкой шейдеров на языке GLSL. Система визуализации имеет модульную структуру, что позволяет наращивать список ее возможностей, а также реализовывать взаимодействие с новыми устройствами управления и вывода (например, с компьютерными перчатками и системами трекинга). Система «GLView» независима от пользовательского интерфейса и реализована в нескольких версиях:

- EXE приложение, выполняющее функции просмотра трехмерных сцен, в котором интерфейс пользователя реализован как надстройка, использующая функции системы визуализации;
- DLL версия, которая интегрируется во внешнее приложение (например, приложение подготовки мультимедийных курсов [1]), взаимодействие с которым осуществляется по согласованному программному интерфейсу;
- ActiveX DLL версия, которая интегрируется в любое приложение, поддерживающее модули ActiveX (например, Internet Explorer, Microsoft Word, PowerPoint и т.д.).

Такой подход позволяет построить произвольный программный комплекс, использующий высококачественную визуализацию трехмерных сцен, с необходимым пользовательским интерфейсом.

Возможности визуализации

Основными задачами системы «GLView» являются высокое качество визуализации и масштаб реального времени. Качество визуализации зависит от степени подробности геометрической модели объектов сцены, а также качества используемых текстур. Масштаб реального времени означает синтез каждого кадра изображения не более чем за 40 миллисекунд, что обеспечивает частоту 25 кадров в секунду. В настоящее время система «GLView» обеспечивает реальное время визуализации в моно режиме для виртуальных сцен до 4 млн. полигонов, соответственно для стерео режима сцена может содержать до 2 млн. полигонов.

Для правдоподобной имитации визуальных свойств реальных поверхностей поддерживается точное моделирование освещения объектов от множества источников света, а также моделирование таких эффектов, как отражение, зеркальный блеск, микро-рельеф поверхности, тени и многих других [2]. Система визуализации разрабатывалась с целью максимально возможной реалистичности имитации визуальных свойств поверхностей при сохранении режима реального времени синтеза изображений.

Базовой моделью освещения, реализуемой в рамках графики реального времени, является модель освещенности Блинна. Она включает моделирование рассеянной (фоновой) составляющей света, диффузной составляющей, а также моделирование зеркальных бликов. В число учитываемых параметров входят цвета отдельных составляющих материала, параметры блика, прозрачность материала, цвет и яркость источников света, а также коэффициенты затухания светимости в зависимости от расстояния. Вычисление освещенности в системе «GLView» выполняется либо повершинно (закрашивание Гуро), либо попиксельно (закрашивание по Фонгу) с помощью так называемой шейдерной обработки [3]. Такая обработка состоит в использовании не фиксированного, а программируемого графического конвейера - это многократно повышает гибкость работы с графикой и позволяет реализовать не только стандартные, но и более сложные визуальные эффекты.



Рис. 1: Моделирование теней

Примером использования шейдерной технологии также служит реализация теней и освещение объектов сцены фарой. Для моделирования теней используется метод теневых карт (shadow maps). Для получения мягкого контура тени поддерживается фильтрация теней с помощью метода PCF. Одной из известных проблем базовой технологии теневых карт является недостаточность разрешения текстурных карт при моделировании теней на пространных картах при моделировании теней на пространных картах большой протяженности. Это может

приводить к визуальным артефактам в виде пикселизации и ступенчатости границ. Для устранения данной проблемы система «GLView» реализует технологию «каскадных теней» [4] (Cascaded Shadow Maps) с поддержкой до 8 карт на каждый источник света. На рис. 1 показан пример моделирования теней. Кроме параметров освещения очень важными параметрами материалов являются текстуры – двумерные изображения, «накладываемые» на поверхность объектов. Система визуализации «GLView» реализует основные (диффузные) текстуры, текстуры прозрачности, отражения, микро-рельефа (bump map) и карты нормалей (normal map). Для синтеза отражений «GLView» использует два подхода – имитацию плоского зеркала и имитацию отражений с помощью синтеза кубической карты среды. Первый подход применяется для плоских (или близких к плоским) поверхностей, второй подход (более ресурсоемкий) применим к объектам произвольной формы.

Стереорежим. Система «GLView» поддерживает технологию отображения сцены в стерео режиме, что в первую очередь направлено на повышение степени вовлечения наблюдателя в виртуальную обстановку. Стереорежим позволяет рассматривать трехмерные образы объектов в наиболее естественном виде, с учетом бинокулярной особенности зрения человека. Ощущение объема виртуальной сцены позволяет более точно понять габариты объектов и их пространственное расположение, что обеспечивает существенно лучшее восприятие синтезируемых визуальных образов. Поддержка стереорежима реализована с использованием нескольких технологий [5]: поляризационной технологии, анаглифа, очков затворного типа, автостереоскопических мониторов, а также с помощью шлемов виртуальной реальности. Основная сложность реализации стереорежима состоит в правильном задании параметров виртуальных камер, образующих стереопару. Для правильного задания параметров виртуальных камер необходимо знать положение наблюдателя, что требует реализации трекинга головы наблюдателя. В рамках системы «GLView» разработан модуль, реализующий оптическую подсистему трекинга с использованием одной видеокамеры и конфигурации из цветных светящихся маркеров [6].

Многопортовая и мультиэкранная визуализация. При моделировании сложных динамических систем часто возникает задача имитации специальных средств наблюдения и специальных режимов отображения. В частности, очень актуален и эффективен режим многопортового отображения визуальной обстановки, когда на экран монитора одновременно выводятся изображения с нескольких виртуальных камер (рис. 2). Это позволяет

оператору лучше представить взаимное расположение узлов и механизмов в пространстве и, тем самым, повысить точность и надежность управления. Кроме того, существует необходимость имитации реальных средств наблюдения, основанных на принципе многокамерного отображения.



Рис. 2: Многопортовая визуализация

Мультиэкранная визуализация обычно предполагает, что изображение для каждого экрана формируется отдельным компьютером. Очень важно, чтобы на стыках экранов изображение оставалось непрерывным, т.е. выводилось синхронно. Для создания непрерывного панорамного изображения в системе «GLView» разработана специальная технология синхронизации формирования и вывода на экраны отдельных частей изображения виртуальной сцены. Это обеспечивает синтез непрерывного панорамного изображения на нескольких экранах с помощью нескольких компьютеров в режиме реального времени.

Эффекты пост-обработки. Система визуализации «GLView» реализует следующие специальные эффекты, использующие технологию пост-обработки: черно-белый режим, срыв синхронизации, расфокусировку, имитацию глубины резкости, засветку, а также «заплывание» ярких областей при избыточной яркости. Рассматриваемые эффекты, являясь, зачастую, специфическими искажениями наблюдаемого изображения, моделируются искусственно для наиболее точной имитации реальной визуальной обстановки. Черно-белый режим характерен для некоторых используемых и в настоящее время средств наблюдения. Эффект расфокусировки и глубины резкости наблюдается в оптических системах средств наблюдения при трансфокации объектива и изменении расстояния фокусировки. Явления засветки и «заплывания» ярких областей возникают вследствие невозможности компенсации средством наблюдения чрезмерного разброса уровней освещенности наблюдаемой визуальной обстановки, например, при попадании солнца в

кадр объектива. Все рассмотренные эффекты могут применяться как по отдельности, так и в совокупности, при этом значения управляющих параметров могут задаваться либо интерактивно, либо с помощью скриптовых команд. Системой также поддерживается визуализация в расширенном диапазоне яркостей (HDR) с последующей нелинейной тональной компрессией в стандартный диапазон монитора [7].

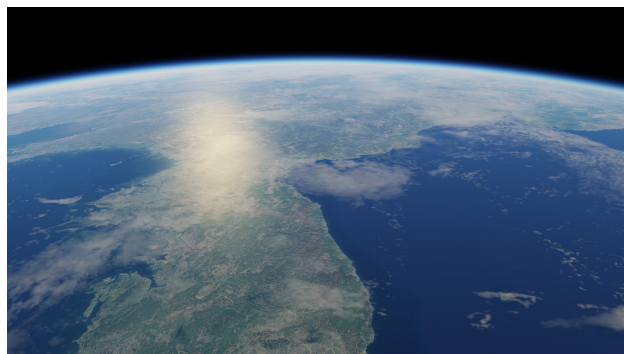


Рис. 3: Визуализация земной поверхности

Визуализация поверхности Земли. Во многих практических задачах, связанных с визуализацией, требуется модель поверхности Земли: например, в космических тренажерах для отработки задач наблюдения, ориентации и поиска отдельных объектов или районов, для определения очагов пожаров, для обеспечения необходимого фона при выполнении стыковки и т.д. В системе «GLView» реализована возможность высокореалистичной визуализации поверхности Земли [8] (а также других планет) с текстурным разрешением 150 м/пиксел и выше. В модели поверхности Земли реализован динамический облачный покров, атмосфера, линия терминатора, а также закатное и рассветное освещение (рис. 3). Кроме того, реализован облет модели Земли по кеплеровской эллиптической орбите. Разработанный модуль визуализации планет использует параллельные шейдерные вычисления на современных многоядерных графических процессорах.

Объемные эффекты. Одной из важных подзадач визуализации для имитационно-тренажерных комплексов является синтез объемных динамических эффектов. К таким эффектам относятся, в частности, такие процессы, как огонь и дым. В модуле визуализации «GLView» реализовано моделирование объемных эффектов на основе шумовых функций Перлина и их визуализация на основе шейдерной трассировки лучей в режиме реального времени [9]. Визуальные и динамические характеристики эффектов регулируются с помощью пользовательских параметров, позволяющих настроить желаемый вид эффекта и его динамику. Смодели-

рованы процессы распространения и затухания эффекта, а также дополнительное мерцающее освещение сцены от источников огня. Иллюстрация моделирования эффекта огня представлена на рис. 4.



Рис. 4: Моделирование огня

Системы частиц. Метод систем частиц широко распространен в компьютерной графике и применяется для моделирования таких эффектов, как, например, струи воды и пены, падающих листьев, а также таких природных явлений, как снег и дождь. Для системы визуализации «GLView» разработан специальный модуль [10], позволяющий моделировать такие системы частиц из 3D Studio Max, как струи, дождь и снег, а также расширенную систему частиц с дополнительными параметрами. Модулем поддерживаются частицы в форме спрайтов с произвольным материалом, а также частицы других форм. Генерация частиц и их визуализация в экранный буфер выполняются полностью на GPU с высокой степенью параллелизма, что позволяет визуализировать в реальном времени системы с количеством частиц до 1 млн. Иллюстрация работы модуля на примере моделирования снега представлена на рис. 5.

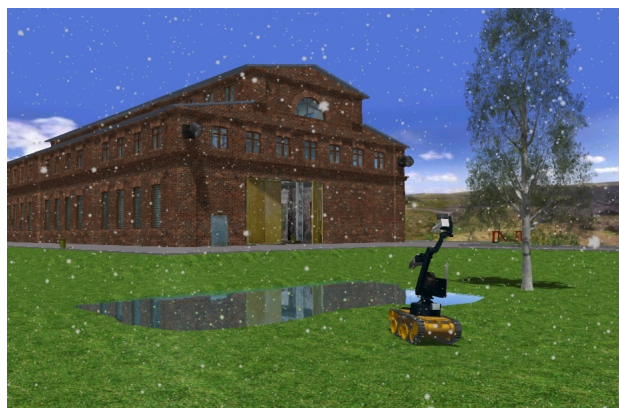


Рис. 5: Моделирование снега с помощью системы частиц

Анимация и управление

Анимация элементов сцены. Анимация объектов виртуальной сцены позволяет задать заранее известное движение объектов. Обычно это относится к неуправляемым объектам, т.е. к объектам, которые не управляются пользователем от клавиатуры, мыши или джойстика и не управляются по сети от моделирующего комплекса. Например, движение виртуальных объектов по фиксированным траекториям, деформация объекта в случае столкновения и т.д. Однако, можно анимировать не только положение и ориентацию объекта или его вершин, но и широкий класс других его параметров (нормали, цвет, текстурные координаты и т.д.), а также параметры камер (положение, направление, угол раствора) и источников освещения (положение, цвет и т.д.).

Базовой для анимации элементов трехмерных сцен является технология «ключевой» анимации. Суть ее состоит в том, что значения анимируемых параметров задаются в определенных (ключевых) кадрах, значения же в промежуточных кадрах вычисляются с помощью интерполяции ближайших ключевых кадров. Функция интерполяции при этом может быть произвольной, например, линейной или сплайновой. В системе «GLView» в рамках технологии ключевой анимации принято понятие «контроллеров», позаимствованное из системы трехмерного моделирования 3D Studio MAX – для каждого типа анимируемых параметров существует набор возможных типов контроллеров, которые и отвечают за получение значения в произвольном кадре на основе ключевых значений. Примерами контроллеров являются линейный, сплайновые по кривым Безье и по TCB сплайнам, а также более сложные (например, контроллер следования по пути). Все контроллеры реализованы в полном соответствии с тем, как они работают в системе 3D Studio MAX. Кроме базовых контроллеров реализована также анимация объектов, моделирующая их сложную деформацию - морфинг объектов и скелетная анимация. Морфинг относится к поверхностным модификаторам и позволяет реализовать сложные деформации виртуальных объектов, моделирующие поведение реальных объектов. С помощью морфинга может быть реализована лицезвая интерактивная анимация, моделирующая эмоции, мимику или речь виртуальных персонажей. Скелетная анимация также относится к модификаторам, работающим поверхностно. Ее идея состоит в задании для объекта каркаса или скелета, представляющего собой систему костей (Bones), чаще всего организованную иерархически. Сам же объект выступает в роли оболочки, «кожи» (Skin) поверх этого скелета. При анимации отдельных костей скелета оболочка повторяет эти движения в той части, где она лежит

поверх данных костей, а в местах сочленений получается плавное сопряжение поверхностей. С помощью скелетной анимации можно реализовать, например, движение человека или антропоморфного робота.

Дополнительно к возможностям анимации, задаваемым системой моделирования, система визуализации «GLView» предоставляет также возможности по регулированию воспроизведения анимационных треков. Анимация отдельных параметров может быть запущена в необходимое время и с нужной скоростью, что позволяет реализовать сложный сценарий поведения сцены в моделируемой динамической обстановке.

Управление элементами сцены. Для управления элементами трехмерных сцен в системе визуализации «GLView» предусмотрены два способа – автономное управление с помощью стандартных средств управления, а также внешнее, поступающее от системы моделирования динамики либо по сети, либо через программный интерфейс с того же компьютера.

При автономном управлении пользователю предоставляются возможности удобного управления перемещением выбранного объекта и изменением его ориентации с помощью стандартных средств, включающих клавиатуру, мышь и джойстик. Чувствительность управления и набор управляющих кнопок можно настроить под конкретную задачу в файле конфигурации.

Управление объектами по сети является штатным режимом работы для распределенных имитационно-тренажерных комплексов. В таких системах выделен отдельный моделирующий комплекс, который занимается расчетом динамики объектов сцены и анализом их столкновений. В масштабе реального времени он формирует параметры положения и ориентации всех управляемых объектов сцены. Сформированные данные передаются по сети в систему визуализации «GLView», которая производит синтез изображений виртуальной сцены. Передача осуществляется на основе специального информационного протокола, регламентирующего формат передаваемых данных. Протокол разработан для удобного управления произвольными элементами сцены и реализован как оболочка над стандартным протоколом UDP, входящим в семейство TCP/IP.

К числу немаловажных составляющих в идеологии системы «GLView» относится работа со скриптовыми командами. Скриптовая команда представляет собой совокупность текстовой строки, задающей назначение и тип команды, и связанных с ней данных произвольного формата и размера. Команды могут поступать из текстовых конфигурационных файлов (скриптов), так и формироваться отдельными модулями для передачи ядру системы,

либо другим модулям. Эта схема позволяет организовать эффективный обмен данными произвольного типа. Система визуализации имеет широкий список поддерживаемых команд: системные команды, команды работы с материалами, команды управления узлами сцены, команды конфигурирования управления и т.д.

Выводы

В работе рассмотрена система визуализации «GLView», ее базовые и расширенные возможности. Рассмотренная система может использоваться в моделирующих и имитационно-тренажерных комплексах управления сложными динамическими процессами, системах виртуального окружения, виртуальных лабораториях, мультимедийных руководствах и других областях, где требуется высококачественная визуализация трехмерных виртуальных сцен в масштабе реального времени.

Литература

- [1] Решетников В.Н., Торгашев М.А., Хураськин И.А. Система создания и просмотра мультимедийных инструкций // Программные продукты и системы. - 2007. - №3. - С.35-37.
- [2] Михайлюк М.В., Торгашев М.А. Система «GLView» визуализации для моделирующих комплексов и систем виртуальной реальности // Вестник РАЕН. - 2011. - Т. 11, №2. - С.20-28.
- [3] Мальцев А.В., Михайлюк М.В. Метод высоко реалистичной совмещенной визуализации 3D сцен в реальном времени // Труды научно-исследовательского института системных исследований Российской академии наук. - 2012. - Т. 2, №2. - С.44-50.
- [4] Мальцев А.В. Моделирование теней в 3D сценах с помощью каскадных теневых карт в режиме реального времени // Информационные технологии и вычислительные системы. - 2014. - №1. - С.46-52.
- [5] Торгашев М.А. Реализация стереорежима в реальном масштабе времени для различных устройств отображения // Программные продукты и системы. - 2010. - №2. - С.23-29.
- [6] Михайлюк М.В., Брагин В.И. Технологии виртуальной реальности в имитационно-тренажерных комплексах подготовки космонавтов // Пилотируемые полеты в космос. - 2013. - № 2(7). - С.82-93.
- [7] Тимохин П.Ю., Торгашев М.А. Визуализация космических сцен в расширенном диапазоне в режиме реального времени // Информационные технологии. - 2014. - №12. - С.53-60.
- [8] Михайлюк М.В., Тимохин П.Ю. Реалистичная модель земли в космических тренажерах // Abstracts of Thirteenth International Seminar on Mathematical Models & Modeling in Laser-plasma Processes & Advanced Science Technologies, Petrovac, Montenegro, 2015. - pp.44.
- [9] Михайлюк М.В., Торгашев М.А. Синтез объемных динамических эффектов для имитационно-тренажерных комплексов // Информационные технологии и вычислительные системы. - 2014. - №4. - С.44-50.
- [10] Мальцев А.В. Реализация системы частиц в реальном времени на GPU // Программные продукты и системы. - 2014. - №4. - С.57-62.

Сведения об авторах

Михайлюк Михаил Васильевич, д.ф.-м.н., профессор, Научно-исследовательский институт системных исследований РАН,
Email: mix@niisi.ras.ru

Торгашев Михаил Александрович, к.ф.-м.н., зав. сектором, Научно-исследовательский институт системных исследований РАН,
Email: mtorg@mail.ru

Оценка точности цифровых моделей рельефа, применяемых для территориальных исследований*

А.С.Коротин, Е.В.Попов

josefknecht1943@gmail.com|popov@sandy.ru

Нижегородский государственный архитектурно-строительный университет, г. Н.Новгород, Россия

В данной статье проведено исследование, направленное на выявления возможностей применения матриц высот Aster Global DEM и SRTM DEM для решения аналитических задач. Представлены результаты оценки точности полученных данных путём их сравнения с высотами локального участка местности.

Ключевые слова: цифровая модель рельефа, дистанционное зондирование Земли, Aster GDEM, SRTM DEM, государственная геодезическая сеть.

Evaluation of precision of digital elevation models used for territorial studies*

A.S.Korotin, E.V.Popov

The Nizhny Novgorod State University of Architecture and Civil Engineering, Nizhny Novgorod, Russia

This paper is dedicated to the research aimed at the identification of opportunities to use the Aster Global DEM and STRM DEM matrices of heights for analytical problems solution. The evaluation of the data precision obtained by comparing them with the heights of the parcel of land are presented here.

Keywords: digital elevation model, remote sensing of the Earth, Aster GDEM, SRTM DEM, geodetic control of the state

Введение

Исследования территорий, связанные с изучением объектов и явлений природы, требуют наличия качественных геометрических моделей рельефа местности, сформированных на базе моделей матриц высот. На сегодняшний день лучшей глобальной подобной моделью высокого пространственного разрешения считается Aster Global DEM, представленная для свободного использования в 2011 году. В данной статье проведено исследование, направленное на выявления возможностей применения общедоступных матриц высот Aster Global DEM и SRTM DEM для решения аналитических задач путём. Представлены результаты оценки точности полученных данных путём их сравнения с высотами локального участка местности в виде сети государственных геодезических пунктов на примере бассейна р. Кудьмы Нижегородской области. Комплексная система наблюдений за состоянием окружающей среды, оценки и прогноза изменений ее состояния под воздействием природных и антропогенных факторов предполагает осуществление систематического или непрерывного сбора информации о параметрах окружающей среды для определения тенденций их изменения. Эти наблюдения можно проводить с помощью сети стационарных пунктов, однако состояние природной среды в отдельных точках или профилях не всегда отражают пространственные изменения. По этой причине ис-

пользование данных дистанционного зондирования Земли (ДЗЗ) является необходимым условием проведения регулярных наблюдений за современным состоянием экосистем. Сравнение их с результатами съёмок, выполненных на протяжении достаточно продолжительного времени, позволяет точно зафиксировать произошедшие изменения. При этом зачастую критическое значение при решении задач, связанных с анализом результатов наблюдений, имеет точность используемой модели рельефа местности.

Современные спутниковые технологии позволяют дистанционно получать информацию и преобразовывать ее в разнообразные представления, в частности, во-первых, в DEM, а во-вторых – представлять в виде различных ЦМР (Цифровых Моделей Рельефа).

С целью оценки точности формируемых моделей в настоящей статье произведён анализ данных спутниковой радиолокационной съёмки Земли, размещённые в открытом доступе (ASTER Global Digital Elevation Map, Aster GDEM v.2). Результаты съёмки предоставляются NASA (the National Aeronautics and Space Administration of the United States) и METI (The Ministry of Economy, Trade, and Industry of Japan) с 17 октября 2011 года в нескольких обменных форматах. В сравнение с этими данными сопоставлялись материалы, опубликованные NASA в 2003 году «в виде цифровой модели рельефа (Digital elevation model, DEM) с пространственным разрешением от 30 до 90 метров. В открытом доступе были опубликованы матрицы высот с шагом 90×90 метров на всю территорию

Работа выполнена при финансовой поддержке РФФИ, грант 15-07-01962 и опубликована при финансовой поддержке РФФИ, грант 15-07-20347.

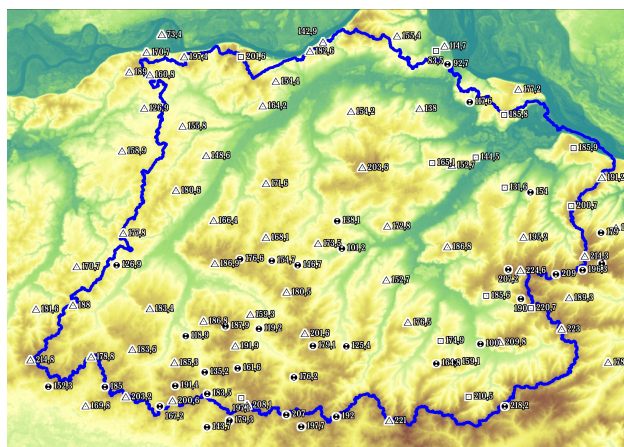


Рис. 1: Размещение сети пунктов ГГС на исследуемом участке бассейна.

Земной поверхности в виде файлов ARC GRID, а также ARC ASCII и в формате Geotiff» [1, с.30–31]. Для оценки точности цифровых моделей в качестве эталонных значений высот приняты высоты пунктов государственной геодезической сети (ГГС), расположенные в пределах рассматриваемой территории. В качестве примера исследования выбран локальный участок местности – бассейн реки Кудьмы Нижегородской области, на который имеются необходимые для анализа сведения, а именно: матрицы Aster GDEM и SRTM DEM, а также сеть пунктов ГГС [2].

Построение модели

Построение трёхмерной модели рельефа местности по матрице высот не требует длительной предварительной подготовки материалов. В простом описании для математико-картографического моделирования поверхности повторяющей формы рельефа нужен набор точечных элементов, содержащих информацию о пространственном положении $\{X, Y, Z\}$ объекта; в отдельных случаях (использование фотограмметрических методов построения) исходными данными может служить пара снимков, образующих стереопару и т.д. [3].

Рассматриваемые матрицы высот Aster GDEM и SRTM DEM представляют собой регулярную сеть (GRID) узлы которой имеют информацию о пространственном положении. По такой сети достаточно просто построить поверхность используя триангуляцию, т.к. узлы расположены в регулярной последовательности. Для оценки точности высотных данных DEM использовалась сеть пунктов государственной геодезической сети (всего 86 пунктов), отметки которых (H_{GGS}) определены в Балтийской системе высот 1977 года (см. Рисунок 1).

Анализ модели

С использованием средств геоинформационной системы MapInfo Professional [4], и приложения 3Dview [5] были получены две ЦМР (по данным SRTM и Aster GDEM). В качестве основного слоя с данными была выбрана таблица, содержащая пространственную информацию о пунктах ГГС. Данная таблица последовательно наполнялась атрибутами (высотные отметки), извлекаемыми с поверхностей ЦМР. При извлечении данных использовался интерполяционный метод на основе триангуляции:

$$h_k = \frac{\sum_{i=1}^n \left(\frac{h_i}{D_{i-k}} \right)}{\sum_{i=1}^n \left(\frac{1}{D_{i-k}} \right)}$$

где участвуют данные высот вершин треугольника (h_i) и дистанции от пункта ГГС до вершин треугольников (D_{i-k}) моделей Aster GDEM и SRTM DEM.

При этом, из-за малой протяжённости исследуемого объекта, перепроецирование на поверхность геоида не проводилось. При подобном переопределении отметок пунктов ГГС, если учитывать, что ЦМР являются приближенными к действительности, расхождения в данных должны быть минимальны, однако интерполяция показала крупные отклонения значений. Затем были определены расхождения (h_A и h_S) вновь полученных интерполяционных отметок пунктов ГГС с действительными:

$$\begin{aligned} h_A &= H_{GGS} - H_{ASGDEM} \\ h_S &= H_{GGS} - H_{SRTM} \end{aligned}$$

По полученным отклонениям были построены поверхности искажений (см. Рисунок 2). Построение поверхностей проводилось в программе «VolGA», разработанной в том числе, для решения пространственных задач и исследования трёхмерных ЦМР.

Анализ показал, что модель SRTM имеет отклонения от ГГС в пределах от –6 до +9 метров, при этом прослеживается зависимость положительных и отрицательных отклонений от покрытия территории древесно-кустарниковой растительностью, а именно: в лесных массивах расположен основной процент отрицательных отклонений, что свидетельствует о некачественной корректировке DEM за высоту древесного покрова. В то же время, на открытых участках местности DEM имеет положительные отклонения в 3 – 4 метра (см. Таблицу 1).

При сравнении Aster GDEM выявить закономерность искажений более затруднительно, т.к. имеется крупный разброс в значениях отклонений (от –16 до +21 м), однако, так же, как и в SRTM основной процент отрицательных отклонений расположен в пределах покрытых древесно-кустарниковой

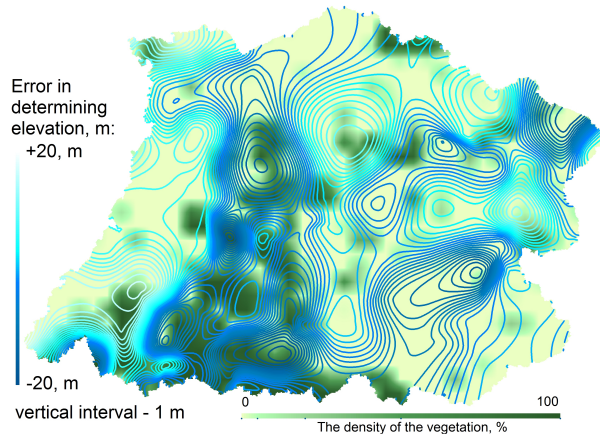
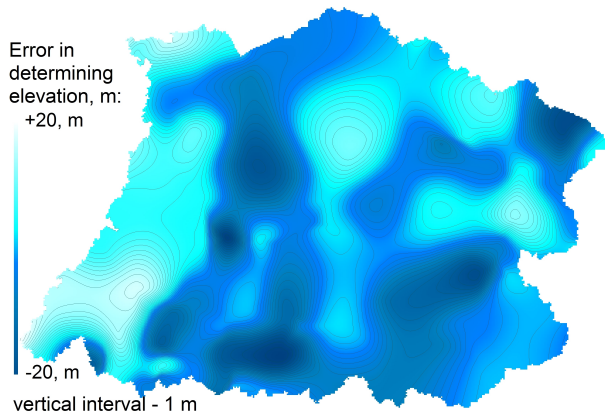
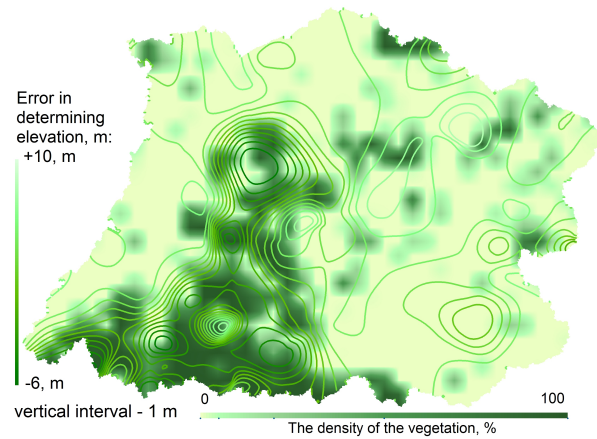
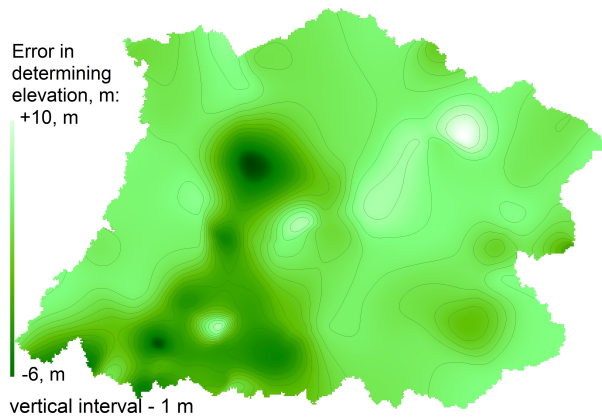


Рис. 2: Поверхности отклонений отметок DEM (SRTM – верхний и Aster GDEM – нижний) от отметок пунктов ГГС.

Рис. 3: Совмещение карты лесистости бассейна и DEM (SRTM – верхний и Aster GDEM – нижний).

Таблица 1: Процентная ведомость отклонений отметок DEM от отметок пунктов ГГС.

Интервал, м	DEM, % от общей площади	
	SRTM	Aster Global DEM
более –15	–	0,2
от –15 до –10	–	4,4
от –10 до –5	0,8	12,7
от –5 до 0	16,4	25,3
от 0 до +5	74,0	31,0
от +5 до +10	8,8	17,5
от +10 до +15	–	6,5
от +15 до +20	–	2,4
более +20	–	<0,1

растительностью участков, а положительных отклонений – в пределах открытых участков местности (см. Таблицу 1).

На схеме (см. Рисунок 3) показано наложение на карту лесистости бассейна (из «ГИС-Кудьма» [2]) полученных матриц отклонений.

Дополнительно для контроля вычислений проводилась калькуляция поверхностей DEM, путём вычитания одной из другой (см. Рисунок 4).

Качество DEM можно оценить путем сравнения

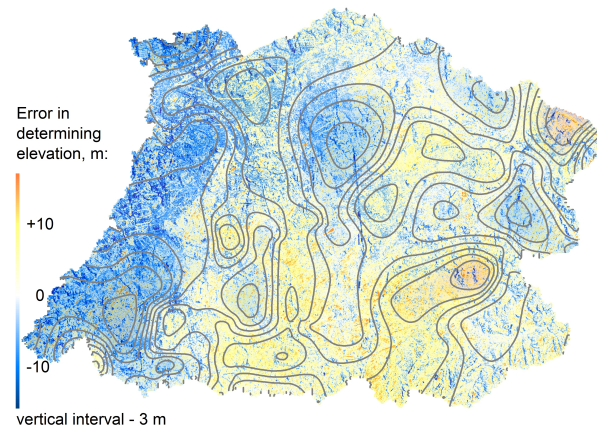


Рис. 4: Калькуляция поверхностей Aster GDEM-SRTM.

полученных результатов с нормативным значением средней квадратической погрешности определения высотного положения точки ($\frac{1}{3}h_c$) на топографической карте. Из сравнения (см. Рисунок 5) видно, что, используя данные матрицы SRTM можно получить топографическую высотную модель рельефа местности с высотой сечения рельефа 10–20 метров (СКП от 3,3 до 6,7 м, что составляет бо-

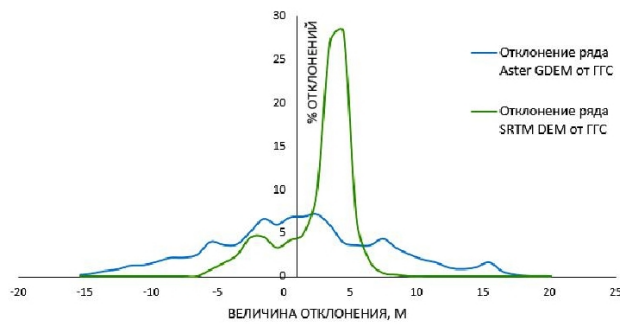


Рис. 5: График суммы процента отклонений ряда отметок.

Таблица 2: Процентная ведомость отклонений отметок DEM от отметок пунктов ГГС.

Интервал, м	DEM, % от общей площади	
	SRTM	Aster Global DEM
более -16	—	<0,1
от -16 до -10	<0,1	0,8
от -10 до -4	3,1	3,4
от -4 до 0	38,5	43,4
от 0 до +4	54,5	48,8
от +4 до +10	3,3	2,5
более +100	—	0,6

лее 90 % всех данных). Данные Aster GDEM могут быть использованы для составления цифровой модели рельефа в виде горизонталей с высотой сечения рельефа 25 метров и более.

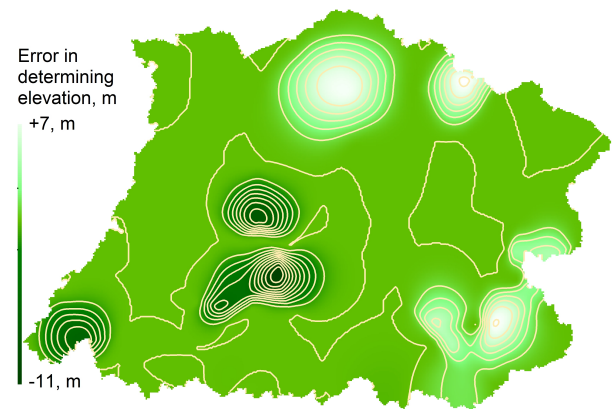
Для приведения имеющихся матриц к Балтийской системе высот (БС-1977), производилось суммирование каждой DEM со своей матрицей отклонений. Результатом стали новые DEM частично приведённые к БС-1977. Затем новые матрицы проверялись на качество отметок, по принципу, описанному выше, в результате чего получались новые матрицы отклонений (см. Рисунок 6).

В сравнении с первоначальным («сырым») вариантом откорректированные поверхности имеют более высокую степень точности и основной объем отклонений (около 90 %) уже разбросан в пределах ± 3 м (см. Таблицу 2).

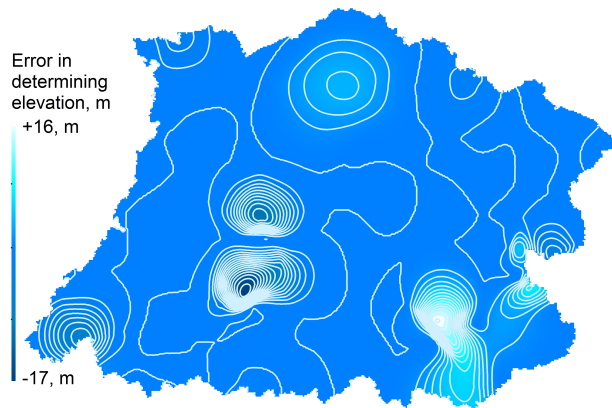
По данным таблицы был построен график (см. Рисунок 7) разброса суммарного процента отклонений новых рядов отметок. Используя данные новых DEM можно получить топографическую высотную модель рельефа местности с высотой сечения рельефа в 5 метров (СКП 1,7 м, что составляет более 85 % всех данных SRTM и Aster GDEM).

Заключение

Особенностью Aster GDEM является ее высокое пространственное разрешение (25×25 м). Распределение погрешностей по закону Гаусса в сравнении с экспозиционным распределением склонов в



Vertical interval - 1 m



Vertical interval - 1 m

Рис. 6: Поверхности отклонений отметок DEM (SRTM – верхний и Aster GDEM – нижний) от отметок пунктов ГГС.

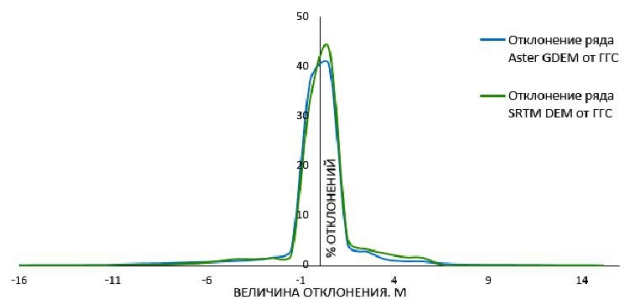


Рис. 7: График суммы процента отклонений ряда «новых» отметок.

бассейне говорит о смещении плановой привязки модели (Y-направление). На рассмотренном участке, коэффициент вытянутости территории которого близок к 1, радиус кривизны Земли достаточно мал, а форма геоида не имеет резких перепадов и крупных отклонений, в связи с чем нет необходимости проводить перепроецирование на эллипсоид, геоид или квазигеоид. Однако, приведение системы высот DEM было выполнено через поверхности отклонений. В результате были получены две равноценные в высотном плане DEM в Балтийской системе высот (см. Рисунок 8), обеспечивающих точ-

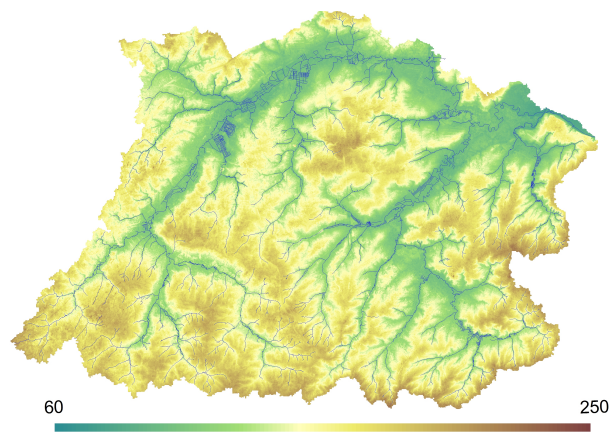


Рис. 8: Цветная цифровая модель рельефа бассейна реки Кудьма Нижегородской области в Балтийской системе высот 1977 года.

ность высотной модели рельефа местности с высотой сечения рельефа в 5 метров. Построения выровненной DEM выполнялись в программе VolGA, специально разработанной для целей трёхмерного моделирования рельефа местности, его визуализации, а также решения аналитических геоинформационных задач.

В связи с тем, что данные ГГС имели не постоянный коэффициент покрытия территорий (имелись «пробелы») в результирующих матрицах возникли сильные локальные всплески ошибок. Повысить качество перевычисления отметок матриц может использование данных более плотного и постоянного покрытия территории пунктами ГГС.

Использование подобных матриц может заменить традиционные методы работ в целом ряде гидрологических, геоморфологических, экологических и т.д. исследований. При этом будет обеспечена высокая точность исходных данных.

Выровненные DEM использовались при определении современной гидрографической сети бассейна реки Кудьма Нижегородской области и были доложены на 15-м Международном научно-промышленном форуме «Великие реки – 2015».

Литература

- [1] Коротин А.С., Попов Е.В. Восстановление трехмерных моделей рельефа местности на основе материалов дистанционного зондирования Земли // Приволжский научный журнал. – 2013. №2(30). – С.29-34.
- [2] Коротин А.С., Никольский Е.К. Проект геоинформационной системы бассейна реки Кудьмы // 15-й Международный научно-промышленный форум «Великие реки'2013» [труды конгресса], Н. Новгород: ННГАСУ, 2013. – С.406-408.
- [3] Alias Abdul-Rahman, Marakot Pilouk Spatial Data Modelling for 3D GIS. – Berlin: Springer-Verlag Berlin Heidelberg, 2008. – 289с.
- [4] MapInfo Professional: руководство пользователя // пер. с англ. фирмы ООО «ЭСТИМАП», MapInfo Corporation. New York, 2008. – 662с.
- [5] 3Dview: руководство пользователя // Москва: ООО «ЭСТИМАП», 2000. – 31с.

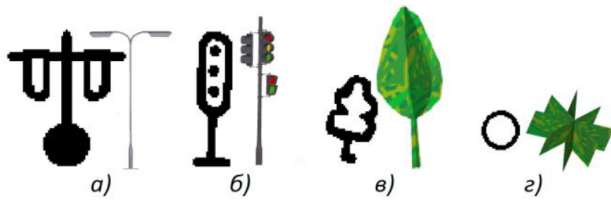


Рис. 2: Символы на топоплане и соответствующие им 3D-модели: а) фонарь; б) светофор; в) дерево; г) клумба.

В рамках данной работы будет рассматриваться только третья цель реконструкции, т.к. первые две могут быть реализованы, например, с помощью метода, описанного в [3].

Контуром является упорядоченная замкнутая последовательность точек с дискретными координатами. Пусть имеется K – массив всех контуров на топоплане, которых на нем n . Тогда K_i – некоторый контур из этого массива, где i может принимать значения от 1 до n .

Выделим несколько классов объектов, которые могут быть распознаны на топоплане: здание, символ, фонарь, светофор, дерево, клумба. Требуется определить соответствует ли контуру K_i какой-нибудь из перечисленных классов и если соответствует, то какой именно. Следует отметить, что одному объекту может соответствовать несколько контуров. Это происходит тогда, когда объект «полый», т.е. имеет область белых точек, окруженных черными. В данном случае у объекта появляется внутренний и внешний контуры.

3. Выделение контуров

Для того, чтобы выделить контуры на топоплане будет использован волновой алгоритм Ли [1]. Принцип его работы основан на запуске «волны» из некоторой точки внутри контура. Данная волна постепенно распространяется от начальной точки до тех пор, пока не пройдет по всем точкам внутренней области контура.

Чтобы получить полный список контуров требуется запускать волновой алгоритм в каждой незамеченной точке топоплана [2]. Это позволит выбрать как внутренние так и внешние контуры (Рис.3).

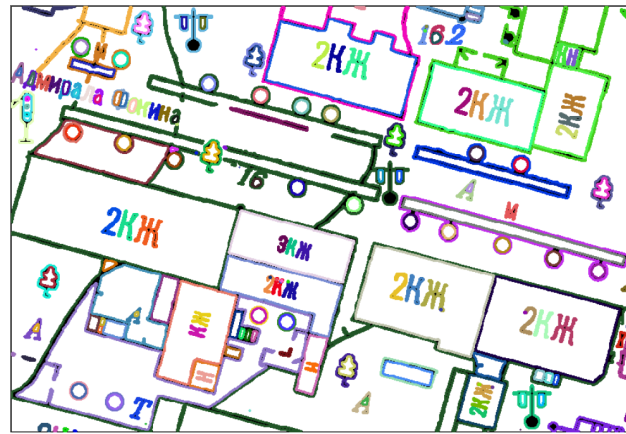


Рис. 3: Контуры на топографическом плане.

4. Свойства контуров

Для того, чтобы сопоставить каждый контур к определенному классу необходимо выделить в каждом классе свои уникальные черты.

Все объекты сцены должны иметь некоторые координаты на топоплане. Положение здания на топоплане определяется на основании координат вершин его основания. Объекты же на топоплане имеют только условное обозначение и не дают никакой информации о себе, кроме своего положения. В связи с этим требуется определить ту точку, на основании которой установлен объект. Для каждого эталонного объекта нужно задать опорную точку (Рис.4).

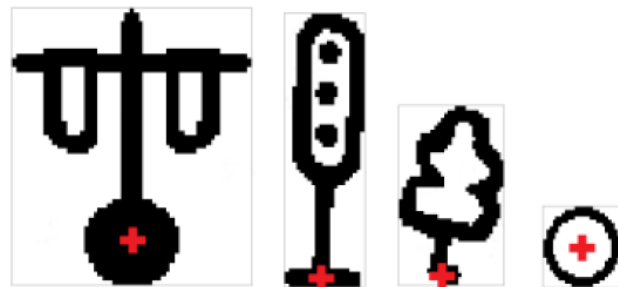


Рис. 4: Эталонные объекты с опорными точками (отмечены красным цветом).

4.1 Здание

Основная цель выделения контура здания состоит в том, чтобы определить геометрию его основания. Как правило оно состоит из прямых линий. В связи с этим, достаточно выделить только вершины основания, а промежуточные точки между ними можно исключить из контура. В большинстве случаев здания имеют форму основания с небольшим числом вершин. На основании этого можно ввести ограничивающий диапазон количества вершин.

Пусть контур K_i состоит из m точек. Тогда первой характеристикой класса зданий будет условие, что

$$4 \leq m \leq M,$$

где M – максимально допустимое число вершин в контуре здания. На практике установлено, что его можно принять равным 20. Нижняя граница выбрана с учетом того, что как правило зданий треугольной формы крайне мало.

Каждое здание на топоплане характерно тем, что имеет внутри своего контура служебную информацию, которая состоит из отдельных контуров вокруг символов. Следовательно, наличие как минимум одного контура внутри основания является еще одним признаком здания. При этом каждый вложенный контур обязательно должен быть контуром вокруг символа.

4.2 Символ

Для того, чтобы определить, является ли контур частью символа или нет, необходимо сравнить его с базой эталонов (Рис.5). Для исходного контура ищется символ с максимальной похожестью. Однако, это не дает гарантии, что контур является частью символа. Чтобы быть уверенным в этом, введем необходимый порог корреляции. На практике выведено, что допустимое значение корреляции составляет 0.7 от общего количества точек внутри контура.

0123456789ЖКМНС/

Рис. 5: Пример базы эталонов, которые встречаются в служебной информации о зданиях.

Дополнительной проверкой корректности распознанных данных является синтаксический разбор найденной последовательности символов. Он основан на том, что есть всего несколько вариантов записи служебной информации: «К», «Н», «Ж», «КЖ», «КН», «СМЖ», «СМН». Перед надписью может быть число, означающее количество этажей. Если этого числа нет, то у здания один этаж. Синтаксический контроль позволяет исправить ошибочно распознанные надписи. Например, текст вида «0МН» или «НЖ» будет преобразован соответственно в «СМН» и «КЖ».

4.3 Фонарь

Внешний контур фонарного столба на топоплане может накладываться на какой-нибудь другой объект. В связи с этим анализ внешнего контура будет проблематичен. Фонарный столб имеет внутренние контуры, которые не содержат никаких других. Определив эти контуры можно будет получить

представление о положении фонарного столба на топоплане.

Пусть имеется часть некоторого топоплана (Рис.6а), на котором обозначение фонаря накладывается на некоторую линию.

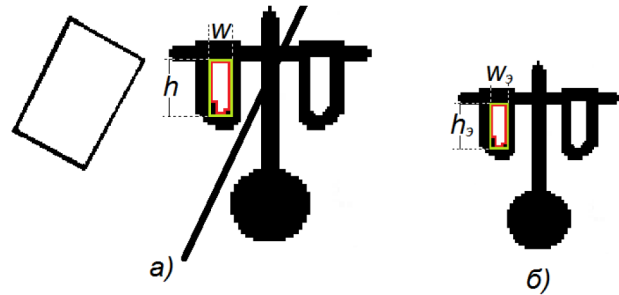


Рис. 6: Прямоугольники описаны вокруг соответствующих контуров на фонаре топоплана и эталона: а) часть топоплана; б) эталон фонаря.

Имеется внутренний контур (выделен красным). Опишем вокруг него прямоугольник, параллельный границам рисунка. Определим его ширину w и высоту h . Пусть w_ϵ и h_ϵ – ширина и высота аналогичного прямоугольника на эталоне фонаря (Рис.6б). Для данных прямоугольников должно выполняться условие:

$$|w_\epsilon/h_\epsilon - w/h| < \delta, \quad (1)$$

где δ – значение погрешности. На практике значение выбирается равным 0,4.

Обозначение фонарного столба имеет два параллельных контура, поэтому для точного определения его положения требуется выделить их оба.

Пусть расстояние между центрами прямоугольников, которые описаны вокруг внутренних контуров эталона фонаря, равно L_ϵ . Поскольку изображение фонаря на эталоне и топоплане может быть разных размеров, то определим масштабирующий коэффициент z :

$$z = w/w_\epsilon.$$

Следовательно соответствующее расстояние L между описанными прямоугольниками на фонаре топоплана будет:

$$L = zL_\epsilon$$

Выполним проверку на наличие второго контура фонаря по близости от контура, выделенного на Рис.6а. Для этого опишем вокруг каждого контура прямоугольник. У второго внутреннего контура фонаря центр описанного прямоугольника должен располагаться не дальше расстояния r от точки, находящейся на дистанции L от центра описанного прямоугольника вокруг исходного контура (Рис.7).

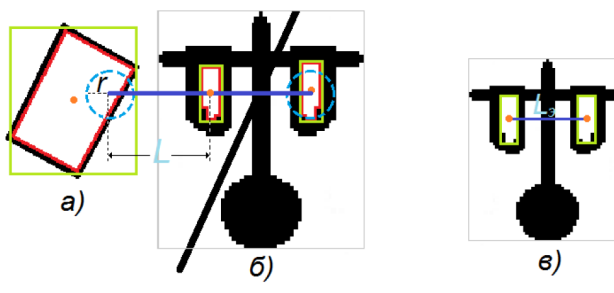


Рис. 7: Данные для распознавания фонаря: а) произвольный контур; б) фонарь на топоплане; в) эталонный фонарь.

На практике установлено, что достаточным значением γ может быть w . Если второй контур будет обнаружен, то он должен удовлетворять условию (1).

Для того, чтобы повысить вероятность распознавания фонаря, проведем сравнение эталона с некоторой прямоугольной областью топоплана. Данная область должна иметь следующую высоту и ширину:

$$W = zW_\varepsilon, \quad H = zH_\varepsilon,$$

где W, H – соответственно высота и ширина области на топоплане; $W_\varepsilon, H_\varepsilon$ – соответственно высота и ширина эталона. Левая верхняя точка данной области на топоплане будет иметь координаты x и y , определяемые как:

$$x = l - zl_\varepsilon, \quad y = t - zt_\varepsilon,$$

где l, t – координаты центра описанного прямоугольника вокруг исходного контура на топоплане; $l_\varepsilon, t_\varepsilon$ – координаты центра описанного прямоугольника вокруг исходного контура на эталоне. Зная координаты левого верхнего угла, а также высоту и ширину области, сравним её с эталоном. На практике выведено, что допустимое значение корреляции составляет 0,8 от общего количества точек эталона.

Получив подтверждение того, что область содержит обозначение фонаря, определим его опорную точку:

$$X = x + zX_\varepsilon, \quad Y = y + zY_\varepsilon,$$

где X, Y – координаты опорной точки фонаря на топоплане; $X_\varepsilon, Y_\varepsilon$ – координаты опорной точки на эталоне.

Таким образом на топоплане выделяется информация о фонарных столбах и их местоположении.

4.4 Светофор

Внутренний контур светофора должен содержать три небольших вложенных контура, центры тяжести которых примерно выстроены в вертикальную

линию. Площади этих контуров должны быть примерно одинаковыми.

Другой характеристикой класса является то, что отношение ширины и высоты прямоугольника, описанного вокруг внутреннего контура должно быть примерно таким же как и на шаблоне. Данное условие аналогично условию (1). Значение погрешности также составляет 0,4.

Если условия выше выполняются, то выделяем требуемую область на топоплане и сравниваем её с шаблоном светофора аналогично тому, как это делалось при выделении фонаря. Если порог корреляции будет больше определенного значения, то можно сделать вывод, что светофор на топоплане обнаружен. Для светофора этот порог составляет 0,8 от количества точек эталона.

4.5 Дерево

Для того чтобы задать характеристики класса контуров для дерева, воспользуемся пропорциональным отношением ширины и высоты прямоугольника, описанного вокруг внутреннего контура прямоугольника к соответствующей высоте и ширине контура на шаблоне. Внутренний контур не должен содержать вложенных.

При выполнении данных условий выделим на топоплане требуемую область аналогично тому, как это делалось при распознавании фонаря и светофора, которую сравним с шаблоном дерева. Контур будет определен как дерево, если коэффициент корреляции при сравнении будет больше 0,8.

4.6 Клумба

Клумба обозначается на топоплане обычным кругом, внутри которого ничего не должно быть. А это значит, что при некоторой погрешности отношение квадрата периметра контура к его площади будет равно всегда примерно одному и тому же числу, а именно $4/\pi \approx 1,27$. Это свойство помогает решить проблему инвариантности объекта к масштабированию. Однако, на пиксельном уровне достаточно маленькие контуры не воспринимаются как круги. Опорная точка клумбы лежит в центре круга.

5. Результат классификации объектов

Чтобы продемонстрировать работу алгоритма был использован топоплан на рисунке 1. Результат реконструкции представлен на рисунке 8.



Рис. 8: Автоматически реконструированная сцена по топоплану. Виды с разных ракурсов.

Как видно из результата реконструкции сцен, удалось воссоздать все объекты на топоплане. Автоматический алгоритм произвел распознавание за 50 секунд на топоплане с размеров 860×600 пикселей.

Возьмем другой пример (Рис.9). Размер топоплана составляет 7840×7840 пикселей.



Рис. 9: Второй пример исходного топоплана.

Результатом реконструкции показан на Рис.10. Время обработки данного топоплана составило 3 часа 10 минут.

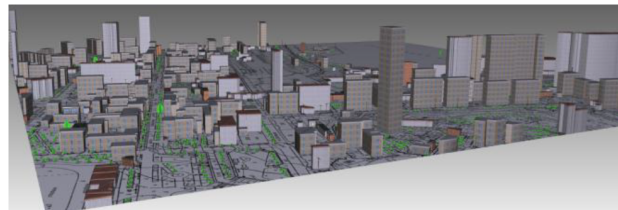


Рис. 10: Результат реконструкции (фрагмент).

6. Заключение

Предложенный способ выделения и классификации объектов на топоплане показал свою эффективность. Доля правильно распознанных объектов достигает 95-97%. Дополнительные объекты сцены придают ей дополнительный реализм, а также дают более полное представление об её устройстве.

Литература

- [1] Lee, C.Y., «An Algorithm for Path Connections and Its Applications», IRE Transactions on Electronic Computers, vol. EC-10, No.2, pp.364-365.
- [2] Кудряшов А. П., Соловьёв И. В., Распознавание контуров зданий на топографическом плане для реконструкции городских сцен // Вестник компьютерных и информационных технологий, 2015.02. С.003-008
- [3] Бобков В.А., Борисов Ю.С., Кудряшов А.П. Реконструкция и визуализация городской обстановки по изображениям. 13-я Всероссийская конференция «Математические методы распознавания образов», 2007, С.282-285.

Двухкритериальный алгоритм распознавания объектов графических изображений на базе КЭКМ*

Л.И. Лебедев, Ю.Г. Васин

lebedev@pmk.unn.ru | pmk@unn.ac.ru

НИИ прикладной математики и кибернетики

Нижегородский государственный университет им. Н.И. Лобачевского

Национальный исследовательский университет, Нижний Новгород, Россия

В работе описывается алгоритм распознавания графических изображений на базе корреляционно-экстремального контурного метода (КЭКМ) с использованием двух критериев для оценки сходства объекта с эталоном. В качестве дополнительного критерия берется оценка расстояния Хаусдорфа между контурами исходного объекта и восстановленного объекта, описание которого формируется на базе эталона, дающего наименьшее среднеквадратическое отклонение (СКО). Предлагается быстродействующий алгоритм оценивания расстояния Хаусдорфа.

Ключевые слова: распознавание, расстояние Хаусдорфа, оценка сходства, объект изображения, эталон, КЭКМ, СКО.

Twocriterial Recognition Algorithm of Graphic Objects on the Base CECM*

L.I. Lebedev, Yu.G. Vasin

Research Institute of Applied Mathematics and Cybernetics

Lobachevsky State University of Nizhni Novgorod

National Research University, Nizhni Novgorod, Russia

The paper describes the algorithm of recognition of graphic images on the basis of correlation-extreme contour method (CECM) from using two criteria for the evaluation of similarity of a object with the standard. As an additional criterion of assessment is taken Hausdorff distance between the contours of the original object and the reconstructed object, the description of which is formed on the basis of standard, giving the lowest meansquare deviation (MSD). It proposed a fast algorithm for estimating the Hausdorff distance.

Keywords: Recognition, Hausdorff distance, Evaluation of similarity, Image object, Standard, CECM, MSD.

1. Введение

Корреляционно-экстремальные контурные методы распознавания базируются на вычисление оценок сходства эталонов с объектом, инвариантных относительно ортогональных преобразований и масштабирования (ОПМ) и используют векторную (контурную [1]) модель представления объектов изображения. Получение оценки сходства осуществляется на основе вычисления среднеквадратической ошибки (оценки близости) для согласованных описаний эталона \mathbf{E} и объекта \mathbf{O} , минимальной по параметрам ОПМ [2, 3, 4]. Таким образом, если описание эталона задано последовательностью точек $\mathbf{w}^e = \{\mathbf{w}_1^e, \mathbf{w}_2^e, \dots, \mathbf{w}_m^e\}$, а объекта последовательностью $\mathbf{w} = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_m\}$, где $\mathbf{w} = (x, y)^T$, то вычисление минимального значения СКО и оптимальных параметров ОПМ осуществляется по формулам:

- минимума СКО ε_m : $\varepsilon_m = \mathbf{Dw} - R^2/\mathbf{Dw}^e$;
- угла вращения α : $\tan \alpha = S_n/C_s$;
- коэффициента масштабирования k_m :

$$k_m = R/\mathbf{Dw}^e;$$

$$- \text{смещения } \Delta \mathbf{w}^e: \Delta \mathbf{w}^e = \bar{\mathbf{w}} - k_m \cdot \mathbf{A} \cdot \bar{\mathbf{w}}^e.$$

В приведенных формулах приняты следующие обозначения используемых величин и правил их нахождения: – значения величин S_n и C_s вычисляются по смешанным корреляционным моментам описаний контуров эталона и объекта

$$S_n = \text{cov}(\mathbf{x}^e, \mathbf{y}) - \text{cov}(\mathbf{y}^e, \mathbf{x}),$$

$$C_s = \text{cov}(\mathbf{x}^e, \mathbf{x}) + \text{cov}(\mathbf{y}^e, \mathbf{y});$$

– R находится из выражения $R = S_n^2 + C_s^2$, а Dw^e и Dw находятся по значениям по координатных дисперсий описаний контуров эталона и объекта соответственно $\mathbf{Dw}^e = \mathbf{Dx}^e + \mathbf{Dy}^e$, $\mathbf{Dw} = \mathbf{Dx} + \mathbf{Dy}$; – $\bar{\mathbf{w}}^e$, $\bar{\mathbf{w}}$ являются средними значениями описаний контуров \mathbf{E} и \mathbf{O} .

Вычислительная сложность КЭКМ определяется сложностью получения смешанных корреляционных моментов и с учетом специфики их нахождения оценивается величиной $O(m+k)$.

2. Постановка задачи

Базовая оценка сходства в КЭКМ отражает среднеквадратическую близость контуров эталона и

Работа выполнена и опубликована при финансовой поддержке РФФИ, гранты 13-07-00521, 15-07-20347.



Рис. 1: Оптимальное наложение контуров.

объекта. Известно, что она наиболее эффективна при искажениях описаний контуров, полученных при наложении случайного шума, представляющего нормальное распределение $N(0, \Sigma)$. Однако, такую же величину СКО можно получить и при различных искажениях контуров при наложении импульсной помехи. В таких случаях, например, буква 'ш' может быть опознана как символ 'щ', если в списке эталонов она отсутствует. Это приводит к ошибкам при распознавании и, как следствие, к снижению качества распознавания. На рисунке 1 черным цветом представлены исходные изображения контуров букв, а синим цветом - оптимальное наложение контуров эталонных символов.

Простое решение этой проблемы состоит в повышении порога на степень сходства, однако при распознавании это ведет к увеличению числа отказов (что также снижает качество распознавания). При распознавании с самообучением это ведет к необоснованному увеличению числа эталонов, а также к уменьшению быстродействия алгоритма. Кроме того, распознавание на базе КЭКМ является основой при формировании эффективного формата представления сжатого представления графических изображений. А при решении задачи сжатия изображений выдвигается требование восстановления описаний объектов с контролируемой точностью, что в полном объеме не может обеспечить базовая оценка сходства. Таким образом, обозначенные проблемы распознавания требовали введения дополнительного критерия сходства, основанного на вычисление максимального расстояния d_H между контурами объекта и эталона при их оптимальном наложении. Отсюда следует, что дополнительная оценка сходства, базирующаяся на нахождении расстояния d_H , должна вычисляться только в случае, когда базовая оценка сходства объекта с эталоном будет удовлетворять требуемым условиям.

3. Методы решения

Для оценки степени отклонения двух контуров, исходного и восстановленного, предлагается использовать расстояние Хаусдорфа. В соответствии с определением, расстояние Хаусдорфа для двух множеств, которые представлены точками контура эталона $\mathbf{z} \in \mathbf{C}_E$ и контура объекта $\mathbf{v} \in \mathbf{C}_O$, будет находиться в соответствии с формулой [5]:

$$d_H = \max\left\{ \sup_{\mathbf{z} \in \mathbf{C}_E} \inf_{\mathbf{v} \in \mathbf{C}_O} \|\mathbf{z} - \mathbf{v}\|, \sup_{\mathbf{v} \in \mathbf{C}_O} \inf_{\mathbf{z} \in \mathbf{C}_E} \|\mathbf{z} - \mathbf{v}\| \right\}.$$

Нахождение расстояние Хаусдорфа d_H в вычислительном плане является в этом случае сложной, и в общем случае, трудно разрешимой задачей в режиме реального времени. Однако в нашем случае требуется не само значение расстояние Хаусдорфа d_H , а в основном его оценка δ_H , которую можно было бы использовать в качестве критерия при распознавании объектов изображения.

Из анализа алгоритма получения базовой оценки близости следует, что вычисление смешанных корреляционных моментов $\text{cov}(\mathbf{x}^e, \mathbf{x})$, $\text{cov}(\mathbf{x}^e, \mathbf{y})$, $\text{cov}(\mathbf{y}^e, \mathbf{x})$, $\text{cov}(\mathbf{y}^e, \mathbf{y})$, требует формирования вспомогательных описаний контуров эталона и объекта [4].

Вспомогательные описания эталона и объекта получают на основании исходных описаний путем вставки на этих контурах дополнительного числа точек, иницируемых вершинами многоугольников объекта и эталона соответственно. Для этого длины ребер исходного описания многоугольников объекта и эталона при вставлении новых точек вначале масштабируются в пропорции, равной отношению длин контуров (то есть в отношении S^e/S и S/S^e соответственно). В вспомогательных описаниях количество вершин как у эталона так и объекта будет одинаковое и не будет превышать $n = (m + k - 1)$.

Поэтому, если вспомогательные описания эталона и объекта будут согласованы и получены для эталона с наибольшей оценкой сходства, то восстановленный контур объекта можно использовать для получения оценки δ_H . Действительно, если $\tilde{\mathbf{w}}^e = \{\tilde{\mathbf{w}}_1^e, \tilde{\mathbf{w}}_2^e, \dots, \tilde{\mathbf{w}}_n^e\}$ вспомогательное описание эталона \mathbf{E} , а $\tilde{\mathbf{w}} = \{\tilde{\mathbf{w}}_1, \tilde{\mathbf{w}}_2, \dots, \tilde{\mathbf{w}}_n\}$ объекта \mathbf{O} , то восстановленное описание объекта будет представлено выражением $\tilde{\mathbf{w}} = \{\tilde{\mathbf{w}}_1, \tilde{\mathbf{w}}_2, \dots, \tilde{\mathbf{w}}_n\}$, где координаты точек $\tilde{\mathbf{w}}_i$ находятся по формуле $\tilde{\mathbf{w}}_i = k_m \cdot \mathbf{A} \cdot \tilde{\mathbf{w}}_i^e + \Delta \mathbf{w}^e$ при оптимальных значениях параметров $(\alpha, k_m, \Delta \mathbf{w}^e)$, \mathbf{A} – матрица вращения на угол α .

Тогда в качестве оценки расстояния Хаусдорфа первоначально берется величина

$$\delta_H = \max_{i=1, n} \|\tilde{\mathbf{w}}_i - \tilde{\mathbf{w}}_i\|$$

В случае, когда значение $\delta_H < P_H$ меньше максимально допустимой величины, то и расстояние Хаусдорфа будет удовлетворять этому условию и, следовательно, восстановленный контур эталона будет удовлетворять требуемым условиям. Если для некоторой вершины j выполняется неравенство $r_j > P_H$, где $r_j = \|\tilde{\mathbf{w}}_j - \tilde{\mathbf{w}}_j\|$, то вычисляются расстояние d_j от точки $\tilde{\mathbf{w}}_j$ до ломаной линии $\tilde{\mathbf{w}}_{j-1}\tilde{\mathbf{w}}_j\tilde{\mathbf{w}}_{j+1}$ (или расстояние d_j от точки $\tilde{\mathbf{w}}_j$ до

одномерных симплексов, сформированных на базе ребер, примыкающих к вершине j). Аналогично находится расстояние \tilde{d}_j от точки $\tilde{\mathbf{w}}_j$ до ломаной линии $\dot{\mathbf{w}}_{j-1}\dot{\mathbf{w}}_j\dot{\mathbf{w}}_{j+1}$ и из полученных расстояний выбирается максимальное значение $d_j : d_j = \max\{\dot{d}_j, \tilde{d}_j\}$. Далее, из полученных значений d_j выбирается наибольшее, которое и принимается за уточненную оценку близости контуров δ_H . Если при этом будет выполняться неравенство $\delta_H > P_H$, то результат восстановления не будет удовлетворять предъявляемым требованиям, и на основе этого объекта будет сформирован новый эталон при распознавании в режиме с самообучением, а в режиме распознавания этот объект будет отнесен к классу "отказов". Если для обновленной оценки расстояния Хаусдорфа будет выполняться неравенство $\delta_H < P_H$, то дальнейшего уточнения оценки δ_H не предполагается, и предъявляемый объект будет отнесен к распознанному.

4. Полученные результаты

Описанный алгоритм получения оценки близости δ_H был встроен в процедуру распознавания с самообучением, реализованной на базе КЭКМ.

Тестируемым изображением была страница текста, выполненная шрифтом Times New Roman. При использовании дополнительной оценки близости δ_H список эталонов увеличился еще на 12 единиц. Половину из них представляли заглавные буквы, которые опознавались по эталонам прописных букв при использовании только базовой оценки сходства. Была исправлена ошибка при распознавании буквы 'ш' путем добавления эталона этого символа. Еще 5 эталонов было дополнительно получено на основе букв полужирного шрифта. В итоге из совокупности 5174 контуров данного изображения было выделено 78 эталонов. Качество распознавания повысилось на 0.8%.

5. Заключение

Проведенный анализ сложности алгоритма распознавания с использованием двух критериев оценки сходства показала, что основное снижение быстродействия происходит за счет увеличения числа эталонов. Вычисление дополнительной оценки близости δ_H практически не сказывается на быстродействии алгоритма распознавания.

Литература

- [1] Делоне Б.Н., Райков Д.А. Аналитическая геометрия. // Том 1. – М.: Государственное изд-во технико-теоретической литературы, 1948. – 456 с.
- [2] Васин Ю.Г., Лебедев Л.И., Пучкова О.В. Контурные корреляционно-экстремальные методы обнаружения и совмещения объектов видеoinформации. // Автоматизация обработки сложной графической информации: Межвуз. темат. сб. науч. тр. // Под ред. Ю.Г.Васина.- Горьков. гос. ун-т, Горький, 1987. С.97-112.
- [3] Васин Ю.Г., Лебедев Л.И. Задача нахождения согласованных описаний в корреляционно-экстремальных контурных методах распознавания. // Математические методы распознавания образов (ММРО-15): 15-ая Всеросс. конф.: Сборник докладов. / М.: Изд-во ООО "МАКС Пресс 2011. – С.342-345.
- [4] Лебедев Л.И. Корреляционно - экстремальные контурные методы распознавания. Теоретические основы: Учебное пособие. – Нижний Новгород, Изд-во Нижегородского государственного университета, 2013. – 113 с.
- [5] Хаусдорф Ф. Теория множеств. // М.-Л.: Объединенное научно-техническое изд-во НКТП СССР, 1937. – 305 с.

Исследование методов пространственного индексирования динамических сцен на основе регулярных октодеревьев

В.А.Золотов, К.С.Петрищев, В.А.Семенов

vladislav.zolotov@ispras.ru | k_petrishchev@ispras.ru | sem@ispras.ru

Москва, Россия, Институт системного программирования РАН

Статья посвящена развитию и исследованию методов пространственного индексирования и анализа сложных динамических сцен, возникающих в приложениях компьютерной графики, робототехники, анимации, виртуальной и дополненной реальности, САПР, системах nD-моделирования и планирования проектов. Подобные сцены представляются композицией большого числа протяженных геометрических объектов, проявляющих индивидуальное динамическое поведение. Главное внимание в статье уделяется алгоритмам исполнения типовых пространственных запросов с использованием регулярных динамических октодеревьев. В частности, исследуются алгоритмы определения столкновений, выборки по заданной области, поиска ближайшего соседа. Для введенных модельных наборов данных на основе вероятностного анализа выводятся оценки сложности для построения индексов и исполнения типовых запросов в среднем. Полученные оценки существенно улучшают известные пессимистические результаты и служат обоснованием целесообразности применения регулярных октодеревьев для пространственного индексирования масштабных динамических сцен. Результаты проведенных вычислительных экспериментов подтверждают полученные теоретические результаты и иллюстрируют возможности создания эффективных приложений компьютерной графики в условиях перманентно растущей сложности визуальных моделей.

Ключевые слова: компьютерная графика, пространственное индексирование, анализа динамических сцен, регулярные динамические октодеревья.

Octree-based approach to spatial indexing of complex dynamic scenes

V.A.Zolotov, K.S.Petrishchev, V.A.Semenov

Moscow, Russia, Institute for System Programming of the RAS

The paper is dedicated to theoretical research of spatial indexing methods in conformity to three dimensional scenes arising in CAD/CAM systems, robotics, virtual and augmented reality applications. Special attention is given to the decomposition methods based on regular dynamic octrees. A particular version of the octrees is described and investigated to satisfy to the efficiency requirements for typical spatial queries in complex dynamic scenes with the extended borders objects. To perform the needed complexity analysis, the developed octree structure is analyzed against typical spatial queries like collision detection, frustum culling. To obtain more relevant estimates of the complexity on average rather than known pessimistic estimates in worst case, model scene datasets have been introduced. They assume uniform distribution of equal size objects over the scene volume and, being parametric, allow simulation of wide range of industry meaningful scenes. Some auxiliary lemmas about octree depth and distribution of the scene objects over octree levels have been proven based on probabilistic analysis. Final theorem statements provides for the complexity estimates of query evaluation for the model scenes. The obtained results show that asymptotic complexity grows with the object sizes and the scene occupancy. However, the estimates on average improve known results in worst case and can be considered as a theoretical background to introduce the investigated indexing structures to practical applications.

Keywords: computer graphics, spatial indexing, analysis of dynamic scenes, regular dynamic octree.

Введение

Стремительный рост объемов информации, а также необходимость ее анализа приводят к развитию новых подходов к управлению данными и, в частности, методов пространственного индексирования, без которых невозможен быстрый поиск и обработка в геоинформационных базах данных, системах логистического обеспечения, системах автоматизации проектирования, системах управления проектами. Как правило, популярные универсальные и специализированные СУБД предусматривают для этих целей средства пространственного индексирования и поиска. Подобные средства успеш-

но справляются с обработкой статической информации, однако часто не приспособлены для данных, подлежащих перманентным изменениям [1]. Проблемы эффективного поиска и анализа еще более усложняются, когда информация представляет собой не просто массивы точек, а структурированные наборы данных, например, множества объектов с протяженными пространственными границами [6]. Класс подобных приложений чрезвычайно широк и охватывает не только перечисленные выше прикладные области, но и многочисленные системы компьютерной графики, визуализации и анимации. Эти факторы определяют актуальность темы и огромный интерес, как

со стороны научного сообщества, так и производителей системного и прикладного программного обеспечения. В частности, активные разработки в этой области ведут компании Google, Oracle, IBM, Autodesk, Bentley, Intergraph, AVEVA, сталкивающиеся с проблемой увеличения объемов анализируемой пространственно-временной информации.

В нашей работе [8] были системно проанализированы фундаментальные семейства методов индексации и поиска многомерных данных. В частности, рассматривались структуры поиска на интервалах, сбалансированные ветвистые деревья во внешней памяти, бинарные деревья пространственной декомпозиции, префиксные деревья, нерегулярные и регулярные многоуровневые сетки, метрические деревья, а также связанные с ними разнообразные методики хэширования, расщепления и кластеризации.

Большое внимание было уделено методам пространственной индексации на основе регулярных октодеревьев. Главным их достоинством является простота развертывания и модификации индексов, обусловленная априори известным положением секущих плоскостей. Это позволяет относительно легко обновлять индексы и обеспечивать эффективность исполнения типовых пространственных запросов при перманентных изменениях самих данных. Однако данные методы не обеспечивают сбалансированность структур индексов и могут приводить к их деградации в тех случаях, когда данные неравномерно распределены по пространству или имеют протяженные границы. Известные оценки сложности в подобных наихудших случаях приводят к пессимистическим выводам и не определяют реальных границ применимости методов декомпозиции на основе регулярных октодеревьев. По-видимому, они нуждаются в более детальном исследовании и развитии, исходя из оценок сложности в среднем, которые могут быть получены на основе вероятностного анализа основных алгоритмов при необходимой конкретизации условий прикладных задач.

Пространственная декомпозиция на основе октодеревьев

Начнем с анализа классического метода пространственной декомпозиции, основанного на регулярных октодеревьях. В трехмерном случае параллелепипед, пространственно ограничивающий весь набор данных (или в англоязычной литературе AABV — Axis Aligned Bounding Box), разбивается на восемь равных частей плоскостями, перпендикулярными каждой из координатных осей. Процесс рекурсивно применяется до тех пор, пока количество элементов данных в каждом вновь образованном октанте не окажется ниже некоторого предустановленного порога $m > 1$. С результирующим

пространственным разбиением ассоциируется соответствующее дерево. Корень дерева соответствует исходному параллелепипеду, содержащему в себе весь набор данных, а вершины — вложенным октантам, группирующим данные на разных иерархических уровнях пространственной декомпозиции. Пример декомпозиции и полученного октодеревья приведен на рис. 1.

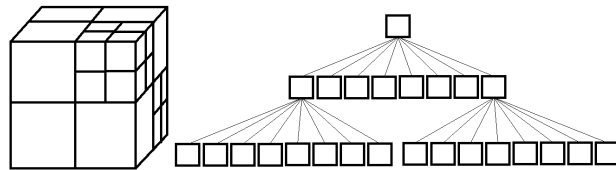


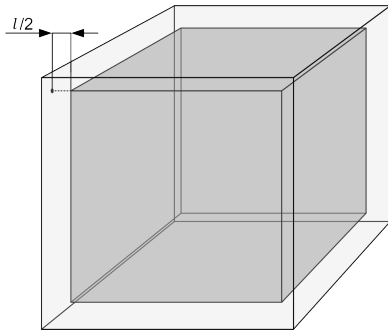
Рис. 1: Пример пространственной декомпозиции и соответствующего ей октодеревья.

Заметим, что способ ассоциирования элементов данных с вершинами октодеревья, в конечном счете, определяется способом их дальнейшей локализации и поэтому зависит от их пространственной семантики. Точечные данные могут быть непосредственно ассоциированы с листовыми октантам. Однако геометрические объекты с протяженными границами могут занимать определенный объем исходного параллелепипеда. Поэтому более рациональным представляется их ассоциирование с теми октантам, в которых они могут быть размещены полностью [5]. Для такого способа есть и иные причины, связанные с возможным пересечением секущими плоскостями даже небольших объектов и невозможностью установить простое соответствие между ними и листовыми октантам.

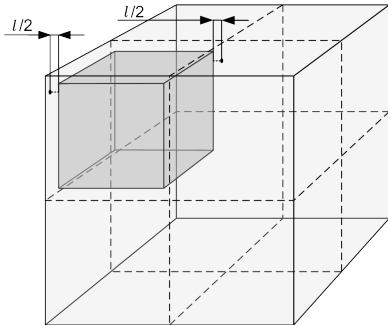
Определение 1. *Октодеревом со строгой многоуровневой локализацией объектов и верхней границей кардинальности ячеек m назовем регулярным и обозначим $Octree(m)$.*

Регулярное октодеревье может использоваться для решения различного рода вычислительных задач, связанных с поиском и анализом пространственно-временных данных. Естественно, что затраты на построение дерева должны сполна компенсироваться более быстрым решением целевых задач. Оценим затраты на построение октодеревья. Заметим, что подобные оценки могут существенно зависеть от специфики прикладных данных и анализ наихудшего случая, как правило, приводит к довольно пессимистическим результатам, не отражающим реальные показатели производительности для большинства приложений. Поэтому получим оценки сложности в среднем на основе вероятностного анализа некоторого упрощенного набора данных.

Для этого рассмотрим набор данных, связанный с визуальным анализом пространственно-трехмерных сцен и допускающий содержательную параметризацию. Пусть сцена представляется набором n трехмерных геометрических объектов, независимо случайным образом расположенных внутри единичного куба. Ограничивающим объемом каждого объекта является куб с характерным размером сторон $0 \leq l \leq 1$. В рамках подобной постановки x, y, z -координаты центров являются независимыми равномерно распределенными величинами на соответствующих отрезках $[l, 1 - l]$. Для обсуждаемых задач анализа сцен будет применяться единая техника пространственной локализации объектов на основе их ограничивающих объемов, поэтому в дальнейшем мы не делаем никаких различий между понятиями геометрического объекта и его ограничивающего параллелепипеда.



(а) область возможного расположения центров объектов внутри ограничивающего объема набора данных.



(б) область возможного расположения центров объектов, локализуемых в октантах следующего уровня.

Рис. 2: Область возможного расположения центров объектов внутри ограничивающего объема набора данных, а также внутри октантов следующего уровня.

Рисунок 2 иллюстрирует область возможного расположения центров объектов внутри объема всей

сцены, а также внутри октантов следующего уровня. Назовем величину

$$\rho(n, l) = \frac{nl^3}{(1-l)^3}$$

фактором пространственной наполненности.

Определение 2. Набор n одинаковых кубов с ребрами, ориентированными вдоль главных координатных осей и имеющими размер $0 \leq l < 1/2$, назовем модельным и обозначим как $S(n, l)$, если они независимым и случайным образом помещены в единичный куб.

Заметим, что при относительном размере объектов $l \geq 1/2$ октарная структура вырождается, поскольку все объекты локализируются в самом верхнем октанте при глубине дерева $h = 1$. Поэтому при дальнейшем рассмотрении мы ограничимся содержательным случаем $l < 1/2$. Исследуем свойства регулярных октодеревьев, развернутых для модельного набора данных.

Утверждение 1. Пусть регулярное октарное дерево глубиной $h \leq \lceil \log_2 \frac{1}{l} \rceil$ построено для модельного набора $S(n, l)$. Тогда математическое ожидание числа объектов в октантах i -го уровня и ниже ($1 \leq i \leq h$) есть

$$n_i = n \frac{1 - 2^{i-1}l^3}{(1-l)^3}$$

Утверждение 2. Пусть регулярное октарное дерево глубиной $h \leq \lceil \log_2 \frac{1}{l} \rceil$ построено для модельного набора $S(n, l)$, тогда математическое ожидание числа объектов в октанте i -го уровня есть

$$N_i = \begin{cases} n \frac{(1-2^{i-1}l)^3 - (1-2^i l)^3}{8^{i-1}(1-l)^3}, & \text{если } 1 \leq i < h; \\ n \frac{(1-2^{i-1}l)^3}{8^{i-1}(1-l)^3}, & \text{если } i = h. \end{cases}$$

Из приведенных утверждений следует следующая теорема об ожидаемой глубине регулярного октарного дерева.

Теорема 1. Ожидаемая глубина регулярного октарного дерева $Octree(m)$, построенного для модельного набора данных $S(n, l)$ с фактором пространственной наполненности $\rho(n, l)$, определяется следующим выражением:

$$h = \begin{cases} \lceil \log_2 \frac{2}{l+(1-l)\sqrt{\frac{m}{n}}} \rceil, & \text{если } m \geq \rho(n, l); \\ \lceil \log_2 \frac{1}{l} \rceil, & \text{если } m < \rho(n, l). \end{cases}$$

Анализ вычислительной сложности

Проанализируем вычислительные затраты на построение регулярного октарного дерева, его обновление, а также на выполнение запросов, связанных

с выборкой объектов по заданной пространственной области и определением столкновений. Воспользуемся описанным модельным набором данных и получим оценки сложности на всем диапазоне изменения параметров задачи.

Анализ сложности построения октодеревя.
Имеет место

Теорема 2. Трудоемкость построения регулярного октарного дерева $Octree(m)$, для модельного набора данных $S(n, l)$, выраженная в операциях определения принадлежности объекта одному из восьми дочерних октантов, равна $Q_{deploy} = q(n, m, l)n$, причем коэффициент $q(n, m, l)$ удовлетворяет следующим соотношениям:

$$q(n, m, l) = \lceil \frac{1}{3} \log_2 n - \frac{1}{3} \log_2 m \rceil \text{ при } l = 0;$$

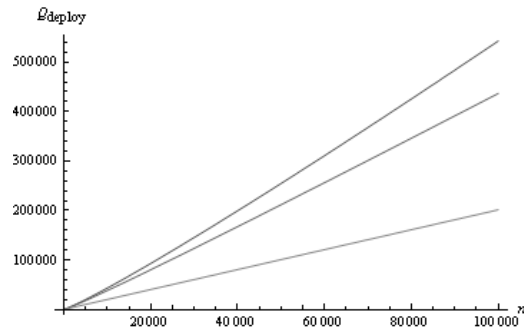
$$q(n, m, l) \leq \log_2 \frac{1}{l} \text{ при } l > 0.$$

Доказательство теоремы основывается на приведенных выше леммах и предположении, что дерево строится путем последовательного добавления объектов в сцену. Теорема имеет важное следствие, связанное с разным асимптотическим ростом сложности построения дерева для точечных и протяженных данных. Для первых ($l = 0$) сложность оценивается как $O(n \log_2 n)$, а для вторых ($l > 0$) — как $O(n)$. На рисунке 3 приведены графики сложности построения октодеревя от количества объектов в модельном наборе n при разных значениях порога наполненности октантов m , построенные для точечных (верхние кривые) и протяженных объектов (нижние кривые).

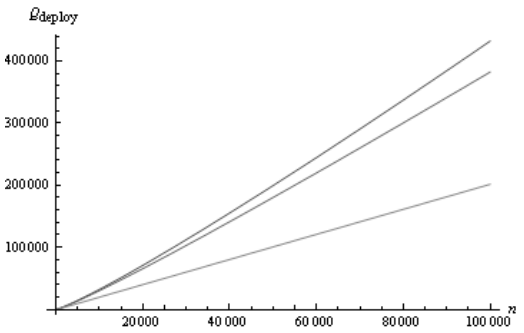
Примечательно, что в исследуемом диапазоне размерности задачи (до ста тысяч объектов) все кривые выглядят схожим образом, несмотря на наличие логарифмического фактора в асимптотической оценке сложности для точечных объектов. Согласно приведенным графикам сложность построения дерева для протяженных объектов несколько ниже, чем для точечных данных. Это объясняется тем обстоятельством, что при одинаковом общем количестве, часть объектов локализуется на более высоких уровнях октарного дерева и не возникает необходимости в их дальнейшем анализе.

Анализ сложности определения столкновений.

Перейдем к оценкам затрат на поиск столкновений объектов. Как правило, точное пересечение двух геометрических объектов сложной формы требует большого объема вычислений [2, 3]. Поэтому эффективной стратегией является предварительная локализация потенциальных столкновений с помощью простых тестов, основанных на сепарации



(a) $m = 10, l = 0, 0.01, 0.1$.



(b) $m = 100, l = 0, 0.01, 0.1$.

Рис. 3: Зависимость стоимости построения октарного дерева от количества объектов при различных габаритах объектов и параметра заполнения листовых октантов.

пространства и пересечении ограничивающих объемов [4, 7]. Точная процедура определения столкновений применяется лишь в случае положительного вердикта, что происходит в приложениях относительно редко и, как результат, существенно уменьшает общее время поиска столкновений. Эффект особенно ощутим для сцен, объекты которых геометрически представлены сложными аналитическими кривыми, поверхностями или полиэдрами с большим числом граней.

Предположим, что октарное дерево развернуто для модельного набора $S(n, l)$ и предстоит выявить все пары объектов, допускающие столкновения. В наивном алгоритме пришлось бы попарно пересечь все ограничивающие объемы объектов, что привело бы к выполнению $n(n-1)/2$ операций поиска пересечения между объектами. Однако, ввиду предварительного отсеивания большого количества априори непересекающихся объектов, поиск коллизий при помощи октарного дерева может быть выполнен гораздо эффективнее. Имеет место

Теорема 3. Пусть регулярное октарное дерево $Octree(m)$ глубиной h развернуто для модельного набора данных $S(n, l)$. Тогда вычислительная трудоемкость поиска столкновений, выраженная в операциях пересечения между ограничивающими

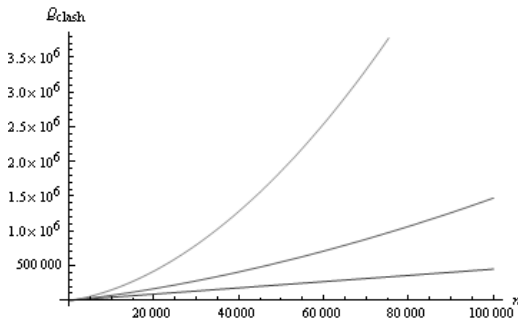
объемами, может оцениваться следующим образом:

$$Q_{clash} \leq \frac{mn}{2} + n^{\frac{4}{3}}m^{\frac{2}{3}}\left(\frac{45l}{2} + 186l^2\right) + 57l^2n^2 + o(l^2), l \rightarrow 0$$

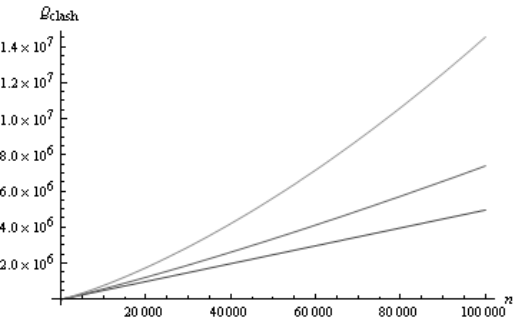
$$Q_{clash} \leq n^2, \forall l \in [0; 1]$$

где m и n ограничены.

Примечательно, что при небольших габаритах объектов, вычислительная трудоемкость поиска столкновений имеет субквадратичный рост, поскольку превалирующими оказываются первые члены суммы.



(a) при пороге заполнения октантов $m = 10$.



(b) при пороге заполнения октантов $m = 100$.

Рис. 4: График стоимости локализации столкновений в зависимости от количества объектов при различных значениях габаритов ($l = 0, l = 0.001, l = 0.003$).

На приведенных графиках (рис. 4) видно, что сложность локализации столкновений возрастает с увеличением числа объектов и их размеров. Нижние прямые на графиках соответствуют точечным данным, а средние и верхние кривые — протяженным объектам. Хотя анализ совпадений точечных данных имеет линейную сложность, а поиск столкновений протяженных объектов — квадратичную, для разреженных сцен различия оказываются не столь существенными даже для значительного числа объектов, составляющего на приведенных графиках 100000. Это служит обоснованием применимости метода в промышленных приложениях,

оперирующих большими данными и подверженных деградации производительности даже при использовании алгоритмов невысокой полиномиальной сложности.

Анализ сложности выборки объектов по заданной области.

Перейдем к оценкам затрат на выборку объектов, лежащих в заданной пространственной области. Для определенности будем считать, что область выборки представляет собой куб с относительным размером ребер L , приведенным к габаритам ограничивающего параллелепипеда всего набора данных. Определим характерный уровень локализации области выборки как $H = \lceil \log_2 \frac{1}{L} \rceil$. Тогда верно следующая оценка трудоемкости выборки объектов, выраженной в операциях пересечения ограничивающих объемов:

$$Q_{clip} = \begin{cases} n \frac{64 - 224 \cdot 2^h l + 336 \cdot 4^h l^2 + 8^h (11l - 81l^2 + (113 - 56h)l^3)}{8^h (1-l)^3} \\ + 8(h-1), \text{ если } h \leq H; \\ n \frac{64 - 224 \cdot 2^H l + 336 \cdot 4^H l^2 + 8^H (11l - 81l^2 + (113 - 56H)l^3)}{8^H (1-l)^3} \\ + \frac{8^h - H + 2 + 56H - 120}{7}, \text{ если } h > H. \end{cases}$$

На рисунке 5 представлены семейства графиков, иллюстрирующие зависимость стоимости выборки объектов от габаритов объектов и размера области выборки. Как видно, затраты растут с увеличением размера области выборки и уже при $L > 1/4$ использование октарного дерева теряет смысл. При меньших размерах применение октарного дерева оправданно, однако рост габаритов объектов может нивелировать суммарный эффект.

Анализ сложности поиска ближайших соседей.

Перейдем к оценке затрат на поиск ближайших соседей. Будем считать, что октарная структура уже развернута и для заданного объекта сцены требуется найти ближайшего соседа. Несложно показать, что в регулярном октарном дереве $Octree(m)$ глубины h , развернутом для модельного набора данных $S(n, l)$, ближайший к исходному объект не может находиться на расстоянии, превышающем размер диагонали листового октанта вдвое, т.е.

$$d_{min} < 2^{2-h} \sqrt{3}$$

Поэтому область поиска соседа всегда может быть ограничена кубом, ориентированным по главным координатным осям, с центром в исходном объекте и стороной $2^{2-h} \sqrt{3}$. Таким образом, в ходе работы алгоритма будут анализироваться лишь те октанты, которые имеют непустое пересечение с данной областью. Тогда в деревьях с глубиной, не превышающей четырех, анализу подлежат все октанты и все объекты сцены. В более глубоких деревьях на

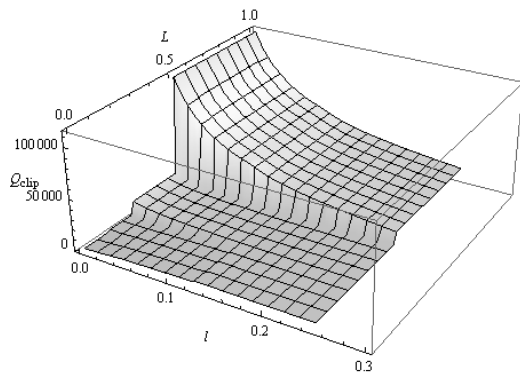
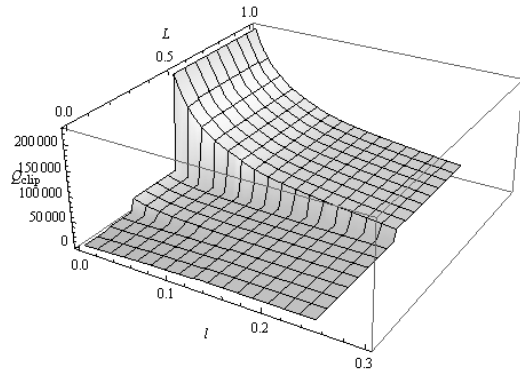
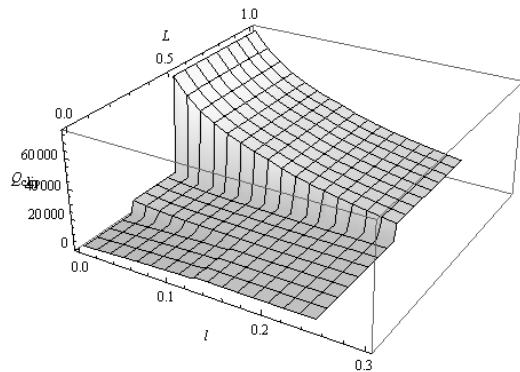
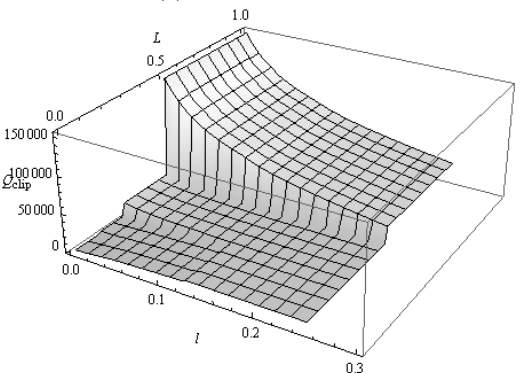
(a) $n = 5000, m = 5$.(b) $n = 10000, m = 5$ (c) $n = 5000, m = 10$ (d) $n = 10000, m = 10$

Рис. 5: Графики стоимости выборки объектов от габаритов и размера области.

верхних уровнях область поиска пересекает не более восьми октантов, а на нижних уровнях ($i \geq h - 2$) — не более $(|2^{i+2} - h\sqrt{3}| + 1)^3$. В каждом анализируемом октанте необходимо вычислить функцию расстояния для всех его объектов. Можно показать, что с учётом распределения объектов по уровням дерева при $h \geq 5$ вычислительная сложность поиска, выраженная в операциях определения расстояния между парой кубов, может быть оценена как:

$$Q_{neighbour} \leq \frac{1}{8^h(1-l)^3} (-4096n + 412^{4+h}ln + 213 \cdot 4^{1+h}l^2n - 8^h(816 + 8h(7l^3n - 8(1-l)^3)) + l(-2445 + 11n + l(2439 - 81n + l(-809 + 202n))))$$

Интерес вызывает тот факт, что в случае точечных данных ($l = 0$) стоимость поиска оценивается как $Q_{neighbour} \leq 4096m + 64 \log_8 \frac{n}{m} + 816$, что согласуется с известной асимптотической оценкой $O(\log n)$.

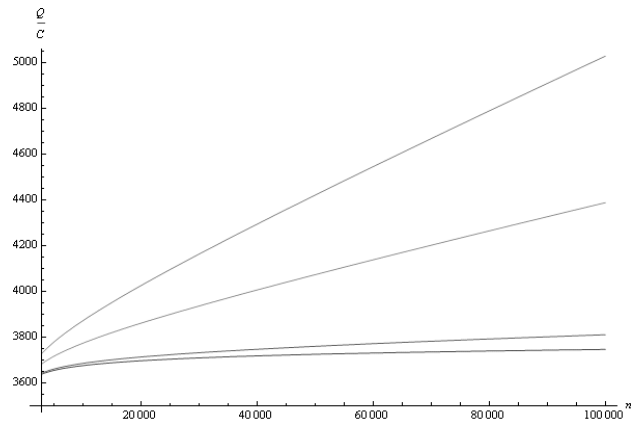


Рис. 6: Вычислительная стоимость поиска соседей в зависимости от числа и габаритов объектов.

На рисунке 6 представлено семейство кривых, отражающих зависимость стоимости поиска ближайшего соседа от числа объектов в сцене и их габаритов при $l = 0, 0.00005, 0.0005, 0.001$. Нижние кривые соответствуют меньшим значениям габаритов. Как видно, в случае точечных данных стоимость поиска имеет логарифмический рост, а с увеличением габаритов объектов проявляет линейный характер.

Результаты вычислительных экспериментов.

Для подтверждения результатов теоретического анализа были проведены серии вычислительных экспериментов с модельными наборами данных. Каждая серия экспериментов предназначалась для

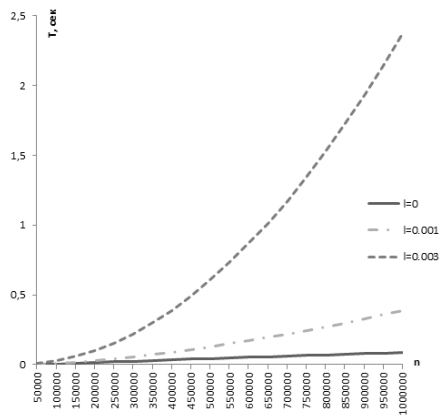
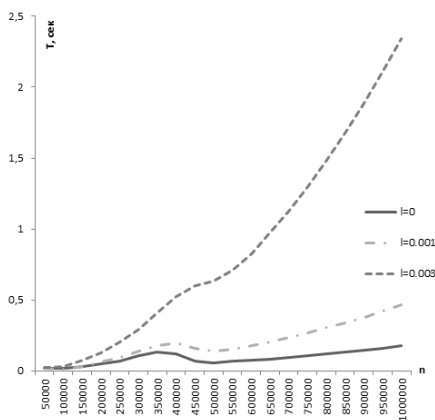
(a) $m = 10$.(b) $m = 100$.

Рис. 7: Зависимость времени локализации столкновений от количества объектов в сцене при различных габаритах и пороге наполненности октантов.

получения практических оценок сложности для исследованных операций, включая построение и обновление пространственных индексов на основе регулярного октодеревя, локализация столкновений в сцене, выборка объектов по заданной пространственной области и поиск ближайших соседей. Для достоверности анализа случайным образом генерировалось множество модельных наборов с идентичными значениями параметров, а затем результаты усреднялись. Эксперименты выполнялись на персональном компьютере с типовой конфигурацией: процессор Core i7 3770 3.4GHz, 16 GB RAM (1600MHz).

Для краткости ограничимся анализом результатов, связанных с локализацией столкновений в сцене. Как видно из графиков 7, порог наполненности октантов существенно не влияет на время разре-

шения запроса, однако может приводить к немонотонному росту затрат с увеличением числа объектов. Это объясняется тем, что в октантном дереве появляется еще один уровень и время разрешения запросов претерпевает скачкообразное изменение.

Другим важным результатом является существенная зависимость затрат от числа объектов в сцене и их габаритов. Однако в случае слабо заполненных сцен можно рассчитывать на оптимистическую оценку трудоемкости пространственных запросов, характерную для точечных данных. С ростом плотности пространственного заполнения сцены затраты на локализацию столкновений неизбежно растут и в предельном случае совпадают с упомянутой пессимистической оценкой, соответствующей наивному алгоритму. Накладные расходы на обновление индексов остаются относительно низкими во всех практически содержательных случаях.

Все полученные экспериментальные данные хорошо согласуются с теоретическими результатами и качественно подтверждают выводы о перспективности использования регулярных октодеревьев для пространственного индексирования сложных динамических сцен.

Выводы

Таким образом, рассмотрено важное семейство методов декомпозиции, основанное на регулярных октантных деревьях и обеспечивающее эффективную индексацию, быстрый поиск и анализ больших пространственных данных. Для алгоритмов определения столкновений и выборки объектов по заданной области получены оценки сложности в среднем, которые улучшают известные пессимистические результаты и служат теоретическим обоснованием их применения к более широкому классу приложений.

Литература

- [1] *Cai M., Revesz P.* Parametric rectangles: an index structure for moving objects // Proceedings of the 10th COMAD International conference on management of data, 2000. – pp. 57-64.
- [2] *Gottschalk S., Lin M. C., Manocha D.* OBB Tree: a hierarchical structure for rapid interference detection // Proceedings of the SIGGRAPH'96 Conference, 1996. –pp. 171-180.
- [3] *Gottschalk S.* Collision queries using oriented bounding boxes Chapel Hill: The University of North Carolina, 2000.
- [4] *Jimenez P., Thomas F., Torras C.* 3D collision detection: a survey // Computers and Graphics. – 2001. –Vol.25, – pp.269-285.
- [5] *Kedem G.* The quad-CIF tree: a data structure for hierarchical on-line algorithms // Proceedings of the 19th Design Automation Conference, 1992. –pp. 352-357

- [6] *Semenov V.A., Kazakov K.A., Morozov S.V., Tarlapan O.A., Zolotov V.A., Degenis T.* 4D modeling of large industrial projects using spatiotemporal decomposition // *eWork and eBusiness in Architecture, Engineering and Construction*, 2010. –pp. 89-95
- [7] *Semenov V.A., Kazakov K.A., Zolotov V.A., Jones H., Jones S.* Combined strategy for efficient collision detection in 4D planning applications // *Computing in Civil and Building Engineering*, 2010. –pp. 31-39
- [8] *Золотов В.А., Семенов В.А.* Современные методы поиска и индексации многомерных данных в приложениях моделирования больших динамических сцен // *Труды Института системного программирования*. –2013. –Т.24 –С. 381-416

Метод сжатия полигональных сеток с потерями*

А.А. Зачесов

azachesov@artec-group.com

Институт Прикладной Математики им. М.В. Келдыша, Москва, Россия

С развитием и повсеместным распространением технологий трехмерного сканирования происходит экспоненциальный рост объема получаемых в результате сканирования данных – главным образом, трехмерных сеток, хранящих геометрические данные снимавшегося объекта. Становится актуальной проблема эффективного хранения этих данных с минимальными потерями или без потерь их точности. Предлагается алгоритм сжатия с потерями для трехмерных полигональных сеток, полученных методом трехмерного сканирования. Описываются характеристики этого алгоритма. Приводится сравнение объема данных, получаемых в результате применения предложенного алгоритма, с популярными форматами для хранения трехмерных сеток.

Ключевые слова: Сжатие с потерями, трехмерная модель, трехмерная сетка, квантование

Lossy compression method for polygonal meshes*

A.A. Zachesov

Keldysh Institute of Applied Mathematics, Moscow, Russia

Development and spreading of three-dimensional scanning technologies result in exponential growth of scanned data volumes – mainly three-dimensional grids that store geometry information. It becomes an actual problem to efficiently store this data with minimal or without precision loss. We propose a lossy compression algorithm for three-dimensional polygon meshes obtained by three-dimensional scanning. We describe algorithms' characteristics and show the comparison of the compression ratio for the proposed algorithm and popular file formats for storing three-dimensional meshes.

Keywords: Lossy compression, three-dimensional mesh, quantization

Все способы получения трехмерных моделей можно разделить на две группы: моделирование, когда создание геометрии модели происходит в полуавтоматическом режиме, и реконструкция, когда построение геометрии осуществляется автоматически. Это может быть реконструкция трехмерной модели из многоракурсного видео, реконструкция по структурированной подсветке или реконструкция из видеоряда с одной камеры. Любой из этих методов осуществляет построение трехмерной сетки с некоторой погрешностью относительно реальных размеров исходных объектов.

Данные, получаемые в результате трехмерного сканирования, как правило, являются промежуточными перед дальнейшим этапом обработки. Это означает, что точность промежуточных данных влияет на качество этой дальнейшей обработки, из-за чего для хранения данных, как правило, используются форматы, предполагающие хранение данных без потерь. Это приводит к значительному объему хранимых промежуточных данных. При этом, как говорилось выше, данные изначально содержат погрешность, которой можно пренебречь. Это позволяет создать алгоритм сжатия данных с потерями, позволяющий существенно уменьшить объем хранимых данных.

Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347

Обзор существующих методов

Существует значительное количество методов, осуществляющих сжатие произвольных полигональных сеток с потерями [2, 3]. Их можно разделить на две большие группы: методы, использующие упрощение исходной сетки – уменьшение количества полигонов – и методы, сохраняющие исходную структуру сетки (количество вершин и треугольников без изменений). Методы из первой группы позволяют достичь низких коэффициентов сжатия, отбрасывая часть информации об исходных полигонах. В [2] уменьшается количество полигонов трехмерной сетки путем объединения нескольких полигонов в один. Также описывается подход, основанный на перестраивании геометрической структуры модели с целью получения равномерной сетки.

При втором подходе структура (количество вершин и треугольников) трехмерной сетки остается неизменной. В [5] используется квантование координат, а также замена значений координат их разностью. Для квантования координат используется фиксированное число бит. В [1] для кодирования значений разности координат используются коды Хаффмана. Для представления информации о треугольниках исходной трехмерной сетки чаще всего используется информация об их связности [4]. При использовании такого подхода для сохранения информации об одном треугольнике достаточно 2 бит.

Предложенный алгоритм

Предлагается алгоритм сжатия с потерями для трехмерной полигональной сетки, описываемой совокупностью массива вершин и объединяющих их треугольников. Каждая вершина описывается тремя вещественными координатами, а каждый треугольник – тремя целочисленными индексами вершин. Выходными данными алгоритма является сжатое представление исходной сетки, которое далее также будет именоваться выходным потоком. Исходные данные можно также представить как последовательность вершин и последовательность треугольников, которые объединяют эти вершины. Рассмотрим обработку этих наборов данных отдельно.

Обработка последовательности координат вершин

Рассмотрим одну из вершин последовательности, например, первую. Координаты этой вершины по каждой из осей могут значительно отличаться друг от друга по абсолютным значениям. В то же время значения координат по каждой из осей будут близки к значениям координат соседней вершины по той же оси. Будем обрабатывать последовательности координат по каждой из осей независимо, а для определенности в дальнейшем рассматривать сжатие последовательности координат по оси x . Для последовательностей координат по остальным осям обработка аналогична.

Основной идеей алгоритма сжатия последовательности координат, как и в методе, описанном в [5], является замена значений координат значениями разницы между соседними координатами. Сжатие достигается за счет квантования значений разницы и использования для них меньшего количества бит, чем для значения исходной координаты. Для каждой вершины в выходной поток записывается значение $\Delta = x_n - x_{n-1}$ с заданной точностью. В отличие от сжатия изображений или видео, диапазон значений Δ не ограничен, поскольку абсолютные значения исходной сетки не ограничены, кроме того, сама исходная сетка может содержать в себе разрывы. Для ограничения диапазона изменения значений разницы координат вводится параметр максимальной кодируемой разницы, или порога квантования, задаваемый однократно для всей последовательности координат по одной оси. В рамках этого диапазона для значений разницы используется операция квантования по уровню с заданным количеством бит. Для каждой координаты ее значение в выходном потоке аппроксимируется, как:

$$x_n \cong x'_n = x'_{n-1} + \Delta. \quad (1)$$

где x'_{n-1} – значение координаты для предыдущей вершины после применения к ней аналогичной ап-

проксимации. Для первой координаты $x_0 = x'_0$, и ее значение помещается в выходной поток «как есть». Значение Δ есть аппроксимация исходной разницы между $x(n-1)'$ и x_n , вычисляемое как:

$$\Delta = \left(\frac{x'_n - x'_{n-1}}{Q} \cdot 2^p \right). \quad (2)$$

где Q – порог квантования, p – количество уровней квантования, которые являются параметрами метода.

Использование координат после их аппроксимации в качестве базовых для кодирования следующих координат в последовательности необходимо для корректной распаковки. Если разница текущей рассматриваемой координаты с предыдущей координатой больше заданного порога квантования, текущая координата помещается в выходной поток «как есть». Такие вершины назовем ключевыми. Вся последовательность исходных координат можно разбить на набор независимых последовательностей, состоящих из ключевой вершины и произвольного количества координат, закодированных разницей.

Оценка коэффициентов сжатия для последовательности координат

Для оценки коэффициентов сжатия будем считать, что для сохранения вещественного значения координаты необходимо 4 байта. Коэффициенты сжатия будем рассматривать только для последовательности координат по одной из осей. Лучший коэффициент сжатия достигается для последовательности координат, в которой все расстояния между соседними вершинами меньше, чем выбранный порог квантования (то есть ключевая координата одна), а для квантования используется два уровня, то есть для сохранения значения разницы достаточно одного бита. Пусть размер исходных данных составляет $32 \cdot n$ бит, где n – количество координат в последовательности. Тогда объем сжатых данных, пренебрегая объемом, необходимым для сохранения ключевой вершины, составит $2 \cdot n$ бит, коэффициент сжатия – 6,25%

Худший коэффициент сжатия достигается для последовательности координат, все расстояния между соседними вершинами которой больше, чем выбранный порог квантования (то есть все координаты являются ключевыми). В этом случае объем увеличится на один бит для каждой ключевой вершины, то есть станет равным $33 \cdot n$ бит, коэффициент сжатия – 103,125%.

Алгоритм выбора оптимальных параметров сжатия

Степень сжатия зависит от выбора параметров сжатия: порога квантования и количества уровней квантования, причем максимальная степень

сжатия достигается, если порог квантования выбрать неограниченно большим, а количество уровней квантования ограничить двумя. Однако это приведет к значительным потерям точности исходных данных, что сводит на нет полезность сжатия в целом. Выбор некоторых универсальных параметров невозможен, поскольку характер исходных данных может сильно варьироваться. Для контроля степени сжатия предлагается метод автоматического подбора оптимальных параметров (при которых размер сжатого представления исходных данных минимизируется) с ограничением погрешности значений вершин после сжатия относительно исходных значений.

Размер данных после сжатия всегда определяется количеством ключевых вершин и количеством уровней квантования, используемых для кодирования значения разницы вершин. Оптимальными параметрами сжатия являются такие Q и p , при которых размер файла S минимален. Функция для минимизации выглядит следующим образом:

$$S = keys \cdot 33 + (n - keys) \cdot (p + 1). \quad (3)$$

Здесь $keys$ – количество ключевых координат, n – общее количество координат, p – используемое количество уровней квантования. Прибавление единицы в $p + 1$ позволяет учесть влияние бита-флага на объем файла. По той же причине используется множитель $33 = 32 + 1$. Количество ключевых вершин $keys$ может быть вычислено для любого фиксированного значения порога квантования Q . При добавлении ограничения на допустимую ошибку относительно исходных значений появляется связь между порогом квантования и количеством необходимых уровней квантования

$$p = \lceil \log_2 \left(\frac{Q}{e} \right) \rceil. \quad (4)$$

где e – значение допустимой погрешности. Действительно, чтобы удовлетворить требование о допустимой погрешности необходимо и достаточно, чтобы интервал, получаемый после квантования имел длину меньше e . Отсюда следует, что при увеличении Q растет p , и наоборот, а также что для минимизации функции достаточно осуществить перебор по Q .

Обработка последовательности индексов треугольников

Кроме координат вершин для восстановления трехмерной сетки необходима также информация о соединяющих их треугольниках. Как правило, для каждого треугольника сохраняются три индекса вершин, которые составляют этот треугольник, то есть исходными данными является последовательность троек целочисленных индексов. Эта последовательность построена таким образом, чтобы соседние тройки в последовательности соответствовали соседним в пространстве треугольникам и имели общие индексы вершин. Такая очередность позволяет построить алгоритм генерации последовательности на основе рекуррентных формул, использующим зависимости между значениями индексов.

В силу наличия разрывов в исходной трехмерной сетке невозможно закодировать всю последовательность треугольников, используя единственную рекуррентную формулу. Вид зависимостей индексов также может отличаться. Для борьбы с этим алгоритм использует несколько предзаданных рекуррентных формул вместо одной, а также вводится понятие «ключевой тройки индексов» – тройки, которая не может быть предсказана из предыдущей при использовании любой рекуррентной формулы. Для каждой ключевой тройки существует подпоследовательность троек, выводимых из нее применением одной и той же рекуррентной формулы (длина этой последовательности может быть нулевой). Каждая подпоследовательность сохраняется в выходной поток следующим образом: ключевая тройка индексов, количество выводимых из нее других троек и код используемой для восстановления рекуррентной формулы.

В силу наличия разрывов в исходной трехмерной сетке невозможно закодировать всю последовательность треугольников, используя единственную рекуррентную формулу. Вид зависимостей индексов также может отличаться. Для борьбы с этим алгоритм использует несколько предзаданных рекуррентных формул вместо одной, а также вводится понятие «ключевой тройки индексов» – тройки, которая не может быть предсказана из предыдущей при использовании любой рекуррентной формулы. Для каждой ключевой тройки существует подпоследовательность троек, выводимых из нее применением одной и той же рекуррентной формулы (длина этой последовательности может быть нулевой). Каждая подпоследовательность сохраняется в выходной поток следующим образом: ключевая тройка индексов, количество выводимых из нее других троек и код используемой для восстановления рекуррентной формулы.

Пример рекуррентной формулы, отражающей зависимости между индексами вершин соседних треугольников

$$i_n = i_{n-1}; j_n = j_{n-1} - 1; k_n = j_{n-1}; n = 2m$$

$$i_n = i_{n-1} + 1; j_n = j_{n-1} + 2; k_n = i_{n-1} + 2; n = 2m + 1$$

где i, j, k – индексы вершин треугольников. Данная рекуррентная формула выведена эмпирическим путем при анализе исходных трехмерных сеток. Всего используется 4 рекуррентные формулы аналогичного вида.

Оценка коэффициентов сжатия для последовательности индексов треугольников

Рассмотрим лучший и худший коэффициенты сжатия для последовательности троек индексов. Будем считать, что каждая тройка индексов требует 12 байт (три целочисленных значения). Длину последовательности будем кодировать двумя байтами. Для поддержки возможного увеличения количества используемых рекуррентных формул, будем считать, что для сохранения типа формулы необходим 1 байт. Лучший коэффициент достигается, если все треугольники исходной последовательности предсказываются с использованием одной из рекуррентных формул. В этом случае одна подпоследовательность из 2^{16} троек индексов – 768 кбайт

Таблица 1: Сравнение объема данных

Используемый формат данных	Объем данных относительно исходного
Object file формат (текстовый)	240%
Ply формат (бинарный)	102%
Ply формат + LZMA	48,8%
Метод, описанный в [1]	17%
Предложенный алгоритм (допустимая погрешность 0.1 мм)	13,7%
Предложенный алгоритм (допустимая погрешность 2 мм)	8,9%
Предложенный алгоритм + LZMA (допустимая погрешность 2 мм)	3,4%

– может быть представлена как 15 байт (исходная ключевая тройка индексов, длина последовательности и тип рекуррентной формулы). Коэффициент сжатия составляет 0,002%. Худший коэффициент достигается в случае, если исходная трехмерная сетка представляет собой набор несвязанных друг с другом треугольников. В этом случае каждая тройка попадает в выходной поток как есть, и коэффициент сжатия равен 125% (каждая тройка занимает 15 байт вместо исходных 12). Среднее количество предсказанных индексов треугольников для нескольких сотен исходных сеток, использованных для тестирования алгоритма, составляет 98% от общего числа, то есть средний коэффициент сжатия последовательности индексов треугольников составляет 2,5%.

Дальнейшее увеличение степени сжатия

Размер блока данных, получаемый в результате сжатия предложенным алгоритмом, может быть дополнительно уменьшен применением к нему алгоритма сжатия без потерь, например, алгоритма LZMA. Действительно, при сжатии координат в случае, если значения соседних координат отличаются друг от друга менее чем на $\frac{\epsilon}{2}$, где ϵ – допустимая погрешность координат, индекс квантования, используемый для их сохранения, будет одинаков. Наличие в выходном потоке подобных повторов позволяет уменьшить объем файла.

Выводы

В таблице 1 приведено сравнение объема данных, получаемых при сохранении трехмерной сетки в широко распространенные форматы с предложенным алгоритмом. За 100% принят объем данных,

рассчитанный тем же способом, который использовался для оценки лучшего и худшего коэффициентов сжатия. Объем, занимаемый координатами вершин и индексами треугольников, суммируется. Значения в таблице являются усредненными по нескольким сотням трехмерных сеток, использованных для тестирования алгоритма. В таблице приведены результаты для сжатия с несколькими различными значениями погрешности (значения координат заданы в миллиметрах).

Представленный алгоритм используется для хранения и передачи по сети данных, полученных трехмерным сканированием. Используется сжатие с допустимой потерей точности 0,1 мм при точности исходных отсканированных данных 2 мм. В соответствии с таблицей это позволяет уменьшить объем хранимых данных в 7-8 раз. Новизна и отличие предложенного метода относительно существующих методов состоит в использовании рекуррентной формулы для восстановления последовательности треугольников, а также использование автоматического выбора параметров при сжатии последовательности вершин.

Описанный алгоритм сжатия трехмерных сеток не поддерживает обработку сеток, которые имеют привязанную к ним текстуру, а именно сохранение информации о текстурных координатах. Дальнейшее развитие алгоритма состоит в добавлении поддержки текстурных координат. Они могут быть сжаты аналогично геометрическим координатам. В качестве допустимой погрешности может использоваться точность значения координаты, измеряемая в пикселях исходного файла текстуры.

Литература

- [1] *Deering M.* Geometry compression // Proceedings of SIGGRAPH Conference, 1995. – pp.13-20.
- [2] *Gotsman C., Gumhold S., Kobbelt L.* Simplification and Compression of 3D Meshes // Springer. Tutorials on Multiresolution in Geometric Modelling. – 2002. – С.319-361.
- [3] *Rossignac J.* 3D mesh compression // Visualization Handbook. – 2003. – С.359-379.
- [4] *Rossignac J.* Edgebreaker: Connectivity Compression for Triangle Meshes // IEEE Trans. Visualization and Computer Graphics, 1999. – vol. 5, no. 1 – pp.47-61.
- [5] *Touma C., Gotsman C.* Triangle Mesh Compression // Proceedings of Graphics Interface, 1998.

Адаптация систем научной визуализации к сторонним решателям*

К.В. Рябинин, С.И. Чуприна

kostya.ryabinin@gmail.com | chuprinas@inbox.ru

Пермь, Россия, Пермский государственный национальный исследовательский университет

В статье описан метод унифицированного решения задачи адаптации систем научной визуализации к сторонним решателям. Метод обеспечивает автоматизированную конвертацию данных, генерируемых решателем, в пригодный для рендеринга формат, а также автоматическую адаптацию систем визуализации к особенностям программно-аппаратных платформ и высокое качество результирующих изображений. Настройка на сторонний решатель осуществляется при помощи методов онтологического инжиниринга. По онтологическому описанию синтаксиса конструкций ввода-вывода различных языков программирования автоматически генерируется синтаксический анализатор для обработки и визуализации структуры входных и выходных данных из исходного кода решателя. При помощи высокоуровневого графического интерфейса пользователь сопоставляет элементы этой структуры со свойствами визуальных объектов, которые также описаны в виде онтологии и хранятся в базе знаний системы. На основе сделанных настроек автоматически генерируется конвертер данных решателя в описание графической сцены, по которому затем строится итоговое изображение. Адаптация к особенностям различных программно-аппаратных платформ осуществляется посредством автоматической генерации графического интерфейса пользователя под каждую конкретную платформу, а также посредством автоматической предобработки подлежащих визуализации данных, которая основана на расширяемом наборе эвристических правил. Благодаря этому оказывается возможной работа системы визуализации не только на настольных компьютерах, но и на мобильных устройствах. Высокое качество рендеринга обеспечивается поддержкой оригинального адаптивного алгоритма сглаживания ступенчатых границ объектов на изображении. На основе предложенного подхода разработана мультиплатформенная система научной визуализации SciVi, которая успешно применена для решения ряда научно-практических задач из различных предметных областей.

Ключевые слова: научная визуализация, онтологический инжиниринг, мультиплатформенность, мобильные устройства, адаптивное сглаживание границы.

Scientific visualization systems adaptation to third-party solvers*

K.V. Ryabinin, S.I. Chuprina

Perm State University, Perm, Russia

In this paper the new unified method of scientific visualization systems adaptation to the third-party solvers is described. This method ensures automatic conversion of the solvers generated data into the renderable format, automatic adaptation of the visualization systems to the specifics of platforms they run on and high quality of the result images. The adaptation to the third-party solvers is based on the ontological engineering. It is proposed to create knowledge base as a part of scientific visualization systems. This knowledge base contains two ontologies: syntax ontology that describes input and output statements of programming languages and visual objects ontology that describes graphical scenes and objects with their properties. The parser for solvers' source code is created automatically based on the syntax ontology to extract input and output data structures to be visualized. The user maps the elements of these data structures to the visual objects' properties obtained from the visual objects ontology. The system then automatically generates the data converter from the solvers' data format into the graphical scene description based on the settings done. According to the scene description the final image is automatically rendered. The adaptation to the platform specifics consists of the automatic graphical user interface generation as well as the data preprocessing based on heuristics to make the data meet the physical limits of particular platform. Thanks to this it is possible to support both desktop computers and mobile devices. High quality of rendering is ensured by the new adaptive anti-aliasing algorithm proposed. Based on the proposed approach multiplatform scientific visualization system SciVi is created and used to solve real-world tasks.

Keywords: scientific visualization, ontological engineering, multiplatform portability, mobile devices, adaptive anti-aliasing.

Введение

В настоящее время практически любая научная задача сводится к измерению параметров реаль-

ных объектов или моделированию протекающих в них процессов. В обоих случаях имеют место аппаратно-программные системы, называемые решателями, которые выполняют расчёт на основе модели, лежащей в основе научного эксперимента,

Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347.

и в ходе вычислений генерируют данные в некотором формате. Как правило, если эти данные предназначены для последующего анализа людьми, их требуется представлять в наглядном виде (визуализировать). Качественное графическое отображение научных данных позволяет исследователям лучше понять суть описываемых этими данными объектов и явлений, выявить скрытые закономерности и особенности [2]. Кроме того, в случае необходимости становится возможным дальнейшее исследование и усовершенствование лежащей в основе решателя модели, так как системы научной визуализации помогают обнаружить её недочёты, исходя из анализа результатов экспериментов.

Однако форматы выходных данных, также как и интерфейсы, у различных решателей в значительной степени отличаются друг от друга из-за высокого разнообразия объектов исследования в разных научных областях. Зачастую эти различия затрудняют использование одного визуализатора для нескольких решателей даже в рамках одной предметной области. В таких случаях приходится либо разрабатывать визуализаторы для решателей с нуля, либо создавать конвертеры данных из форматов решателей в форматы визуализаторов. Возникает острая необходимость разрабатывать системы визуализации таким образом, чтобы их можно было легко адаптировать к специфике различных решателей, поскольку это способно снизить трудозатраты на разработку промежуточного программного обеспечения и увеличить эффективность работы исследователей [3].

Часто решатели требуют для своей работы большой вычислительной мощности или специализированных устройств ввода/вывода информации, например, магнитно-резонансных томографов или секвенаторов ДНК. В этом случае они выполняются на суперкомпьютерах или на специальных аппаратных установках, и визуализаторы не могут быть их неотъемлемой частью. Чтобы в наглядном виде отображать данные, полученные от решателей такого рода, может быть использована клиент-серверная архитектура построения систем визуализации. Клиент может работать на настольном компьютере или даже на мобильном устройстве пользователя и связываться с решателем по локальной сети или сети Интернет. При этом, однако, система визуализации должна автоматически учитывать особенности инфраструктуры сети и операционных систем, чтобы обеспечить эффективную работу в различных (в частности, динамически меняющихся) условиях [3].

Многие научные задачи требуют трёхмерной, или даже многомерной визуализации [1]. Поскольку визуализация основана на растеризации, являющейся процессом дискретизации, возникает проблема ступенчатости границ объектов на изображении.

Чтобы улучшить визуальное качество и реалистичность изображения, необходимо производить сглаживание границ. Для этого существует большое количество алгоритмов, однако их реализация на мобильных платформах сталкивается с целым рядом нерешённых проблем. Например, проблемы снижения производительности визуализации и нежелательного размытия мелких деталей объектов на изображении до сих пор не имеют своего унифицированного решения.

Целью данной работы является описание принципов построения практически значимых систем научной визуализации, обеспечивающих:

1. Автоматизированную адаптацию к специфике сторонних решателей.
2. Автоматическую адаптацию к инфраструктуре программно-аппаратного обеспечения.
3. Высокое качество рендеринга.

На сегодняшний день отсутствуют какие-либо популярные и широко используемые системы научной визуализации, которые бы удовлетворяли всем вышеуказанным требованиям одновременно [3].

Предлагаемая архитектура систем научной визуализации

Предлагаемая архитектура систем научной визуализации на примере системы SciVi, разработанной авторами совместно с ИТ-компанией ООО «Ньюлана» (г. Пермь), представлена на рис. 1.

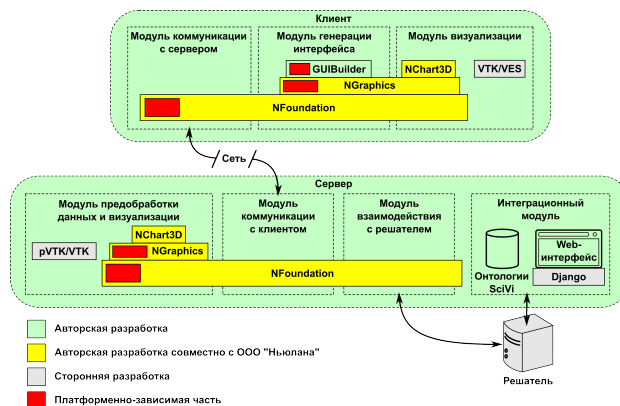


Рис. 1: Архитектура системы научной визуализации SciVi.

Для того чтобы в едином комплексе удовлетворить сформулированным требованиям, предлагается использовать клиент-серверную архитектуру систем научной визуализации и включить в состав этой архитектуры такие важные элементы, как

1. Интеграционный модуль, отвечающий за настройку на сторонние решатели, автоматически генерирующий декларативные описания шаблона графической сцены и графического интерфейса пользователя с решателем.

2. Модуль предобработки данных, отвечающий за преобразование визуализируемых данных к виду, пригодному для отображения на стороне клиента с учётом его программно-аппаратных особенностей.
3. Модуль генерации интерфейса, отвечающий за автоматическое построение графического интерфейса пользователя по его декларативному описанию.
4. Модуль визуализации, отображающий декларативно описанную графическую сцену (в качестве языка для декларативных описаний используется XML).

Онтологический инжиниринг как основа адаптивности систем научной визуализации

Для организации высокоуровневой настройки системы научной визуализации на специфику сторонних решателей предлагается использовать методы онтологического инжиниринга. Онтологический инжиниринг – это методологии обработки и хранения знаний на основе онтологий. Онтология O – это формальная модель предметной области, включающая в себя множество понятий T этой предметной области с их определениями, множество связей R между понятиями и множество аксиом A , $O = \langle T, R, A \rangle$.

В данной работе выдвинута и подтверждена на практике гипотеза о том, что для решения задачи адаптации системы научной визуализации к сторонним решателям достаточно использовать прикладные онтологии без аксиоматики: $A = \emptyset$.

Процесс адаптации осуществляется при помощи двух онтологий: онтологии конструкций ввода-вывода языков программирования L и онтологии визуальных объектов U .

Онтология L описывает синтаксис конструкций объявления переменных и конструкций ввода-вывода различных языков программирования. В настоящее время реализована поддержка языков C/C++, Fortran и Java; трудоёмкость добавления поддержки нового языка сравнима с написанием БНФ для соответствующих синтаксических конструкций этого языка. На основе онтологии L автоматически генерируется синтаксический анализатор, который посредством регулярных выражений разбирает исходный код решателя и вычленяет описание всех переменных, участвующих в операциях ввода-вывода. Пример результата работы анализатора представлен на рис. 2. Если по тем или иным причинам исходный код решателя не может быть предоставлен, пользователю предлагается описать входные и выходные данные вручную, используя для этого высокоуровневый графический интерфейс.

Онтология U описывает внешний вид и визуальные характеристики объектов (например, цвет, положение в пространстве и т.п.), которые могут быть использованы для построения графической сцены с целью визуализации результатов работы решателя. Пользователю предлагается при помощи высокоуровневого графического интерфейса выбрать визуальные объекты, подходящие для его задачи, и связать их характеристики с выходными и, при необходимости, входными, данными решателя. Этот процесс схематично представлен на рис. 3.

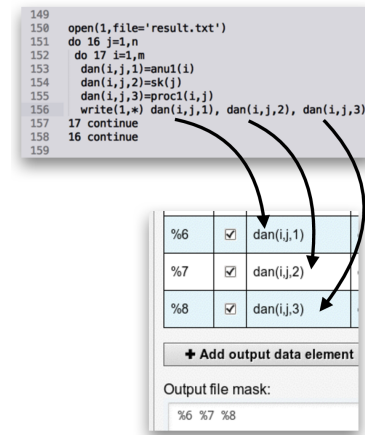


Рис. 2: Результат синтаксического разбора исходного кода решателя в системе SciVi.

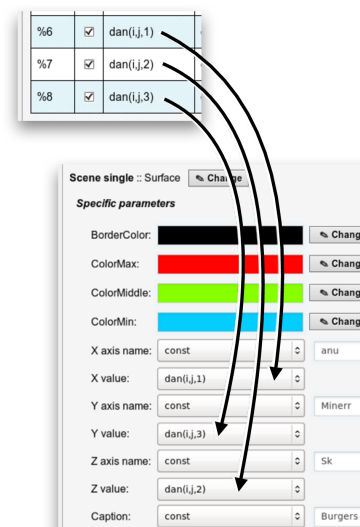


Рис. 3: Результат сопоставления данных решателя с визуальными характеристиками объектов сцены в системе SciVi.

После выполнения указанных настроек автоматически генерируется шаблон графической сцены в формате XML, содержащий описания всех выбранных пользователем объектов. Вместо значений их

характеристик в XML-файле указаны ссылки на соответствующие фрагменты маски разбора входного и выходного файлов решателя. Затем эти значения автоматически замещаются данными, извлекаемыми из указанных файлов. Примеры визуализации различных сцен в системе SciVi представлены на рис. 4.

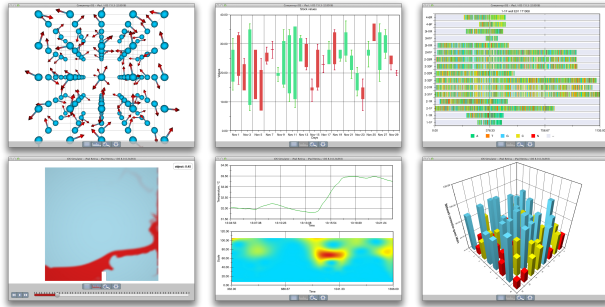


Рис. 4: Результат визуализации в системе SciVi.

На этапе настройки графической сцены пользователь также может указать, какие из входных данных решателя он хотел бы изменять в процессе визуализации. Для таких данных также автоматически генерируется описание соответствующих элементов графического интерфейса, которые встраиваются в итоговый интерфейс визуализатора.

Работа с решателем через сгенерированный интерфейс возможна только при наличии исполняемого файла решателя и возможности запустить его на стороне сервера (при наличии всех необходимых ресурсов, таких, как достаточный объём памяти, динамические библиотеки и пр.). В этом случае данные для визуализации порождаются динамически в процессе работы решателя («режим on-line»). Если исполняемый файл решателя недоступен, остаётся возможность визуализации данных непосредственно из файла, хранящего результаты работы решателя («режим off-line»).

Таким образом, работа с системой визуализации делится на 2 этапа: этап настройки на решатель пользователя и визуализируемые данные (регистрация решателя в системе, выбор параметров визуализации) и этап построения графической сцены (в on-line или off-line режиме).

Автоматический учёт особенностей различных платформ

Важным требованием к системе научной визуализации является эффективная мультиплатформенность, позволяющая автоматически настраиваться на особенности инфраструктуры программно-аппаратного обеспечения для достижения максимальной скорости визуализации и эргономики работы.

Абстрагирование от конкретной платформы достигается использованием библиотек NFoundation и NGraphics, разработанных в соавторстве с компанией ООО «Ньюлана». Библиотека NFoundation обеспечивает уровень абстрагирования от ОС. По функциональности она схожа с библиотекой Boost и включает в себя все необходимые средства для работы с динамической памятью, ресурсами, сетью, потоками, функциями обратного вызова и делегатами, контейнерами, математическими функциями и оптимизационными алгоритмами. Библиотека NGraphics обеспечивает уровень абстрагирования от низкоуровневого графического API и поддерживает работу с OpenGL, OpenGLES, DirectX9 и DirectX11. NGraphics используется для отображения сложных трёхмерных объектов, а также двумерных элементов графического интерфейса пользователя.

Графический интерфейс пользователя автоматически генерируется библиотекой GUIBuilder для каждой конкретной платформы по единому высокоуровневому декларативному описанию на языке XML. Описание интерфейса составлено из высокоуровневых понятий, таких как «кнопка», «ползунок», «текстовая метка» и т. д. Внешний вид и модель взаимодействия пользователя с каждым элементом управления для каждой конкретной ОС извлекаются из встроенной базы данных, за счёт чего нивелируются различия в парадигмах управления на настольных компьютерах и мобильных устройствах. Библиотека NGraphics переиспользуется для отображения графического интерфейса пользователя вне зависимости от программно-аппаратной платформы. Таким образом, отсутствует необходимость переключения графических API и слияния различных изображений при визуализации элементов управления поверх основной сцены, а, следовательно, не снижается производительность рендеринга.

Для осуществления специализированной визуализации используются библиотеки семейства VTK (для отображения сечений и объёмов) и библиотека NChart3D (для построения диаграмм, графиков и поверхностей).

Для поддержки устройств, обладающих малой вычислительной мощностью (таких, как смартфоны и планшетные компьютеры), сервер системы визуализации автоматически планирует распределение нагрузки и, при необходимости, на основе эвристических правил выполняет предобработку подлежащих визуализации данных. При этом учитывается тип клиента (настольный компьютер или мобильное устройство), производительность клиента (быстродействие и объём доступной памяти), скорость сетевого соединения клиента с сервером и загруженность сервера (объём доступной памяти

и количество подключённых клиентов). По результатам планирования выбирается один из режимов работы системы визуализации (от наиболее предпочтительного с точки зрения качества визуализации и скорости отклика на команды пользователя к менее предпочтительному):

1. Визуализация в полном объёме осуществляется на стороне клиента, а сервер лишь передаёт шаблон графической сцены, заполненный подлежащими визуализации данными.
2. Сервер перед отправкой выполняет частичное упрощение данных путём сокращения числа вершин в 3D-моделях, уменьшения разрешения текстур и сокращения числа ключевых кадров анимации.
3. Сервер целиком выполняет визуализацию на своей стороне и передаёт клиенту готовое изображение (как в случае использования технологии VNC).

Благодаря описанным архитектурным решениям удалось обеспечить совместимость системы научной визуализации и с настольными компьютерами, и с мобильными устройствами.

Обеспечение высокого качества визуализации

Для обеспечения высокого качества визуализации используется адаптивный алгоритм сглаживания границ объектов на изображении, являющийся суперпозицией модифицированных алгоритмов сглаживания на основе увеличенного изображения и быстрого ашпроксимированного сглаживания [4].

Суть предложенного алгоритма состоит в том, что сначала сцена визуализируется в текстуру разрешения, в s^2 раз превышающего разрешение экрана (коэффициент s вычисляется динамически на основании максимально допустимого размера текстуры). Затем текстура накладывается на полноэкранный спрайт с фильтрацией, включающей в себя билинейную интерполяцию при уменьшении изображения, а также поиск ступенек по перепаду яркости и их размытие в перпендикулярном к ним направлении. Алгоритм фильтрации реализован во фрагментном шейдере и оптимизирован для выполнения на графических процессорах мобильных устройств.

В отличие от стандартного для многих графических систем алгоритма сглаживания на основе множественной выборки, предложенный алгоритм:

1. Обеспечивает эффективную реализацию как для настольных компьютеров, так и для мобильных устройств, а также автоматически учитывает аппаратные особенности различного графического оборудования.

2. Обеспечивает быстрый отклик системы визуализации, так как сглаживание автоматически отключается на периоды динамического изменения сцены.
3. Обеспечивает визуальное качество результата не ниже, чем у системного алгоритма, при этом позволяет устранить нежелательное размытие объектов.

Заключение

В данной статье предложено использовать подход на основе онтологий для автоматизации процесса настройки систем научной визуализации на специфику сторонних решателей. Реализованная на основе предложенного подхода система научной визуализации SciVi не только легко адаптируема к сторонним решателям, написанным на различных языках программирования, но и позволяет адаптивным образом распределять процесс визуализации между клиентом и сервером. За счёт этого удаётся обеспечить одновременно высокую интерактивность и оптимальную загрузку вычислительной сети с учетом производительности клиента, скорости сетевого соединения и загруженности сервера. Высокое визуальное качество итогового изображения обеспечивается использованием нового алгоритма сглаживания границ, не увеличивающего задержку отклика системы. Мультиплатформенность обеспечивает эффективную работу SciVi как на настольных компьютерах, так и на мобильных устройствах.

SciVi протестирована под управлением операционных систем Windows, GNU/Linux, OS X, iOS и Android в процессе решения практических задач научной визуализации в различных предметных областях: физике, экономике, биологии, медицине, информатике и вычислительной математике.

Литература

- [1] Бондарев А.Е., Галактионов В.А. Анализ многомерных данных в задачах многопараметрической оптимизации с применением методов визуализации // Научная визуализация. – М.: Национальный исследовательский ядерный университет МИФИ, 2012. – К. 2, Т. 4, №2. – С. 1-13.
- [2] Васильев В.Р., Волобой А.Г., Вьюкова Н.И., Галактионов В.А. Контекстная визуализация пространственных данных // Информационные технологии и вычислительные системы. – М., 2004. – №4. – С. 25-34.
- [3] Ryabinin K., Chuprina S. Development of Multiplatform Adaptive Rendering Tools to Visualize Scientific Experiments // Procedia Computer Science. – Elsevier, 2014. – Vol. 29. – P. 1825-1834.
- [4] Рябинин К.В. Адаптивное сглаживание границ объектов на изображении для мобильных устройств // Вестник компьютерных и информационных технологий. – М.: Спектр, 2014. – №8. – С. 23-28.

Автоматизация генерации серий реалистичных изображений с использованием языка сценариев Python*

Н.Б. Дерябин¹, В.Г. Соколов¹, Д.Д. Жданов², М.С. Копылов¹

dek@keldysh.ru|sokolov@integra.jp|ddzhdanov@mail.ru|dvaag@hotmail.com

¹Институт прикладной математики им. М.В. Келдыша РАН, Москва, Россия;

²Государственный оптический институт им. С.И. Вавилова, Санкт-Петербург, Россия

В работе рассматриваются вопросы использования языка сценариев Python для различных направлений оптического моделирования и визуализации, таких как автоматическая генерация большого количества реалистичных изображений для анимации, или множественные световые расчеты со сложными источниками света, включая дневное освещение, и др. Предложенный подход с добавлением в сцену параметрических сценарных объектов-симуляторов делает процесс моделирования простым и удобным для конечного пользователя. Интерактивно изменяя параметры объекта-симулятора, пользователь настраивает моделирование под свои конкретные нужды. Предусмотрен механизм расширения программного комплекса новыми классами объектов-симуляторов. Так же рассматриваются примеры конкретных симуляторов.

Ключевые слова: автоматизация моделирования, генерация изображений, дневное освещение, оптические системы, расширяемость, язык сценариев, Python.

Automation of generating series of realistic images using Python*

N.B. Deryabin¹, V.G. Sokolov¹, D.D. Zhdanov², M.S. Kopylov¹

¹Keldysh Institute of Applied Mathematics, Moscow, Russian Federation;

²Vavilov State Optical Institute, St. Petersburg, Russian Federation

Practical issues of using Python script language for various applications of optical modeling and visualization such as automatic generation of a large number of realistic images for animation or multiple lighting simulations with complex light sources including daylight are considered. The presented approach which proposes to add the parametrical script-based simulator objects to the scene makes the modeling process simple and convenient for the end user. The proposed approach allows adjusting of modeling procedure according to user's requirements by changing parameters of the simulator interactively. Software extension mechanism by means of adding new script-based simulator classes is provided. Also examples of concrete simulators are considered.

Keywords: modeling automation, image generation, daylight, extensibility, script language, Python.

Введение

Каким бы продвинутым ни был графический пользовательский интерфейс сложной системы оптического моделирования, его оказывается недостаточно для решения многообразных задач, возникающих на практике. Для преодоления этой ограниченности используются языки сценариев. Выделим два преимущества, которые можно получить с помощью языков сценариев:

- автоматизация моделирования. Пользователь может запрограммировать последовательность действий, которую нужно выполнить для конкретной задачи моделирования, сохранить ее в скрипте (файле, написанном на языке сценариев), и затем многократно использовать как единое целое.
- расширяемость. Используя адекватный язык сценариев и соответствующую системную поддержку, пользователь может создавать в виде

скриптов новые классы (модули, пакеты), расширяющие систему оптического моделирования новыми типами объектов и новыми возможностями.

Мы успешно внедрили оба эти преимущества в наш программный комплекс, используя язык сценариев Python. Скрипты широко используются в наших программах для создания специальных сценарных объектов – симуляторов, выполняющих различные конкретные задачи моделирования. Использование симуляторов делает процесс моделирования простым и удобным для пользователя.

В работе описывается соответствующий интерфейс пользователя – оператора программного комплекса оптического моделирования, излагаются принципы написания сценариев моделирования на языке Python, и приводятся примеры конкретных процессов моделирования.

Интерфейс конечного пользователя

Простейший способ автоматизации моделирования при помощи сценариев – это записать последовательность требуемых действий на языке Python,

Работа поддержана грантами РФФИ 13-01-00454, 15-01-01147, а также фирмой Integra Inc. (Япония). Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347.

загрузить эту последовательность (например, из файла) в специальное командное окно комплекса оптического моделирования, и выполнить (Рис. 1).

На практике этот способ, поддерживаемый в том или ином виде во многих современных графических системах, не удобен из-за существенных ограничений: алгоритм моделирования хранится вне сцены; изменение параметров моделирования затруднено и требует изменения самого скрипта, который на практике бывает большим и сложным. Поэтому нами была разработана возможность расширения сцены специализированными параметрическими объектами моделирования.

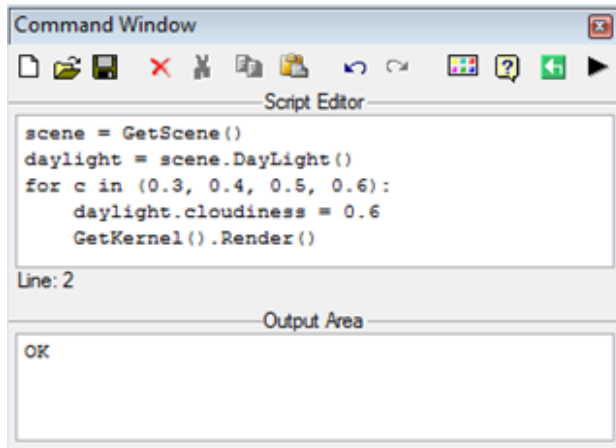


Рис. 1: Командное окно

Такие объекты создаются в библиотеке объектов при помощи специальной инструментальной кнопки создания сценарного объекта (см. Рис. 2).

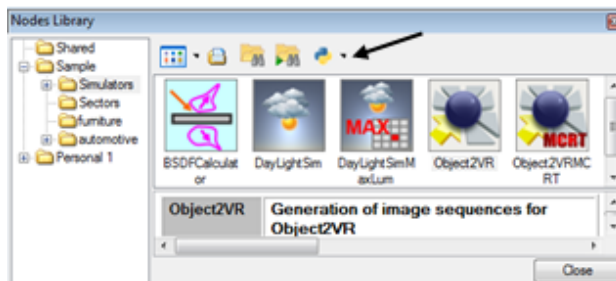


Рис. 2: Библиотека симуляторов

Процесс создания включает создание класса объекта – специального скрипта на языке Python. Ряд готовых, наиболее употребительных объектов моделирования вместе с online-документацией заранее включен в библиотеку, так что конечный пользователь обычно просто выбирает требуемый объект.

Желаемый объект моделирования включается в сцену путем простого перетаскивания мышью из

библиотеки в иерархию сцены (отмечено черной стрелкой на Рис.3).

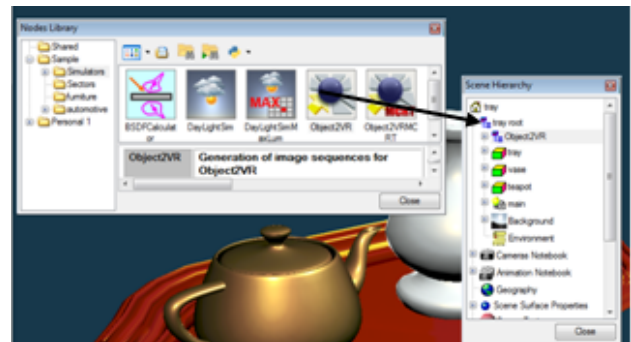


Рис. 3: Включение объекта-симулятора в сцену

После этого объект моделирования становится частью сцены и сохраняется вместе с ней.

При желании скрипт (класс) объекта можно непосредственно редактировать как в библиотеке, так и в сцене при помощи встроенного редактора скриптов, вызываемого через контекстное меню (см. Рис. 4).

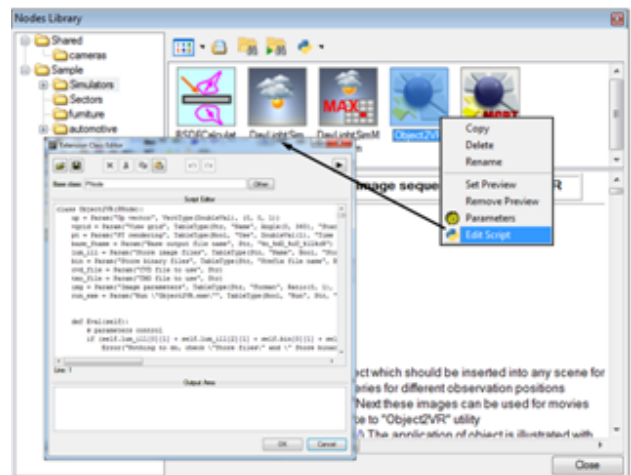


Рис. 4: Редактирование скрипта симулятора

Каждый сценарный объект имеет параметры, которые служат для настройки конкретного процесса моделирования. Набор возможных параметров сценарного объекта задается в классе (скрипте) объекта. Типы параметров варьируются от простых скаляров и строк до сложных таблиц со смешанными типами элементов.

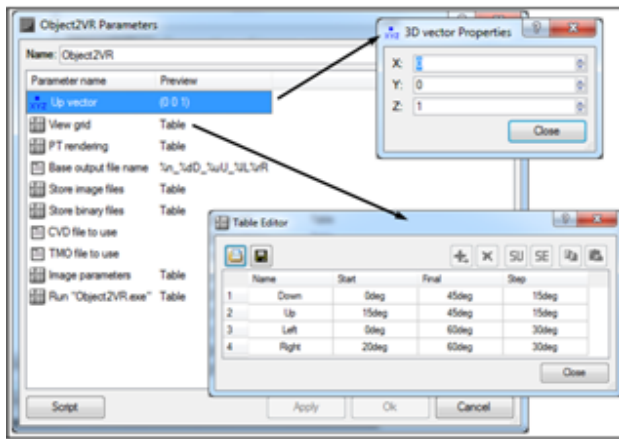


Рис. 5

Параметры сценарного объекта можно редактировать при помощи редактора параметров (Рис. 5), вызываемого через контекстное меню. Редактируя параметры, пользователь настраивает процесс моделирования под свои конкретные нужды. Текущие значения параметров симулятора сохраняются вместе со сценой. Помимо параметров моделирования, скрипт объекта описывает и саму процедуру моделирования. Эта процедура именуется специальным образом, так что ее имя появляется в контекстном меню сценарного объекта. Моделирование запускается путем вызова этой процедуры через контекстное меню сценарного объекта в иерархии сцены (соответствующий элемент меню отмечен черным прямоугольником на Рис.6). Один симулятор может определять несколько процедур моделирования.

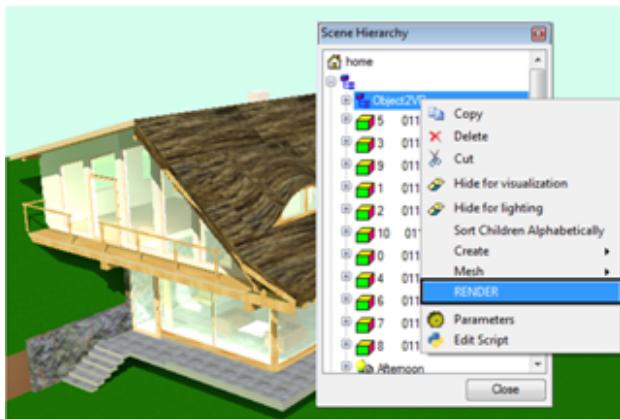


Рис. 6: Запуск моделирования

В ходе моделирования могут изменяться атрибуты объектов сцены (например, положение камеры и солнца, визуальные свойства поверхностей, оптические свойства материалов и т.п.). Эти изменения отображаются в главном окне приложения, что дает возможность пользователю визу-

ально контролировать процесс моделирования. Результаты моделирования (например, набор сгенерированных изображений) обычно записываются процедурой моделирования в файлы для последующего использования.

Все это обеспечивает удобную и эффективную работу для конечного пользователя.

Интеграция с языком Python

Наш комплекс оптического моделирования построен на объектно-ориентированной архитектуре [1], которая естественным образом перекладывается на язык Python. Не вдаваясь в технические детали, опишем принципы этого переложения.

Сцена представляет собой сложную иерархию объектов, имеющих общий унифицированный программный интерфейс, называемый целевым интерфейсом. Для каждого объекта, целевой интерфейс определяет набор его свойств (атрибутов) и набор действий (процедур), которые можно выполнить над ним.

Целевой интерфейс естественным образом транслируется в инструкции на языке Python. Доступ (чтение, модификация) к свойству объекта транслируется в язык Python как обращение к переменной - члену класса. Используется обычная запись через точку:

```
x = obj.prop
obj.prop = x
x = obj.list[i]
```

Исполнение действия записывается как вызов члена-метода:

```
obj.Proc(p1, p2)
res = obj.Func(p1)
```

Наличие унифицированного целевого интерфейса позволило нам реализовать интерпретатор языка Python как отдельную независимую программную компоненту.

Используя программный интерфейс комплекса оптического моделирования, пользователь может выполнять произвольное моделирование со сценой через командное окно (Рис. 1). Нам осталось описать, формат класса расширения для конкретных процессов моделирования (Рис. 4).

Формат класса простой. Заголовок класса, такой как

```
class Object2VR(PNode) :
```

задает имя класса (Object2VR). Класс моделирования всегда наследуется от предопределенного типа (класса) PNode. Далее следуют определения параметров:

```
up = Param("UpVector VectType(Float),
(0, 0, 1))
```

```
vgrid = Param("Viewgrid
TableType(Str, "Name Angle,
"Start" Final" Step ...))
```

...

Описание параметра задает два имени параметра: для использования в скриптах ("*up*") и для использования в редакторе параметров ("*UpVector*"). Задается также тип параметра и (опционально) значение по умолчанию. Параметры являются свойствами сценарного объекта.

Табличный тип задается при помощи мета-типа `TableType`. Строки в кавычках задают имена колонок таблицы. В приведенном примере первая колонка таблицы имеет имя "*Name*" содержит строки (тип *Str*), а следующие три колонки ("*Start*" "*Final*" "*Step*") содержат углы в градусах (тип *Angle*).

Помимо параметров, класс моделирования обычно задает функцию моделирования:

```
@Method("RenderImages")
def Render(self) :
```

...

Это описание функции использует специальный декоратор `@Method()`. Этот декоратор включает данный метод в контекстное меню сценарного объекта.

Таким образом, как мы надеемся, использование языка Python для оптического моделирования не представляет проблем.

Примеры симуляторов

Первый пример сценарного симулятора, включенный в наш программный комплекс, предназначен для генерации серий изображений с различных угловых точек наблюдения (точка 2 на Рис. 7) нацеленных на заданную точку 1 в пространстве (точка 1 на Рис. 7). Основные параметры генерации: угловые диапазоны горизонтального (Γ - Γ) и вертикального (B - B) перемещения точки наблюдения и угловой шаг. Множество других параметров симулятора позволяет управлять различными характеристиками генерации изображений от выбора методики расчета до формата имен генерируемых файлов с изображениями.

Результатом работы симулятора может быть серия обычных изображений или изображений в формате с динамической яркостью, допускающих последующую постобработку [2]. Эти изображения могут быть непосредственно использованы для создания интерактивных трехмерных анимационных моделей, например, с помощью программы `Objetct2VR`.

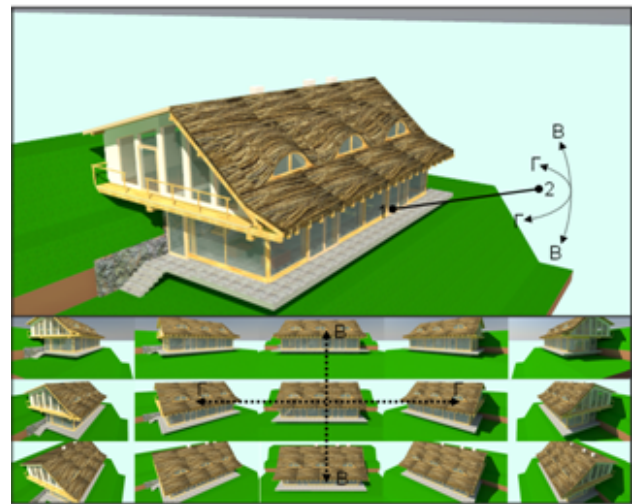


Рис. 7: Симулятор для создания анимаций

Второй пример симулятора, разработанный нами и включенный в наш программный комплекс, используется для эргономического анализа освещенности [3]. Этот симулятор позволяет вычислять точку максимальной яркости изображения для различных условий дневного освещения.

При помощи параметров, пользователь может задавать географическое положение объектов сцены, список дат (год, число, месяц) для моделирования, интервал и шаг изменения времени и даже вращение объектов сцены. В результате работы симулятора создаются текстовые файлы с данными о максимальной яркости с указанием координат (номер колонки и ряда) на изображении, где была зафиксирована максимальная яркость. По желанию пользователя так же могут быть созданы и сами изображения (их создание управляется порогом яркости). Рис. 8 представляет выходные данные симулятора.

Варианты 1 и 2 отличаются различной ориентацией сцены, которая управляется с помощью вращения вектора направления на юг. В примере исследуется яркость внутри салона автомобиля при вращении руля и в различное время. Рисунок показывает фрагменты файла-отчета, содержащие максимальную яркость, и несколько примеров реалистичных изображений сцены. Таким образом, пользователь может проводить быстрый анализ результатов моделирования [4, 5].

Заключение

Описанный подход с использованием объектов-симуляторов, определяемых на языке сценариев, обеспечивает разработку удобных, автономных методов для различных сложных множественных моделирований [4]. Приведены примеры таких симуляторов, нашедшие широкое применение у пользователей наших программных продуктов. Развитый

пользовательский интерфейс позволяет разрабатывать новые специализированные симуляторы с минимальными затратами времени.

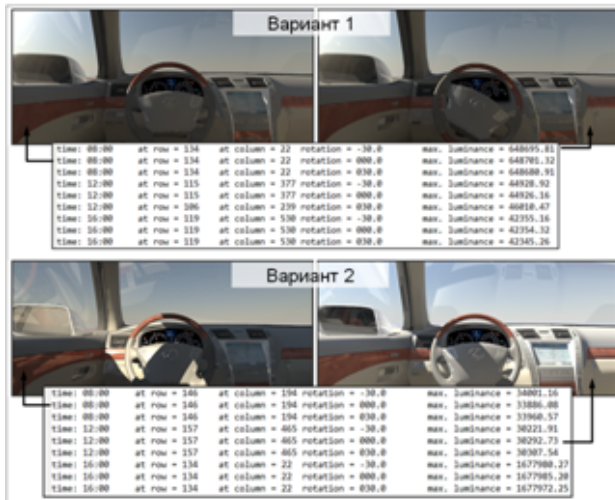


Рис. 8: Симулятор для эргономического анализа

Литература

[1] Галактионов В.А., Дерябин Н.Б., Денисов Е.Ю. Объектно-ориентированный подход к реализации систем компьютерной графики. //

Информационно-измерительные и управляющие системы. – 2009. - № 6. - С.96-108.

- [2] Барладян Б., Бирюков Е., Валиев И., Волобой А., Шапиро Л. Постобработка и анализ синтезированных реалистичных изображений. // ГрафиКон'2014: 24-я Международная конференция по компьютерной графике и зрению, Ростов-на-Дону: Академия архитектуры и искусств ЮФУ, 2014. - С.55-58.
- [3] Волобой А.Г., Галактионов В.А. Компьютерная графика как эффективный инструмент развития современных технологий. // ГрафиКон'2013: 23-я Международная конференция по компьютерной графике и зрению, Владивосток, Институт автоматки и процессов управления ДВО РАН, 2013. – С.186-190.
- [4] Клышинский Э.С., Рысаков С.В., Шихов А.И. Обзор методов визуализации многомерных данных. // Новые информационные технологии в автоматизированных системах: материалы семнадцатого научно-практического семинара, Москва: ИПМ им. М.В.Келдыша, 2014. – С.519-530.
- [5] Бондарев А.Е., Галактионов В.А., Четкин В.М. Анализ развития концепций и методов визуального представления данных в задачах вычислительной физики. // Журнал вычислительной математики и математической физики. – 2011. - Т.51, № 4. - С.669-683.

Сравнение методов распознавания символов номерного знака автомобиля

С.А. Куксова

s.kuksova@inbox.ru

Национальный исследовательский университет «МИЭТ», Москва, Зеленоград, Россия

В данной статье рассмотрена задача распознавания символов автомобильного номера. Алгоритм распознавания опирается на метод сопоставления шаблонов. Предлагаются два подхода получения признакового описания шаблонов: расчет инвариантов моментов и distance transform. Затем проводится их сравнение. Представлен сравнительный анализ рассматриваемых подходов на экспериментальных данных.

Ключевые слова: метод сопоставления шаблонов, моментные инварианты, distance transform, метрика Хаусдорфа, распознавание образов

Comparison of vehicle's license plates characters recognition methods

S.A. Kuksova

National Research University of Electronic Technology, Moscow, Zelenograd, Russia

The problem of license plate character recognition is considered in this article. The recognition algorithm is based on the pattern matching method. There are two approaches of patterns feature descriptions obtaining: calculating invariants moments and distance transform. Then they are compared. The comparative analysis of these approaches to the experimental data is represented.

Keywords: pattern matching method, moment invariants, distance transform, Hausdorff metric, pattern recognition

Введение

Задача автоматического распознавания автомобильных номеров на цифровых изображениях и видео имеет множество решений, но все равно остается актуальной из-за отсутствия достаточно точного решения, универсального для различных программно-аппаратных систем. Алгоритмы распознавания номеров в таких системах должны быть устойчивыми к искажениям изображений номерных пластин, а также изменениям условий внешнего света. Рассмотрим подробнее этап в решении задачи распознавания, связанный с классификацией символов автомобильного номера.

В целом, методы распознавания символов можно разбить на два класса: структурные методы, основанные на морфологии и анализе контура, имеющие дело с бинаризованным изображением, и растровые методы, основанные на анализе самого изображения. При этом зачастую используется комбинация структурных и растровых методов.

В данной работе рассматривается метод основанный на непосредственном сравнении изображений тестового и эталонного символов. Описывается подход к решению задачи на основе метода сопоставления шаблонов, использующий в качестве признакового описания моментные инварианты. Далее дается описание этого же метода распознавания, использующего «дистантное преобразование». Затем на подготовленном наборе тестовых изображений проводится эксперимент с целью сравнения описанных выше методов. В результате

сравнения делается вывод о количестве правильно распознанных тестовых изображений.

Постановка задачи

Рассматривается задача распознавания изображений символов автомобильных номеров. Заданы несколько классов изображений чисел и букв номерных знаков и одно тестовое изображение. Размеры изображений различны, поэтому в этапе предварительной обработки они приводятся к одному размеру. Задача состоит в том, чтобы для тестового изображения найти наиболее похожее на него изображение из заданных образцов, и затем определить принадлежность к классу. Степень схожести определяется метрикой на множестве изображений.

Метод сопоставления шаблонов

Метод распознавания символов с помощью сопоставления шаблонов предполагает наличие шаблонов для всех возможных изображений символов. Принятые решения о принадлежности текущего изображения символа из тестовой выборки к определенному классу символов осуществляются по критерию минимума (максимума) некоторой метрики сходства изображения символа и его шаблона [1]. Каждый класс определяется его усредняющим вектором, т.е. используется среднее значение по обучающему множеству данного класса в качестве представителя данного класса векторов. Способ отнесения неизвестного образа к вектору признаков к некоторому классу состоит в выборке того

класса, чей прототип ближе всего к рассматриваемому вектору. В качестве меры близости образов задача заключается в вычислении расстояний Хаусдорфа. После этого исследуемый образ относится к какому-либо классу, имеющему наименьшее расстояние [2].

Метрика Хаусдорфа

В этом разделе описывается подход к решению задачи распознавания, использующий метрику Хаусдорфа для определения схожести изображений. Дадим определение метрики [3].

Отклонением множества из одной точки x'_0 от компактного множества G называется

$$\sigma(x'_0, G) = \min_{y' \in G} \|x'_0 - y'\|, \quad (1)$$

где $\min \|x'_0 - y'\|$ – евклидово расстояние между точками x'_0 и y' .

Отклонением множества G_1 от G_2 называется

$$\sigma(G_1, G_2) = \max_{x' \in G_1} \sigma(x', G_2) \quad (2)$$

Метрикой Хаусдорфа называется

$$\rho(G_1, G_2) = \max(\sigma(G_1, G_2), \sigma(G_2, G_1)) \quad (3)$$

Признаковое описание

Выделение признаков является одной из наиболее трудных и важных задач в распознавании образов. Для распознавания символов могут использоваться различные системы признаков. Трудность заключается в том, чтобы выделить наиболее эффективные признаки, которые позволят достаточно хорошо отличать один класс символов от всех остальных.

Существует группа методов, выделение признаков в которых осуществляется на основе анализа различных статистических распределений точек. Наиболее известный – метод вычисления моментных инвариантов.

Инварианты моментов – это признаки изображения, построенные на основе степенных моментов и описывающие силуэт некоторого объекта. В соответствии со своим названием данные признаки являются инвариантными к аффинным преобразованиям изображения [4].

Рассматриваются центральные моменты, которые являются функцией расстояния точки от центра тяжести символа [1]:

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad (4)$$

где $\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}}$ – координаты центра тяжести. Вследствие большей инвариантности к преобразованиям изображений рассматриваются нормированные центральные моменты, которые получаются в результате деления центральных моментов на моменты нулевого порядка:

$$n_{pq} = \frac{m_{pq}}{m_{00}^\gamma} \quad (5)$$

при $p, q = 0, 1, 2, \dots$, где $\gamma = \frac{p+q}{2} + 1$ при $p+q = 2, 3, \dots$

Имея все эти моменты, можно определить следующие семь инвариантов моментов, которые инвариантны относительно переносов, осевой симметрии, поворотов, а также растяжений и сжатий:

$$\begin{aligned} \varphi_1 &= n_{20} + n_{02}, \\ \varphi_2 &= (n_{20} - n_{02})^2 + 4n_{11}^2, \\ \varphi_3 &= (n_{30} - n_{12})^2 + (3n_{21} - n_{03})^2, \\ \varphi_4 &= (n_{30} + n_{12})^2 + (n_{21} + n_{03})^2, \\ \varphi_5 &= (n_{30} - 3n_{12})(n_{30} + n_{12})[(n_{30} + n_{12})^2 - 3(n_{21} + n_{03})^2] + (3n_{21} - 3n_{03})(n_{21} + n_{03})[3(n_{03} + n_{12})^2 - (n_{21} + n_{03})^2], \\ \varphi_6 &= (n_{20} - n_{02})[(n_{30} + n_{12})^2 - (n_{21} + n_{03})^2] + 4n_{11}(n_{30} + n_{12})(n_{21} + n_{03}), \\ \varphi_7 &= (n_{20} - n_{03})(n_{30} + n_{12})[(n_{30} + n_{12})^2 [3(n_{30} + n_{12})^2 - (n_{21} + n_{03})^2]]. \end{aligned}$$

В итоге формируется вектор признаков для каждого изображения обучающего множества, состоящий из описанных выше инвариантов. В дальнейшем на основе этих векторов формируется шаблон для каждого класса изображений. Усредненный вектор для каждого семейства изображений можно представить в виде простого среднего:

$$m_x = \frac{1}{K} \sum_{k=1}^K x_k \quad (6)$$

при $K = MN$, где M, N – размеры изображения.

Distance Transform в качестве яркости точки выдает расстояние до ближайшей черной точки, соответствующей контуру.

В начале формируется шаблон для каждого класса цифр путем усреднения интенсивностей изображений. Затем для каждого шаблона применяется детектор Кэнни, с целью выделения контуров.

В итоге шаблоны описываются векторами признаков, полученных в результате «дистантного преобразования».

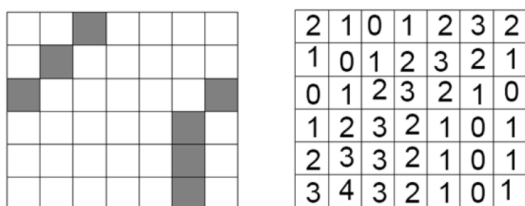


Рис. 1: Результат выполнения Distance Transform.

Описание алгоритма

Так как изначально матрицы шаблонов символов и теста обладают различными размерами, то на первом этапе изображение приводится к размеру шаблона путем масштабирования. Далее проводится предварительная обработка: выполняется высокочастотная фильтрация с целью увеличения контраста. На втором этапе используются два вида признаковых описаний, рассмотренных выше.

Далее с помощью метода сопоставления шаблонов проводится сопоставление тестового изображения с шаблоном каждого класса. Данный метод сравнения основывается на основе описанного выше расстояния Хаусдорфа. Проводится поиск такого взаимного расположения изображений теста и шаблона, чтобы значение метрики Хаусдорфа между ними было минимальным. Из этого минимума определяется класс принадлежности вектора признаков тестового изображения.

Преимущества этого подхода к распознаванию изображений символов – в высокой скорости алгоритма классификации и его простоте.

Результаты работы

Теперь приведем результаты работы алгоритмов сравнения на наборе тестовых изображений. Имелась тренировочная выборка изображений символов «0» ... «9», «А, В, С, Е, Н, К, М, О, Р, Т, Х, У», разбитая уже на классы. В каждом классе в среднем 200-300 образцов. Тестовая выборка получена из одного набора данных путем их разбиения случайным образом. Эксперимент для каждого алгоритма включал в себя 50 тестовых изображений для каждого типа символов.

Результаты работы представлены в таблице 1:

Таблица 1: Зависимость точности от метода

Алгоритм	Точность
Использование инвариантов моментов	37%
Использование Distance Transform	70%

Типичные ошибки предлагаемых алгоритмов заключаются в том, что подаваемый символ понимается за другой. Часто возникает ошибочная классификация между двумя похожими символами, таких как «0» и «О», «8» и «В», «Х» и «К», «Х» и

«У». Решением этой проблемы предлагается дополнить методы алгоритмами уточнения символов.



Рис. 2: Примеры тестовых изображений.

Заключение

В работе был рассмотрен метод распознавания символов автомобильного номера на основе сопоставления шаблонов с разными подходами получения шаблонов. Первый растровый метод, основанные на анализе непосредственно изображения (расчет инвариантов моментов), второй – структурный метод основанные на морфологии и анализе контура (distance transform). Описана метрика Хаусдорфа, которая используется в методе сопоставления шаблонов. В завершение приведены результаты тестирования обоих описанных подходов на одной и той же тестовой выборке. По результатам тестирования, метод на основе distance transform превзошел метод на основе расчета инвариантов моментов в 1,89 раз по точности распознавания на тестовой выборке.

Для дальнейшей работы выбирается подход, основанный на использовании Distance Transform для получения признакового описания. С целью увеличения точности распознавания предлагается использовать уточняющие алгоритмы для «проблемных» символов. В частности, метод опорных векторов, который относится к инструментам машинного обучения. Здесь будут рассматриваться отдельные обучающие выборки для каждого определенного символа. В качестве дополнительного признакового описания можно рассмотреть метод гистограмм ориентированных градиентов, часто применяющийся в других задачах распознавания.

Следующим этапом исследований является распознавание символов номера методом на основе машинного обучения и сравнения с результатами, полученными в этой работе.

Литература

- [1] Гонсалес Р., Вудс Р. Цифровая обработка изображений. Москва: Техносфера, 2005. 1070с.
- [2] Гонсалес Р., Вудс Р., Эддинс С. Цифровая обработка изображений в среде MATLAB. Москва: Техносфера, 2006. 615с.
- [3] Михайлов И.А. Об одном методе распознавания изображений. // Моделирование и анализ информационных систем. 2007. Т.14, №4, С.7-12
- [4] Самарина О.В. геометрический подход к определению инвариантов изображения. Автореф. дис. кфмн // Югорский государственный университет. Х.-М., 2008, 20с.

Расчет объективной и субъективной интегральной яркости объекта по изображению широкого динамического диапазона *

К.С. Зипа, Е.Е. Земцов, А.В. Игнатенко

kzipa@graphics.cs.msu.ru|zemtsov.e.e@gmail.com|ignatenko@graphics.cs.msu.ru

Факультет вычислительной математики и кибернетики

Московский государственный университет им.М.В.Ломоносова

Задача расчета интегральной яркости объекта актуальна при оценке визуальных свойств предметов. В ряде случаев (ткани, косметика, предметы интерьера) важно рассчитать именно воспринимаемую, а не физическую яркость. В данной статье впервые формализуется задача поиска воспринимаемой яркости объекта. В статье предлагается решение задачи расчета как интегральной объективной, так и интегральной воспринимаемой яркости объекта по изображению широкого динамического диапазона. Изображения широкого динамического диапазона содержат физические значения яркости объектов, поэтому работа с ними можно получить более корректный результат с точки зрения объективного значения отражательной способности. Была проведена экспериментальная верификация, показавшая превосходство выбранного метода оценки воспринимаемой яркости над простым усреднением объективной физической яркости.

Ключевые слова: HDR, яркость, восприятие, динамический диапазон.

The calculation of the integral objective and subjective brightness of the object in the image of a wide dynamic range*

K.S. Zipa, E.E. Zemtsov, A.V. Ignatenko

Faculty of Computational Mathematics and Cybernetics

Lomonosov Moscow State University, Moscow, Russia

The problem of calculating the integral brightness of the subject is relevant in assessing the visual properties of objects. In some cases (textiles, cosmetics, home furnishings) it is important to calculate the perceived rather than the physical brightness. This article was first formalized the task of finding the perceived brightness of the subject. The paper proposes a solution to the problem of calculation as an integral objective and integrated the perceived brightness of the object in the image of a wide dynamic range. Images of the wide dynamic range of brightness values contain physical objects, so you can work with them to get a correct result in terms of the objective value of reflectivity. Experimental verification was carried out, which showed the superiority of the selected method of assessing the perceived brightness of a simple averaging objective physical brightness.

Keywords: HDR, brightness, perception, dynamic range.

Введение

Человек воспринимает окружающий мир в основном через зрение. Происходит это благодаря зрительной системе, которая превращает световые раздражители в сигналы для последующей передачи по нервным путям в мозг, где происходит анализ поступившей информации и формирование субъективных ощущений.

Начальные этапы обработки сигналов в настоящее время изучены неплохо, однако до сих пор не существует полного описания процессов, происходящих в мозгу человека. До сих пор наука не в состоянии предоставить модель преобразования физических яркостей в воспринимаемые. Примеры таких иллюзий, вступающих в противоречие с физическими величинами, приведены на Рис. 1.

Тем не менее, некоторые закономерности известны и опубликованы в литературе. Это позволяет

частично решать задачу расчета интегральной яркости объекта по фотографии, опираясь на достижения компьютерной графики и психофизиологии, чему и посвящена данная статья.

Повествование ведется следующим образом.

В главе 2 ставится задача расчета интегральной воспринимаемой яркости объекта по фотографии. Эта задача является новой, однако её решение возможно на основе уже существующих алгоритмов.

Работа ведется с изображениями широкого динамического диапазона. Изображение называется изображением широкого динамического диапазона, если количество различных уровней яркости в нём превышает 256 значений. Процесс получения изображений широкого диапазона хорошо изучен, однако в нём есть некоторые тонкости, которые будут рассмотрены в главе 3. Рассмотренный процесс позволяет получить точные фотометрические значения интегральной яркости объекта с использованием обыкновенной фотокамеры.

Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347

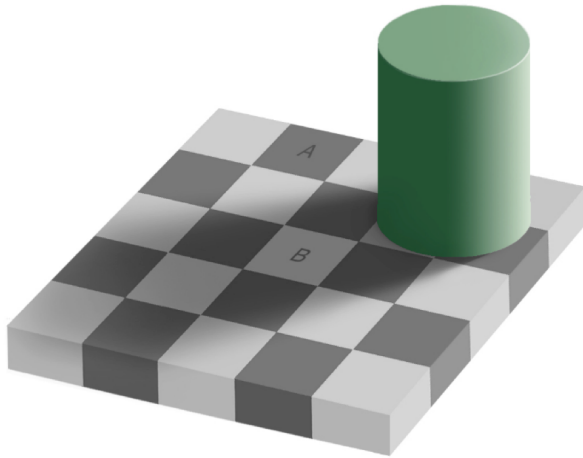


Рис. 1: Иллюзии, демонстрирующие отличие воспринимаемой яркости от физической.

В главе 4 описывается предложенный подход к решению задачи определения воспринимаемой яркости объекта. В главе 5 описывается процесс верификации и производится сравнение с другими методами. В главе 6 выписаны краткие результаты работы. В главе 7 представлен список используемой литературы.

2. Постановка задачи

Задача заключается в определении воспринимаемой человеком яркости некоторого объекта на изображении. Яркость, как характеристика восприятия, не имеет стандартных единиц измерения. Чтобы провести расчет корректно, необходимо иметь входные данные зрительной системы: физические яркости сцены, информацию об условиях освещения, местоположения и границах интересующих объектов.

Определению физических яркостей сцены посвящена следующая глава. В ней описывается технология получения корректных фотометрических яркостей с помощью калиброванного фотоаппарата. Остальную информацию – условия освещения, местоположение и границы объекта нужно получить извне.

Итак, на входе алгоритма:

- изображение широкого динамического диапазона со значениями фотометрической яркости $(L(p))$,
- маска исследуемого объекта (M) ,
- информация об освещении (θ) .

На выходе алгоритма A требуется получить интегральную характеристику яркости объекта, заданного маской.

$$I = A(L(p), M, \theta),$$

$p \in P$ – множество пикселей изображения.

К решению предъявляются следующие требования:

- значения должны быть представлены в некоторой субъективной шкале,
- если между двумя объектами изображения человек не видит различий по яркости (воспринимаемые яркости совпадают), то различие между ними не должно превышать некоторого Δ , складывающегося из едва различимого инкремента восприятия и погрешности алгоритма,
- субъективное отношение яркости между любыми двумя объектами должно соответствовать отношению величин рассчитанной интегральной яркостью,
- при предъявлении экспертам двух изображений: исходного и имеющего карту со значениями, равными интегральной яркости объекта, они должны в среднем оценить область яркости объекта на них как одинаковую.

3. Определение интегральной объективной яркости объекта с помощью построения изображения широкого диапазона

Получить изображение широкого диапазона можно двумя основными способами:

- 1) Синтез изображения с помощью компьютерной графики
- 2) Получение изображения широкого диапазона с помощью наборов фотографий одного кадра, снятых с разной выдержкой[1]. Пример двух фотографий с разной выдержкой представлен на Рис. 2.



Рис. 2: Фотографии одной сцены, снятые с разной выдержкой.

Первый способ не годится, так как не работает с объектами реального мира. Второй способ выглядит самым подходящим, он требует лишь наличие обыкновенной цифровой камеры и штатива.

Итак, рассмотрим задачу синтеза HDR-изображения по набору LDR-изображений. Это классическая задача, решение которой было предложено в статье [1].

Что происходит, когда свет попадает на матрицу камеры? Когда затвор камеры открывается, фотоны начинают воздействовать на сенсор (чем ниже ISO, тем больше фотонов должно попасть на матрицу для активации сенсора). По мере увеличения числа фотонов, сигнал сенсора усиливается. При закрытии затвора на матрице оказывается готовый отпечаток для заданной выдержки.

До этого момента ясно, что если выдержки на двух последующих кадрах отличаются в 2 раза, то и значения на матрицах будут тоже отличаться в 2 раза. Однако после этого производители фотоаппаратов применяют нелинейное преобразование, которое называют кривой отклика камеры. Это преобразование уникально для разных фотоаппаратов.

Метод, описанный в статье [1] как раз и позволяет восстановить кривую отклика камеры, а после этого рассчитать значения физической яркости, которые записываются в изображение широкого динамического диапазона.

Особенность этого алгоритма заключается в том, что функция отклика камеры восстанавливается лишь с точностью до константы. Для получения абсолютных значений фотометрической яркости эту константу необходимо найти. Статья [2] рассматривает несколько способов поиска. Но все они объединены общей идеей: необходимо сфотографировать объект с известной физической яркостью.

Чтобы измерить физическую яркость какого-либо объекта, можно использовать специальные приспособления. Варианты устройств:

- 1) luminance meter – измеряет напрямую яркость, но дорог,
- 2) lux meter – измеряет количество падающего света, для расчета излучаемого света необходимо специальное освещение,
- 3) спотметр – измеряет exposure value, используется фотоаппаратами, полученное значение преобразуется в яркость, не дорого стоит, не требует специального освещения.

Итак, каждому фотоаппарату кроме кривой отклика соответствует калибровочная константа. Её можно получить, если сфотографировать объект с известной яркостью. Яркость объекта определяется с помощью спотметра (I_{spot}). В качестве калибровочного объекта может выступать печатное изображение на Рис. 3.

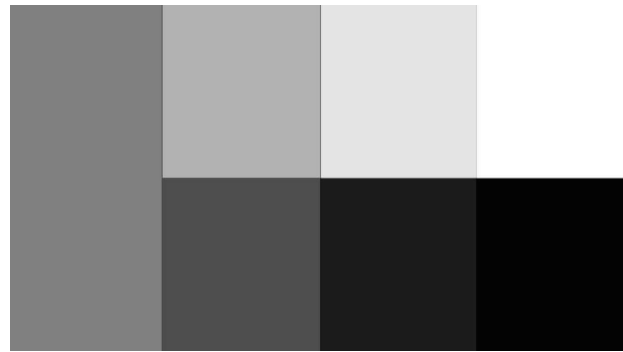


Рис. 3: Калибровочный объект.

Калибровочную константу получим по формуле:

$$k = I_{spot} / I_{hdr},$$

I_{hdr} – результат алгоритма,

I_{spot} – физическая яркость объекта.

В своих экспериментах мы брали медиану из нескольких замеров (параметры ISO100/400, f4.0/5.6, ориентация шаблона в пространстве (поворот на 180 градусов), глянцева и матовая бумага). Общая схема получения объективной яркости объекта по фотографии представлена на Рис.4.

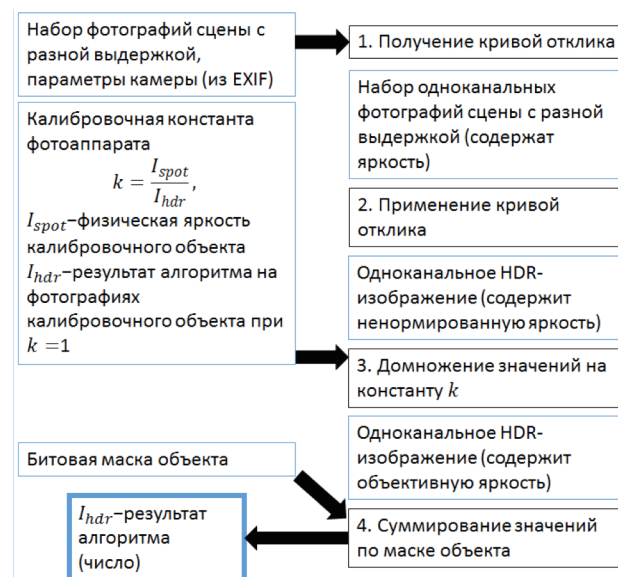


Рис. 4: Схема получения интегральной физической яркости объекта.

Было установлено, что наименьшую ошибку имеют замеры, проведенные по серым квадратам из тестового шаблона. Таким образом, имея скалированный фотоаппарат, можно получить точные фотометрические значения яркости в каждом пикселе изображения.

Далее, для получения объективной яркости объекта, для которого известна его маска на изображении, остается только усреднить значения яркостей

пикселя объекта:

$$B_{objective}(object) = \frac{\sum_{i=1}^N I_{objective}(p_i)}{N}, p_i \in object.$$

4. Определение интегральной субъективной яркости объекта с помощью построения изображения широкого диапазона

Мы решаем поставленную задачу впервые, поэтому законченных решений, с которых можно было бы начать рассмотрение, не существует. Поэтому сразу проведем декомпозицию задачи на 2 подзадачи:

1. Расчет воспринимаемой яркости для каждого пикселя.
2. Суммирование точечных яркостей для расчета интегральной яркости объекта.

Поскольку в этой статье предлагается первая попытка к решению задачи, мы решили использовать в качестве суммирования классическое понятие усреднения значений яркости в изображении:

$$B_{subjective}(object) = \frac{\sum_{i=1}^N I_{subjective}(p_i)}{N}, p_i \in object.$$

Для расчета точечной яркости мы решили сосредоточиться на методе глобальной коррекции, который преобразует яркость пикселя в соответствии с общим ощущением яркости, имитируя процесс глобальной адаптации человеческого зрения.

4.1 Нелинейность восприятия яркости, хроматическая адаптация

Сначала стоит отметить актуальность проблемы определения именно воспринимаемой (субъективной) яркости. Во введении был приведен пример иллюзии, демонстрирующей качественное отличие результата восприятия от физических значений интенсивности света, попавшего на сетчатку.

Теперь мы перечислим некоторые особенности человеческого зрения, которые мы хотим промоделировать в построенной карте воспринимаемой яркости.

Во-первых, это нелинейность восприятия яркости. Закон нелинейного восприятия был описан Стивенсом [3] и заключается в том, что воспринимаемая яркость монотонного поля приблизительно изменяется по степенному закону относительно физической яркости этого поля.

$$B = k * I^n$$

I – физическая яркость,

B – воспринимаемая яркость.

Другая важная особенность зрения – способность адаптироваться к различному освещению. Это явление получило название хроматической адаптации. На Рис. 5. представлена сцена, снятая в

различных освещениях. Благодаря хроматической адаптации цвет лилия в сцене не изменяется с точки зрения человека.



Рис. 5: Эффект цветовой адаптации. Источник – http://traditio-ru.org/wiki/Цветовая_адаптация

4.2 Глобальные модели яркости

Рассмотрим некоторые цветовые модели, которые берут в расчет специфику человеческого зрения. Первая модель – модель фон Криса [4]. В ее основе лежит предположение о том, что каждый вид рецепторов приспособляется независимо друг от друга. Эта модель обладает низкой точностью, но на её основе построены более сложные преобразования.

Улучшения предложил Nayatani и др. [5]. Нервные клетки проявляют постоянную активность, даже когда на них не оказывается воздействия, то есть они генерируют некоторый случайный сигнал (шум). Авторы добавили искажения во входное изображение, чтобы симитировать реальное действие клеток человека. Отличие этой модели в применении нелинейного преобразования, что и обеспечивает преимущество данной модели от модели фон Криса.

Модель [6] – наиболее сложная и точная из всех. Однако на вход требуется большое количество характеристик сцены, их замеры сложны и верное применение требуют серьезного опыта от экспериментатора.

Последней рассмотрим CIECAM02[7] – стандарт, принятый международной комиссией по освещению (CIE) в 2002 году. Она изначально разрабатывалась, как универсальная модель, вобравшая в себя все наработки. Модель получилась менее точной, чем Hunt, однако и гораздо более простой для применения.

Для расчета точечной субъективной яркости решено использовать модель CIECAM02, потому что

она не требовательна ко входным данным и при этом показывает достойные результаты.

4.3 Итоговая схема алгоритма

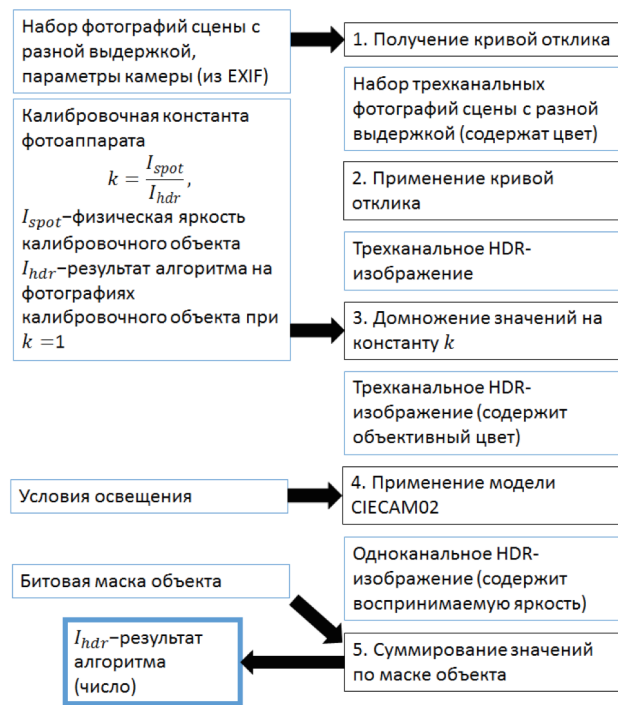


Рис. 6: Схема получения интегральной воспринимаемой яркости.

На вход подаётся: набор изображений с разной выдержкой и параметры камеры (выдержка для каждой фотографии, диафрагма, значение ISO, калибровочная константа). На выходе первого этапа – HDR-картинка со значениями физической яркости. Далее мы добавляем информацию об условиях освещения: цвет источника света, абсолютная фоновая яркость и пр. Физические яркости поточно преобразуются в воспринятые по модели CIECAM02. В конце значения суммируются в пределах объекта, границы которого задаются маской. Схема алгоритма представлена на Рис. 6.

5. Тестирование и результаты

Алгоритм оценивался экспериментально. Для проведения тестирования была сгенерирована тестовая база стимулов. Человеку предъявлялся стимул и однородное серое поле. Предлагалось подобрать яркость поля так, чтобы она была равна средней яркости стимула.

Проверялись результаты двух описанных алгоритмов определения интегральной яркости объекта.

Для каждого алгоритма подсчитывалась относительная разница между суммарными характеристиками стимула и однородного поля. В результате проведенного исследования было установлено, что точность простого усреднения физической яркости ошибка составляет всего лишь 15%, а с использованием адаптационного преобразования она снижается до 1%.

6. Заключение

В статье предложены 2 метода расчета интегральной яркости объекта:

- 1) Метод расчета объективной яркости.
- 2) Метод расчета субъективной яркости.

Метод расчета объективной яркости базируется на процессе не затратной калибровки фотоаппарата, представлена методика калибровки и тестовый объект.

Для расчета субъективной яркости, к изображению широкого диапазона применяется модель CIECAM02. Полученные в результате тестирования значения ошибок показали выигрыш предложенного алгоритма по сравнению с алгоритмом наивного усреднения объективной физической яркости.

Литература

- [1] Debevec P., Jitendra Malik. Recovering high dynamic range radiance maps from photographs. // Proceeding SIGGRAPH '08 ACM SIGGRAPH 2008 classes Article No. 31. 2008.
- [2] Hiscocks P, Eng P. Measuring Luminance with a Digital Camera. // Syscomp Electronic Design Limited. 2011.
- [3] Stevens, S.S. (1957). "On the psychophysical law". Psychological Review 64 (3): 153–181.
- [4] J. v. Kries (1905). Die Gesichtsempfindungen. Handbuch der Physiologie der Menschen.
- [5] Nayatani Y. Formulation of a Nonlinear Model of Chromatic Adaptation. // Color Research & Application Volume 6, Issue 3, pages 161–171, Autumn (Fall) 1981
- [6] RWG Hunt. Revised colour-appearance model for related and unrelated colours. // Color Research & Application Volume 16, Issue 3, pages 146–165, June 1991
- [7] Moroney N. The CIECAM02 color appearance model. // Color and Imaging Conference, 10th Color and Imaging Conference Final Program and Proceedings, pp. 23-27(5)

Метод преобразования динамического диапазона цифровых изображений с целью улучшения различимости объектов*

Д.В. Начаров, Ю.П. Михайлюк

denis_nacharov@list.ru|mypsev@gmail.com

Севастопольский государственный университет, Севастополь, Россия

В работе предложен метод преобразования динамического диапазона с целью улучшения различимости объектов, применимый к цифровым изображениям, полученным в условиях недостаточной видимости.

Ключевые слова: недостаточная видимость, контраст, динамический диапазон, преобразование яркости.

The method of digital images dynamic range transformation in purpose of improvement of object distinctiveness*

D. V. Nacharov, Y. P. Mikhayluck

Sevastopol State University, Sevastopol, Russian Federation

The method of dynamic range transformation in purpose of improvement of distinctiveness of objects on digital images recorded in adverse visibility conditions is presented.

Keywords: low visibility, contrast, dynamic range, intensity transformation

1. Введение

При регистрации изображений в условиях недостаточной видимости, вызванных такими метеороусловиями как туман, дымка и т.п., происходит снижение контраста и, как следствие, ухудшение различимости объектов сцены. Это приводит к снижению эффективности работы систем визуализации, и обработки изображений, предполагающих как визуальный, так и машинный анализ изображений, таких как системы видеонаблюдения и распознавания автомобильных номеров. Снижение контраста объектов сцены на изображениях, полученных в условиях недостаточной видимости, характеризуется пространственной неравномерностью: как правило, области изображения, отображающие удаленные объекты имеют малый динамический диапазон и характеризуются низким контрастом. В свою очередь области изображения, отображающие объекты ближней сцены, имеют больший динамический диапазон и контраст. Задача увеличения контраста и улучшения различимости объектов может рассматриваться как задача согласования динамических диапазонов различных областей изображения с динамическим диапазоном воспроизводящего устройства.

В работе предлагается метод улучшения различимости объектов сцены, предполагающий увеличение контраста путем преобразования динамического диапазона яркостей изображения.

2. Постановка задачи

Простейшим методом увеличения контраста изображений является линейное контрастирование [1].

Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347.

Исходное полутоновое изображение представляется как двумерный массив яркостей $I(x, y)$, $x = 1, \dots, M; y = 1, \dots, N; M, N$ – вертикальный и горизонтальный размеры изображения, а обработанное изображение – массив $s(x, y)$. Для общности изложения можно пользоваться значениями яркости, нормированными к числу градаций яркости L , в диапазоне $0 \dots 1$. Максимальный динамический диапазон изображения при этом будет равен 1. Линейное контрастирование является поэлементным преобразованием и описывается функцией преобразования яркости общего вида $s = T(I)$. Обозначим минимальную и максимальную яркости исходного изображения как I_{min} и I_{max} , а минимальную и максимальную яркости обработанного изображения – s_{min} и s_{max} . Функция преобразования яркости линейного контрастирования описывается выражением

$$s(x, y) = T(I) = \frac{I - I_{min}}{I_{max} - I_{min}} * (s_{max} - s_{min}) + s_{min}$$

Увеличение динамического диапазона изображения определяется крутизной преобразования яркости \varkappa .

На рис. 1 показаны изображение $I(x, y)$, полученное в условиях тумана, имеющее динамический диапазон $I_{min} \dots I_{max} = 0, 48$, нормированная гистограмма яркости $p(I)$ этого изображения и линейная функция преобразования яркости $T(I)$.

Крутизна функции преобразования составляет $\varkappa = 2, 11$. На рис. 2 показаны обработанное изображение $s(x, y)$ и его нормированная гистограмма яркости $p(s)$. Из визуального анализа рис. 2 следует, что обработанное изображение по-прежнему характеризуется низким контрастом удаленных объектов сцены.

На изображении рис.1 объекты дальней сцены располагаются в верхней половине изображения, а объекты ближней сцены – в нижней. Рассмотрим верхнюю и нижнюю части изображений $I(x, y)$ и $s(x, y)$, определенные как: верхние части изображений – $I_{top}(x, y) = I(x, 1 < y < N/2)$; $s_{top}(x, y) = s(x, 1 < y < N/2)$; нижние части изображений – $I_{bot}(x, y) = I(x, N/2 \leq y < N)$; $s_{bot}(x, y) = s(x, N/2 \leq y < N)$.

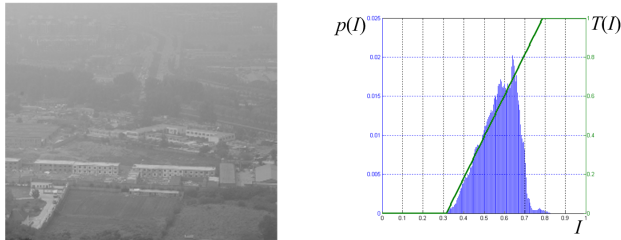


Рис. 1: Изображение $I(x, y)$, полученное в условиях тумана (слева), нормированная гистограмма $p(I)$ и функция преобразования яркости $T(I)$ (справа).

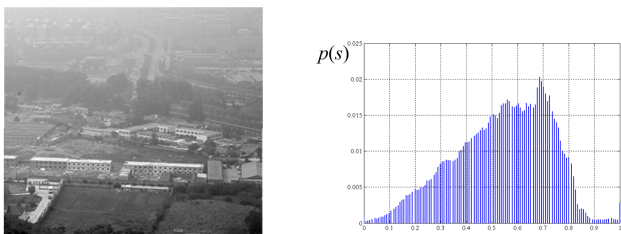


Рис. 2: Обработанное изображение $s(x, y)$ (слева) и его нормированная гистограмма $p(s)$ (справа).

Динамический диапазон изображения $I_{bot}(x, y)$ составляет 0,47, динамический диапазон изображения $I_{top}(x, y)$ – 0,26. В результате линейного преобразования яркости с крутизной $\kappa = 2,11$ динамический диапазон изображения $s_{bot}(x, y)$ близок к максимальному и составляет 0,99. Динамический диапазон изображения $s_{top}(x, y)$ составляет 0,56. Таким образом, линейное контрастирование изображений, полученных в условиях недостаточной видимости, не учитывает неравномерность динамических диапазонов и контрастов областей изображения, отображающих объекты, расположенные на различном удалении от регистрирующего устройства.

С учетом этого предлагается следующий подход к коррекции изображений, полученных в условиях недостаточной видимости. Диапазон яркостей исходного изображения разделяется на два поддиапазона, соответствующих ближней и дальней сценам изображения. В основе предлагаемого подхода к разделению диапазона яркости находится предположение о зависимости яркости объекта от его

удаленности: объекты с большей яркостью считаются более удаленными, чем объекты с меньшей яркостью. На основе исходного изображения формируются два изображения: изображение, содержащее поддиапазон малых яркостей (изображение ближней сцены) и изображение, содержащее поддиапазон больших яркостей (изображение дальней сцены). Сформированные таким образом изображения подвергаются процедуре линейного контрастирования, при этом оба поддиапазона яркостей преобразовываются на весь возможный диапазон яркостей. Затем выполняется процедура совмещения этих двух изображений.

3. Описание метода

Коррекция дальней сцены изображения

Уровни яркости изображений, полученных в условиях недостаточной видимости, располагаются в верхней части диапазона, что обусловлено наличием аддитивной составляющей рассеянного света [2]. Величина рассеянного света может оцениваться значением средней яркости изображения I_{mean} . Поэтому первым этапом коррекции дальней сцены изображения является вычитание средней яркости изображения. В результате этой операции формируется разностное изображение, динамический диапазон которого увеличивается путем умножения на масштабный коэффициент.

Яркость разностного изображения s_d описывается как

$$s_d = k_m(I - I_{mean}),$$

где $k_m = \frac{I_{max}}{I_{max} - I_{mean}}$ – масштабный коэффициент.

Эффективность использования всего диапазона яркостей можно повысить за счет линейного контрастирования разностного изображения. При этом входной диапазон преобразования может быть ограничен уровнями яркости, наиболее часто встречающимися на изображении. Для определения ограниченного диапазона яркостей используется интегральная функция распределения яркостей $F(s_d)$, определяемая как

$$F(s_{dj}) = \sum_{j=1}^L p(s_{dj}).$$

Значение функции $F(s_{dj})$ от j -го уровня яркости s_d характеризует количество пикселей, яркость которых находится в диапазоне яркостей $0 \dots s_{dj}$.

На рис. 3 показана интегральная функция распределения $F(s_d)$ разностного изображения. Из рис. 3 видно, что 99% пикселей разностного изображения имеют яркость в диапазоне $0 \dots 0,69$. Таким образом, за счет исключения редко встречающихся уровней яркости, можно значительно увеличить динамический диапазон разностного изображения.

На рис. 4 показаны изображение, полученное в результате линейного контрастирования разностного изображения с ограничением входных уровней, и его нормированная гистограмма.

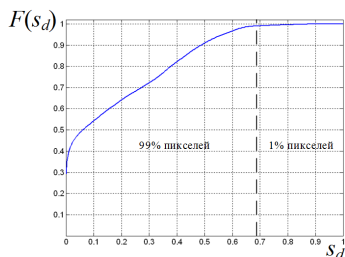


Рис. 3: Интегральная функция распределения яркости s_d .

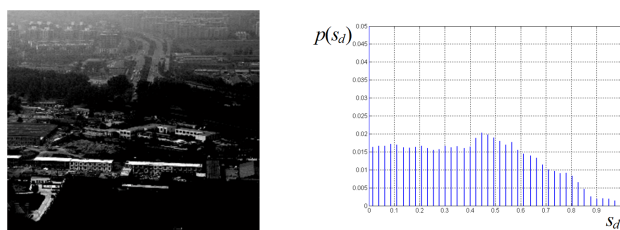


Рис. 4: Изображение, полученное в результате линейного контрастирования изображения $s_d(x, y)$ (слева), и его нормированная гистограмма $p(s_d)$ (справа).

На рис. 4 объекты ближней сцены с яркостью, меньшей I_{mean} , на изображении отсутствуют. Таким образом, достигается селекция объектов, находящихся на разном удалении от регистрирующего устройства.

Коррекция ближней сцены изображения В результате коррекции дальней сцены на разностном изображении присутствуют уровни яркости в диапазоне $I_{mean} \dots I_{max}$. Коррекция ближней сцены изображения заключается в линейном контрастировании оставшегося поддиапазона яркостей $I_{min} \dots I_{mean}$ и описывается выражением

$$s_n = T(I) = \begin{cases} \frac{I - I_{min}}{I_{mean} - I_{min}}, & \text{при } I_{min} < I < I_{mean}; \\ 1, & \text{при } I \geq I_{mean}. \end{cases}$$

На рис. 5 показаны результаты линейного контрастирования изображения ближней сцены $s_n(x, y)$ и его нормированная гистограмма $p(s_n)$. Для повышения эффективности преобразования входной диапазон также ограничен 99% наиболее часто встречающимися уровнями яркости.

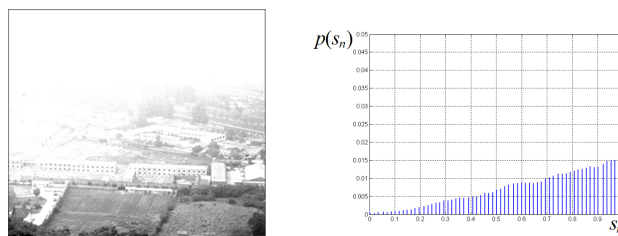


Рис. 5: Изображение ближней сцены $s_n(x, y)$ (слева) и его нормированная гистограмма $p(s_n)$ (справа).

Таким образом, в результате линейного контрастирования поддиапазонов ближней и дальней сцен исходного изображения $I(x, y)$ сформированы два изображения $s_n(x, y)$ и $s_d(x, y)$ с динамическими диапазонами, равными 1. Следующим этапом предлагаемой обработки является совмещение изображений ближней и дальней сцен.

Совмещение изображений ближней и дальней сцен

Совмещение изображений ближней и дальней сцен заключается в представлении диапазонов яркостей этих изображений в одном общем диапазоне яркостей. Процедура совмещения заключается в суммировании изображений ближней и дальней сцен, умноженных на весовые коэффициенты. Яркость суммарного изображения s_Σ будет описываться выражением

$$s_\Sigma = w_g s_d + (1 - w_g) s_n,$$

где w_g – «глобальный» весовой коэффициент.

Коэффициент w_g назван «глобальным», поскольку его значение одинаково для всех пикселей изображения. Значение коэффициента w_g определяет вклад яркостей изображений ближней и дальней сцен в яркость суммарного изображения. При $w_g > 0,5$ суммарное изображение в большей степени определяется изображением дальней сцены, при $w_g < 0,5$ – изображением ближней сцены. На рис. 6 показаны результаты совмещения изображений ближней и дальней сцен при различных значениях коэффициента w_g .

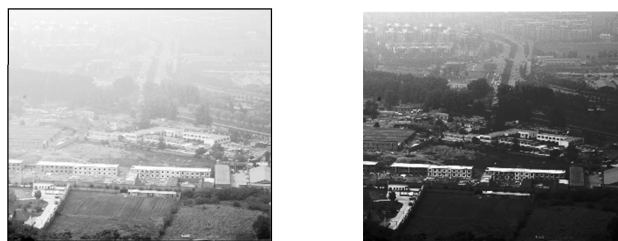


Рис. 6: Результаты совмещения изображений ближней и дальней сцен при значениях «глобальных» весовых коэффициентов $w_g = 0,2$ (слева), $w_g = 0,8$ (справа).

Как видно из рис. 6, применение «глобального» весового коэффициента w_g не позволяет учитывать пространственную неравномерность ухудшения контраста, свойственную изображениям, полученным в условиях недостаточной видимости. Использование коэффициента w_g нивелирует выигрыш от увеличения динамического диапазона различных поддиапазонов яркости за счет усреднения с экстремальными значениями яркости изображений $s_d(x, y)$ и $s_n(x, y)$.

Для наилучшего отображения обоих поддиапазонов яркости следует применять весовые коэффициенты, определяемые значениями и пространственным распределением яркости изображения $I(x, y)$. В области больших яркостей следует выбирать большие весовые коэффициенты для изображения дальней сцены, а в области малых яркостей – для изображения ближней сцены. Для учета локальных свойств изображения $I(x, y)$ вместо «глобального» коэффициента w_g может применяться массив локальных весовых коэффициентов $w_l(x, y)$ с размерами, равными размерам $I(x, y)$.

В качестве массива весовых коэффициентов $w_l(x, y)$ используется локально усредненные яркости исходного изображения $\bar{I}(x, y)$, нормированные к максимальному значению \bar{I}_{max}

$$w_l(x, y) = \frac{\bar{I}(x, y) - \bar{I}_{min}}{\bar{I}_{max}}.$$

Локальное усреднение осуществляется с помощью сглаживающей пространственной фильтрации. Изображение $\bar{I}(x, y)$ определяется в результате свертки изображения $I(x, y)$ с маской фильтра $g(x, y)$ и описывается как

$$\bar{I}(x, y) = I(x, y) * g(x, y) = \sum_{\tau=-\frac{A-1}{2}}^{\frac{A-1}{2}} \sum_{\nu=-\frac{B-1}{2}}^{\frac{B-1}{2}} g(\tau, \nu) I(x + \tau, y + \nu),$$

где τ, ν – переменные сдвига; A, B – размеры маски сглаживающего фильтра.

Маска фильтра $g(x, y)$ определяется выражением

$$g(x, y) = \frac{1}{AB} \begin{pmatrix} 1 & \dots & 1 \\ \dots & 1 & \dots \\ 1 & \dots & 1 \end{pmatrix}.$$

Соотношение размеров маски фильтра A, B выбирается таким же, как и соотношение размеров изображения M, N . Для известных M, N достаточно задавать только один, например, горизонтальный размер маски B . Суммарное изображение $s_\Sigma(x, y)$ определяется выражением

$$s_\Sigma(x, y) = w_l(x, y) s_d(x, y) + [1 - w_l(x, y)] s_n(x, y).$$

На рис. 7 показан весовой массив $w_l(x, y)$ при $B = 10$ и соответствующее ему изображение $s_\Sigma(x, y)$. Применение сглаживающего фильтра с размерами A, B эквивалентно снижению пространственного разрешения изображения в AB раз. Массив весовых коэффициентов $w_l(x, y)$ является изображением $I(x, y)$ с подавленными верхними пространственными частотами. Выражение для $s_\Sigma(x, y)$ содержит как коэффициенты $w_l(x, y)$, так и инверсные коэффициенты $[1 - w_l(x, y)]$. Инверсия весовых коэффициентов приводит к обратному эффекту – происходит подавление нижних частот и усиление верхних частот изображения $s_n(x, y)$. При малых B это проявляется в виде изменения значения постоянной составляющей и повышения резкости контуров в области ближней сцены изображения $s_\Sigma(x, y)$ (см. рис.7).



Рис. 7: Массив $w_l(x, y)$ при $B = 10$ (слева) и соответствующее ему изображение $s_\Sigma(x, y)$ (справа).

Представим изображение $s_n(x, y)$ в виде суммы низкочастотной $s_n^{LF}(x, y)$ и высокочастотной $s_n^{HF}(x, y)$ составляющих

$$s_n(x, y) = s_n^{LF}(x, y) + s_n^{HF}(x, y).$$

При $B = 10$, что соответствует отсутствию сглаживания, получим

$$\begin{aligned} [1 - w_l(x, y)] s_n(x, y) &= \\ &= [1 - I^{LF}(x, y)] [s_n^{LF}(x, y) + s_n^{HF}(x, y)] = \\ &= s_n(x, y) - I^{LF}(x, y) s_n^{LF}(x, y) - I^{LF}(x, y) s_n^{HF}(x, y). \end{aligned}$$

Поскольку к изображениям применяются только линейные яркостные преобразования, то спектры изображений $I(x, y)$ и $s_n(x, y)$ одинаковы по составу пространственных частот. Поэтому, без учета амплитудных соотношений, можно записать

$$I^{LF}(x, y) \approx s_n^{LF}(x, y).$$

Тогда, при $B = 0$, получим

$$\begin{aligned} [1 - w_l(x, y)] s_n(x, y) &= \\ &= s_n(x, y) - s_n^{LF}(x, y)^2 - I^{LF}(x, y) s_n^{HF}(x, y) \end{aligned}$$

При $B = N$, что соответствует снижению пространственного разрешения в MN раз, изображе-

ние $I^{LF}(x, y)$ содержит только постоянную составляющую, и тогда получим

$$\begin{aligned} w_l(x, y) &= I^{LF}(x, y) = const = w_g; \\ [1 - I^{LF}(x, y)] &= const \leq 1; \\ [1 - w_l(x, y)]s_n(x, y) &= (1 - w_g)s_n(x, y). \end{aligned}$$

Таким образом, при $B = N$ все коэффициенты массива $w_l(x, y)$ равны между собой и его применение эквивалентно применению глобального весового коэффициента w_g .

Для снижения эффекта подавления нижних частот изображения $s_n(x, y)$ и, в то же время, сохранения пространственного распределения значений коэффициентов массива $w_l(x, y)$, размер маски фильтра B должен выбираться из интервала $(0, N)$. Анализ относительно большого числа изображений, полученных в условиях недостаточной видимости, показал, что в силу слабоконтрастной природы таких изображений и малой ширины спектра пространственных частот уже при $B = 0, 1N$ обеспечивается значительное снижение эффекта ослабления нижних частот. Массив $w_l(x, y)$ при $B = 0, 1N = 60$ и соответствующее ему изображение $s_\Sigma(x, y)$ показаны на рис. 8.

Итоговый результат применения предложенного метода преобразования показан на рис.9. Динамические диапазоны верхней $s_{\Sigma t}(x, y) = s_\Sigma(x, 1 < y < N/2)$ и нижней $s_{\Sigma b}(x, y) = s_\Sigma(x, N/2 \leq y < N)$ частей изображения $s_\Sigma(x, y)$ равны 0,88 и 0,99, соответственно.



Рис. 8: Массив $w_l(x, y)$ при $B = 60$ (слева) и соответствующее ему изображение $s_\Sigma(x, y)$ (справа).



Рис. 9: Исходное изображение, полученное в условиях тумана (слева) и результат применения предложенного метода преобразования (справа).

В результате применения предложенного метода увеличены динамические диапазоны и контрасты различных областей обрабатываемого изображения, в следствие чего улучшена различимость объектов сцены.

4. Заключение

В работе предложен метод преобразования динамического диапазона с целью улучшения различимости объектов, применимый к изображениям, полученным в условиях недостаточной видимости. На примере изображения, полученного в условиях тумана, проведены качественная и количественная оценки эффективности предложенного метода.

Литература

- [1] Грузман И.С., Киричук В.С., Косых В.П., Перетягин Г.И., Спектор А.А. Цифровая обработка изображений в информационных системах. - Новосибирск: Изд-во НГТУ, 2002. - 352 с.
- [2] Mikhailyuk Y.P., Nacharov D.V. The method of correction of digital television images recorded in bad weather // Proc. of 21st International Crimean Conference «Microwave & Telecommunication Technology». - Sevastopol, 2011. - pp.1122-1123.

Об авторах

Начаров Денис Владимирович — младший научный сотрудник кафедры радиоэлектроники и телекоммуникаций Севастопольского государственного университета.

Михайлюк Юрий Петрович — к.т.н, доцент, доцент кафедры радиоэлектроники и телекоммуникаций Севастопольского государственного университета.

Структурная оптимизация алгоритма распознавания на базе КЭКМ*

А.Д. Рябушева¹, Л.И. Лебедев²

lebedev@pmk.unn.ru | pmk@unn.ac.ru

¹Факультет вычислительной математики и кибернетики,

²НИИ прикладной математики и кибернетики,

Нижегородский государственный университет им. Н.И. Лобачевского

В работе описывается алгоритм распознавания с самообучением графических изображений на базе корреляционно-экстремального контурного метода (КЭКМ) с использованием иерархического классификатора древовидной структуры. Специфика построения классификатора состоит в том, что его структура формируется на основе кластерного анализа неопознанных объектов, пополняющих список эталонов, и ведется в автоматическом режиме в процессе распознавания. Построение классификационного дерева позволяет существенно понизить вычислительную сложность алгоритма распознавания с самообучением.

Ключевые слова: распознавание с самообучением, КЭКМ, иерархический классификатор, кластер, оценка сходства, объект изображения, признаки описания, образ, эталон.

Structural Optimization of Recognition Algorithm on the Basis CECM*

A.D. Rybusheva¹, L.I. Lebedev²

¹Faculty of Computational Mathematics and Cybernetics,

²Research Institute of Applied Mathematics and Cybernetics,

Lobachevsky State University of Nizhni Novgorod

The paper describes a recognition algorithm of graphic images with self-learning on the basis of correlation-extreme contour method (CECM) using a hierarchical tree structure of the classifier. Specifics of the classifier consists in the fact that its structure is formed on cluster analysis of unrecognized objects, replenishing the list of standards, and is conducted in automatic mode in the recognition process. Constructing a classification tree can significantly reduce the computational complexity of the algorithm recognition with self-learning.

Keywords: recognition of self-learning, CECM, hierarchical classifier, cluster, evaluation similarity, object of image, signs descriptions, pattern, reference.

1. Введение

Корреляционно-экстремальные контурные методы распознавания базируются на вычисление оценок сходства эталонов с объектом, инвариантных относительно ортогональных преобразований и масштабирования (ОПМ) [1, 2]. Так как КЭКМ относятся к методам сравнения объекта с эталонами, поэтому их быстродействие зависит, в том числе, и от количества предъявляемых для распознавания образцов. В условиях большого разнообразия объектов на изображении (количество классов сотни, тысячи и более) требование к снижению временной сложности алгоритма распознавания на базе КЭКМ по-прежнему остается актуальным. Поэтому, оптимизация временной сложности алгоритма распознавания предполагает либо возможность принятия решения на основе частично вычисленных оценок близости, либо использование технологий и средств параллельного программирования. Естественное решение этой задачи заключается в разбиении потока последовательного вычисления

оценок близости на несколько потоков, в каждом из которых используется только некоторая часть исходного состава эталонов. При этом решение о принадлежности объекта к тому или иному классу может приниматься как по результатам вычислений внутри одного потока, так и по совокупности результатов вычислений во всех потоках. Рассмотрим два направления в разбиении последовательного потока вычисления оценок близости.

Наиболее простой по исполнению является возможность использования многоядерных компьютеров для распараллеливания вычислений оценок близости объекта с эталонами. Этот способ распараллеливания алгоритма распознавания на базе КЭКМ является наиболее эффективным при распределении вычислений по выбранной группе шрифтов и использовании суперкомпьютеров (то есть без существенного ограничения числа ядер).

Второе направление связано с построением классификатора с древовидной структурой для некоторой группы эталонов, например, для группы, представляющей символы какого-либо шрифта. В этом случае сложность распознавания будет определяться только суммарной сложностью вычисления всех

Работа выполнена и опубликована при поддержке РФФИ, гранты 13-07-00521, 15-07-20347.

оценок сходства с эталонами данного листа классификационного дерева.

Для решения задачи классификации некоторого множества эталонов необходимо сформировать признаковое пространство для их описания. Очевидно, что получение компонент этого пространства должно быть быстродействующей процедурой, значительно менее затратной, чем вычисление оценки сходства эталона с объектом. В нашем случае такая система признаков может быть сформирована на основании параметров, полученных от согласования описаний эталона и объекта, необходимого для вычисления оценки сходства. Для согласования описаний используются методы относительных смещений и метод парабол [3]. Метод относительных смещений основан на вычислении оценок близости $\varepsilon_m(\mathbf{B}, \mathbf{E}_j, s = s_i)$ базового эталона \mathbf{B} с эталоном \mathbf{E}_j при различных положениях начальной точки описания контура $s = s_i$. Местоположения начальных точек берутся в порядке обхода контура эталона с шагом приращения ΔS по длине контура. Значение находится ΔS из выражения $\Delta S = S^j/n$, где S^j - длина контура, n - число точек для получения различных описаний контура эталона. Тогда описание эталона \mathbf{E}_j можно задать системой признаков $c_{ij} = \varepsilon_m(\mathbf{B}, \mathbf{E}_j, s_i)$, $i = 1, 2, \dots, n$, где $s_i = i \cdot \Delta S$. Для того, чтобы предлагаемая система признаков была инвариантна не только относительно ОПМ, но относительно исходного описания эталона (объекта) циклической перестановкой сделаем так, чтобы первый признак был наименьшей величиной, то есть $c_{j1} = \min c_{ji}$, $i = 1, 2, \dots, n$. На рисунке 1 представлены графики оценок близости с базовым эталоном $\varepsilon_m(\mathbf{B}, \mathbf{E}_j, s)$ некоторых символов топографического шрифта Т-132, полученных при малых значениях шага ΔS . Из соображений быстродействия для вычисления оценок близости $\varepsilon_m(\mathbf{B}, \mathbf{E}_j, s = s_i)$ в качестве базового эталона был определен треугольник Пифагора (только три точки в исходном описании контура). На этом же рисунке отмечены значения четырех аргументов $s = \nu_q$, в которых вычисленные оценки близости $\varepsilon_m(\mathbf{B}, \mathbf{E}_j, s = \nu_q)$ обеспечивают решение задачи классификации шести представленных символов.

2. Постановка задачи

Отметим требования, которые должны выполняться при решении задачи построения классификатора. Во-первых, число признаков описания каждого из эталонов определяется количеством вычисляемых оценок близости в методе относительных смещений, то есть $n = 20$. Во-вторых, должны быть использованы только методы автоматической классификации, так как состав эталонов заранее неизвестен и определяется в процессе распознавания с самообучением. В третьих, по мере уве-

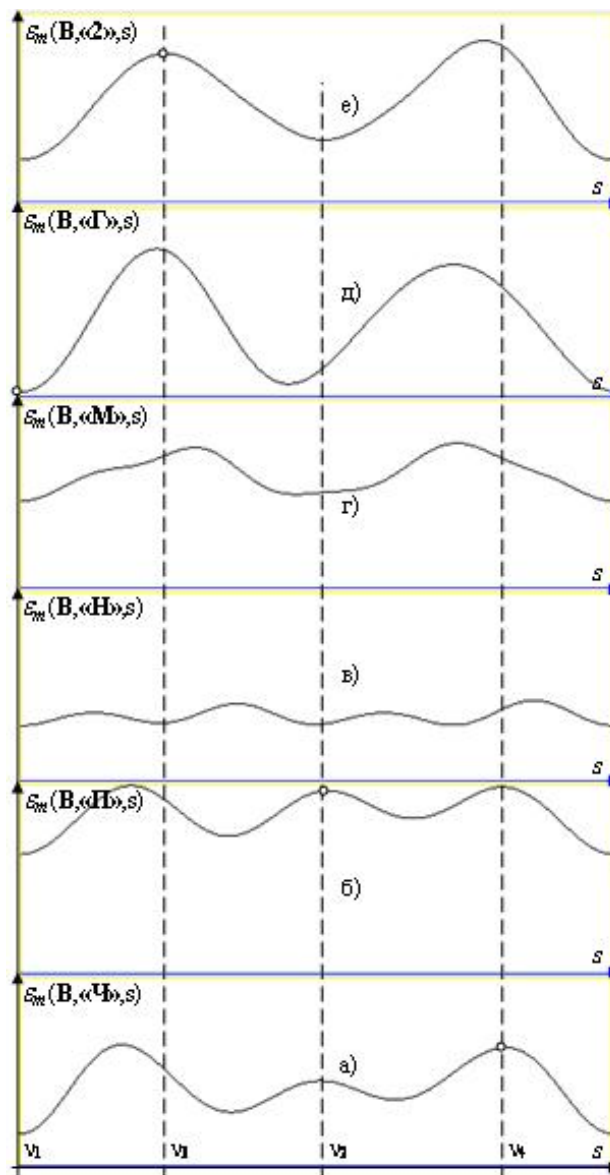


Рис. 1: Графики оценок близости с базовым эталоном.

личения числа эталонов для обеспечения быстродействия алгоритма распознавания должна иметься возможность перестроения (обновления) полученного классификатора в целях уменьшения числа представителей на листах дерева. В-четвертых, построение классификатора должно вестись одновременно с процедурой распознавания. Наконец, временная сложность построения классификатора должна быть ниже, чем сложность распознавания с использованием того множества эталонов, для которого строится или обновляется классификатор.

3. Методы решения

В результате исследования, проведенного в целях выбора алгоритма построения классификатора, удовлетворяющего сформулированным вы-

ше требованиям, был взята следующая технология разбиения множества эталонов на кластеры и построения решающих правил. В качестве способа разбиения множества на кластеры был взят алгоритм нечёткой кластеризации методом k ближайших соседей (k -means) [4]. В качестве входных параметров алгоритм k -means следующие данные:

- Тренировочная выборка.
- Количество кластеров.
- Максимальное количество итераций.
- Метод вычисления расстояния.

При разбиении заданной выборки эталонов на кластеры в любой вершине классификационного дерева их возможное количество задавалось равным трем. В этом случае сложность разбиения множества точек объемом v при ограничении на количество итераций числом l будет определяться выражением $O(3 \cdot v \cdot l)$.

Для построения решающих правил был выбран алгоритм SVM (support vector machine) - машина опорных векторов с полиномиальным ядром отображения в спрямляющее пространство признаков [5]. Входными данными для алгоритма SVM являются кластеры, полученные в результате применения алгоритма k -means. Алгоритм SVM при построении разделяющей гиперплоскости максимизирует расстояние между опорными векторами (ближайшими точками кластеров в спрямляющем пространстве). Построение решающих правил осуществляется независимо для каждой пары кластеров (дихотомия). Сложность построения решающих правил в каждой вершине зависит только от объема выборки в ней и оценивается величиной $O((3 \cdot v)^4)$.

На базе полученных алгоритмом SVM расстояний между кластерами в данной вершине классификатора могут быть созданы две или три дочерние вершины в зависимости от результатов проведенного разделения множеств:

- Расстояния между кластерами **A**, **B** и **C** удовлетворяют заданному порогу - образуются три дочерних вершины с формированием обучающих выборок на основе точек этих кластеров;
- Расстояние между одним из кластеров **A** и объединением двух других **B ∪ C** больше δ (расстояние между объединяемыми кластерами было меньше δ) - образуются две дочерних вершины с формированием обучающих выборок кластеров **A** и **B ∪ C**;
- Расстояние только между двумя из кластеров **A** и **B** больше δ - образуются две дочерних вершины с формированием обучающих выборок кластеров **A ∪ C** и **B ∪ C**;
- Все расстояния между кластерами меньше δ - вершина объявляется листовой.

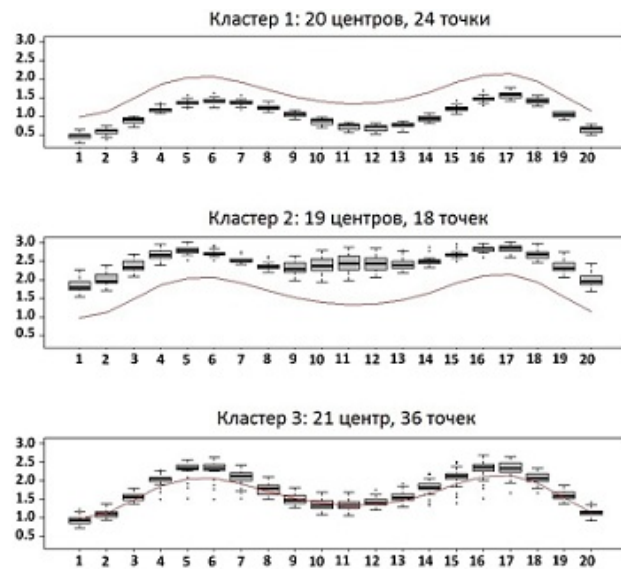


Рис. 2: Кластеры начальной классификации.

При поступлении нового эталона определяется листовая вершина классификатора, которой он будет принадлежать. Со временем, накопление эталонов, приводит к тому, что объект, поступивший на распознавание, будет попадать в лист с большим количеством эталонов. Это повлечет снижение быстродействия алгоритма распознавания, и следовательно, потребует необходимость перестройки классификатора. Для каждой обремененной листовой вершины возможны два варианта обновления классификатора. Если уровень данного листа и количество эталонов не входящих в его кластер допускают создание дочерних вершин, то вершина считается висячей и доработка классификатора производится обычным образом. Если количество объектов не входящих в кластер данного листа меньше требуемого, то будет рассматриваться предок этой вершины. В ней объединяются в кластер все эталоны, не принадлежащие кластерам листов всех ветвей классификатора. В итоге, всплытием вверх по дереву находится вершина, для которой имеется возможность построения решающих правил для выделения точек вновь образованного кластера. При этом из списка эталонов всех листов нижних уровней данной вершины удаляются точки нового кластера.

4. Полученные результаты

Описанный алгоритм получения иерархического классификатора с древовидной структурой был апробирован на информации о символах топологического шрифта T-132. Исходное количество эталонов включало 78 символов. В результате построенный классификатор содержит 35 листовых вершин с максимальной глубиной из 10-и уровней и минимальной глубиной поддеревя из 4-х уровней.

Среднестатистическая глубина составила 7.37 уровней. 50 эталонов в дереве принадлежат только одному листу, 15 эталонов отметились присутствием на двух листах, а для классификации 12 эталонов использовались три ветви дерева и наконец, один эталон присутствует на четырех листах дерева. На рисунке 2 показано разбиение исходного множества эталонов на три кластера.

5. Заключение

Проведенный анализ сложности алгоритма построения и перестройки классификатора показал, что они могут быть вписаны в процедуру распознавания при существенном увеличении его быстродействия. Если предположить, что каждый эталон используется для опознавания в среднем 40 объектов, то сравнение быстродействия алгоритма распознавания с алгоритмом построения и перестройки классификатора показало, что наибольшая эффективность от применения классификатора достигается, если обновление совершается, когда число новых эталонов будет в диапазоне от 150 до 200. При этом быстродействие распознавания увеличивается в среднем на порядок для графических изображений, содержащих около 45000 объектов при накоплении до 1100 эталонов.

Литература

- [1] *Васин Ю.Г., Лебедев Л.И., Пучкова О.В.* Контурные корреляционно-экстремальные методы обнаружения и совмещения объектов видеоинформации. // Автоматизация обработки сложной графической информации: Межвуз.темат.сб.науч.тр. // Под ред. Ю.Г.Васина.- Горьков.гос.ун-т, Горький, 1987. С.97-112.
- [2] *Васин Ю.Г., Лебедев Л.И.* Задача нахождения согласованных описаний в корреляционно-экстремальных контурных методах распознавания. // Математические методы распознавания образов (ММРО-15): 15-ая Всеросс. конф.: Сборник докладов. / М.: Изд-во ООО "МАКС Пресс 2011. – С.342-345.
- [3] *Лебедев Л.И.* Корреляционно - экстремальные контурные методы распознавания. Теоретические основы: Учебное пособие. – Нижний Новгород, Изд-во Нижегородского государственного университета, 2013. – 113 с.
- [4] http://www.basegroup.ru/glossary/definitions/k_means/. – BaseGroup Lab – 2015.
- [5] <http://www.basegroup.ru/glossary/definitions/svm/> – BaseGroup Lab – 2015.

Visual Graph – система визуализации сложно структурированной информации большого объема на основе графовых моделей *

В.Н. Касьянов^{1,2}, *Т.А. Золотухин*¹
kvn@iis.nsk.su|tzolotuhin@gmail.com

¹Институт систем информатики им. А.П. Ершова СО РАН, Новосибирск, Россия;

²Новосибирский государственный университет, Новосибирск, Россия

Доклад посвящен создаваемой в лаборатории конструирования и оптимизации программ ИСИ СО РАН универсальной системе Visual Graph для визуализации сложно структурированной информации большого объема на основе графовых моделей.

Ключевые слова: *атрибутированные иерархические графы, визуализация графов большого размера, визуализация информации на основе графовых моделей, интерактивная визуализация, системы визуализации.*

Visual Graph – a system for visualization of big size complex structural information on the base of graph models*

V.N.Kasyanov^{1,2}, *T.A.Zolotuhin*¹

¹Institute of Informatics Systems, Novosibirsk, Russia;

²Novosibirsk State University, Novosibirsk, Russia

The paper is devoted to the Visual Graph system being under development by the laboratory for program construction and optimization of Institute of Informatics Systems as an universal system for visualization of big size complex structural information on the base of graph models.

Keywords: attributed hierarchical graphs, big size graph visualization, information visualization based on graph models, interactive visualization, visualization systems.

Введение

Визуализация информации играет большую роль в жизни человека. Считается, что около 90% всей получаемой информации человек получает через зрение. Человечество за тысячи лет преодолело путь от простейших способов визуализации в виде наскальных рисунков до карт, схем и диаграмм. В настоящее время визуализация на основе теоретико-графовых моделей – неотъемлемый элемент обработки сложной информации о строении объектов, систем и процессов во многих приложениях в науке и технике [1-5]. На рынке широко представлены наукоемкие программные продукты, использующие методы визуализации информации на основе графовых моделей. В основном это многочисленные специализированные системы или встроенные компоненты систем, ориентированные на визуализацию определенных подклассов графовых моделей и/или специальные применения, но есть и универсальные системы, предназначенные для визуализации графов общего вида и широкого назначения, такие как Nigres [6, 7], aiSee [8], yEd [9] и Cytoscape [10].

Поскольку информация, которую желательно визуализировать, постоянно увеличивается и усложняется, возникает все больше ситуаций, в которых

классические графовые модели перестают быть адекватными. Требуются более мощные теоретико-графовые формализмы для представления информационных моделей, обладающих иерархической структурой. Иерархичность является основой многочисленных методов анализа и синтеза сложных информационных моделей в различных областях применения вычислительных систем. Одним из таких формализмов являются иерархические графы и графовые модели [11, 12].

В данном докладе будет рассмотрена универсальная система визуализации атрибутированных иерархических графов Visual Graph, которая распространяется под лицензией BSD и создается в рамках ведущегося в лаборатории конструирования и оптимизации программ ИСИ СО РАН проекта по разработке методов и средств для визуализации сложно структурированной информации большого объема на основе графовых моделей.

Область применения

При создании экспериментальной версии системы визуализации атрибутированных иерархических графов Visual Graph мы рассматривали в качестве основного ее использование разработчиками систем конструирования программ (таких как компилятор) для визуализации структур данных, возникающих при работе этих систем.

Структуры данных, возникающие в таких системах, естественным образом представляются в ви-

Работа выполнена при финансовой поддержке РФФИ, грант 15-07-02029 и опубликована при финансовой поддержке РФФИ, грант 15-07-20347.

де теоретико-графовых моделей большого размера. Так, например, синтаксические деревья используются в качестве внутреннего представления транслируемых программ почти во всех компиляторах или интерпретаторах. Оптимизирующие и реструктурирующие компиляторы требуют выявление в транслируемой программе управляющих и информационных связей и их представление в виде более общих теоретико-графовых моделей, таких как, например, управляющий граф.

Номенклатура этих графовых моделей в той или другой системе конструирования может своя, отличная от других. Более того, представления одинаковых графовых моделей могут различаться между собой в разных системах и даже в разных версиях одной и той же системы. Однако все эти теоретико-графовые модели являются частными случаями атрибутированных иерархических графов.

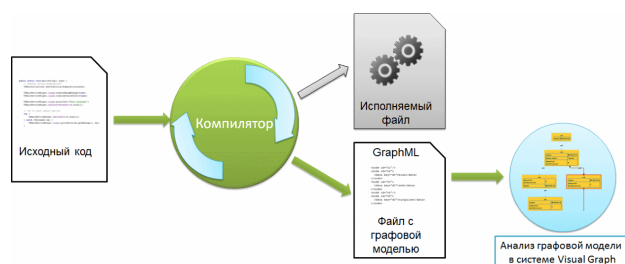


Рис. 1: Использование системы Visual Graph

Поэтому предполагается следующий сценарий использования системы Visual Graph. Сначала компилятор (сам либо с помощью вспомогательной программы) переводит графовую модель из внутреннего представления в файл одного из поддерживаемых системой Visual Graph форматов, как правило, GraphML-файл (Рис. 1). После этого система Visual Graph сможет прочитать эту графовую модель из файла, визуализировать ее и предоставить пользователю средства навигации по ней.

Иерархические графы и графовые модели

Рассмотрим ряд определений из [12].

Пусть G обозначает граф произвольного вида, элементы (вершины и ребра) которого отличаются один от другого какими-либо пометками, называемыми их *именами*, например, G может быть неориентированным или ориентированным графом, мультиграфом (с кратными ребрами) или псевдографом (с петлями).

Граф C называется *фрагментом* графа G , обозначаем $C \subseteq G$, если C – часть графа G , т. е. C образован подмножеством элементов графа G .

F – *иерархия фрагментов* графа G , если F – такое множество фрагментов графа C , что $G \in F$

и для любых двух фрагментов C_1 и C_2 из F либо фрагменты C_1 и C_2 не пересекаются, либо один из них является частью (*подфрагментом*) другого. Фрагмент G – *основной* (*главный*) фрагмент иерархии F . Фрагмент $C \in F$ – *элементарный*, если в F нет фрагментов G , являющихся подфрагментами фрагмента C .

Пусть задана некоторая иерархия фрагментов F графа G . Для любых $C_1, C_2 \in F$ фрагмент C_1 – *прямой* подфрагмент C_2 (или, что то же самое, фрагмент, *непосредственно* вложенный в C_2), если C_1 – подфрагмент C_2 и не существует такого $C_3 \in F$, отличного от C_1 и C_2 , что $C_1 \subseteq C_3 \subseteq C_2$.

Иерархический граф $H = (G, T)$ состоит из графа G и корневого дерева T , вершины которого соответствуют элементам некоторой иерархии в G , а дуги отражают отношение их непосредственной вложенности. T называется *деревом вложенности*, а G – *основным графом* иерархического графа H .

Важный частный случай иерархических графов образуют так называемые *простые* иерархические графы, в которых все фрагменты являются подграфами основного графа. В частности, он содержит составные и кластерные графы.

Поскольку подграфы однозначно определяются множествами своих вершин, есть возможность определять простой иерархический граф H как пару (G, T) , состоящую из основного графа G и *вершинного дерева вложенности* T , удовлетворяющего следующим условиям. Вершины дерева T соответствуют некоторым подмножествам вершин основного графа G таким образом, что подмножества вершин, соответствующие листьям T , образуют разбиение множества всех вершин графа G на одноэлементные подмножества. Дуги дерева T отражают непосредственную вложенность соответствующих подмножеств. Такое представление иерархического графа называется *вершинным* (Рис. 2).

Под графовой моделью в общем случае мы понимаем класс графовых объектов, имеющих вид атрибутированных (помеченных) графов, с заданным на нем отношением эквивалентности [12].

При этом при задании графовой модели будем различать статическую (или синтаксическую) часть описания, определяющую класс помеченных графов, образующих указанную модель, и динамическую (или семантическую) часть, задающую разбиение данного класса графов на подклассы попарно эквивалентных.

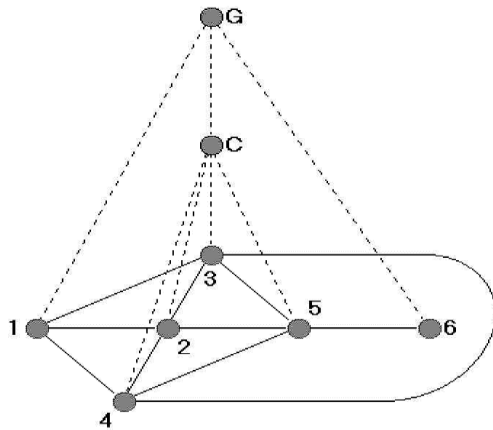


Рис. 2: Вершинное представление простого иерархического графа с двумя нетривиальными фрагментами: сплошные линии – это ребра основного графа, штриховые линии – ребра вершинного дерева вложенности

Пусть имеется множество объектов V , называемых *метками*, распадающееся на попарно непересекающиеся подмножества *классов* меток. В качестве классов меток могут использоваться определенные множества чисел, символов, строк (цепочек символов), формул, графов и объектов других видов.

Пусть задано множество объектов W , называемых *типами*, и пусть каждому элементу $w \in W$ поставлено в соответствие множество *пометок* $V(w)$, имеющее вид декартового произведения $V_{i_1} \times V_{i_2} \times \dots \times V_{i_s}$, где $V_{i_j} \in V$ – некоторый класс меток для любого j .

Со статической точки зрения *иерархическая графовая модель* – это тройка (H, M, L) , где H – иерархический граф, M – *функция типа*, приписывающая каждому элементу (вершине, ребру и фрагменту) h иерархического графа H его тип $M(h) \in W$, а L – *функция меток*, приписывающая каждому элементу h графа H его пометку – некоторый элемент $L(h) \in V(M(h))$.

При изображении графовой модели тип ее элементов может быть связан с определенной геометрической формой соответствующих представлений и/или их цветовой гаммой, а также местом и способом представления пометок, относящихся к элементам соответствующего типа.

Что касается динамической части иерархической графовой модели, то она привносит в визуализацию графовых моделей различные анимационные аспекты.

Обычно используется два основных подхода к заданию семантической части графовой модели: путем явного задания набора *инвариантов* (свойств,

присущих всем эквивалентным между собой моделям), который различает классы эквивалентности графовых моделей, либо через так называемые *эквивалентные* преобразования графовых моделей, которые сохраняют указанный набор инвариантов. Оба подхода к заданию семантической части графовой модели опираются на преобразования графов и активно развиваются в рамках теории схем программ.

Язык GraphML

Долгое время среди используемых форматов представления графов не находилось ни одного, который был бы достаточно широко принятым в качестве стандартного. По-существу, инструменты работы с графами поддерживали (да и сейчас многие из них поддерживают) лишь некоторую часть из существующих клиентских форматов, обычно состоящую из видов графовых представлений, ограниченных по выразимости и специфике конкретной областью применения.

Поэтому не случайно в 2000 г. наблюдательный комитет симпозиума по рисованию графов (Graph Drawing Steering Committee) инициировал работы по выработке формата обмена графами GraphML, который, в конечном счете, должен лечь в основу стандарта описания графовых моделей.

Основную цель, которую ставили перед собой разработчики языка, они сформулировали следующим образом. Формат обмена графами должен быть в состоянии представлять произвольные графы с произвольными дополнительными данными, включая укладку и графическую информацию. Дополнительная информация должна сохраняться в формате, подходящем для заданного конкретного приложения, но не должна усложнять представление данных из других приложений или мешать ему.

GraphML проектировался с ориентацией на эту цель, а также с учётом следующих более прагматических требований:

- 1) Простота (Simplicity). Формат должен быть простым для разбора и интерпретации как людьми, так и машинами. В качестве общего принципа формулируется отсутствие неоднозначностей и таким образом существование единственной хорошо-определенной интерпретации для каждого валидного (valid) GraphML-документа.
- 2) Общность (Generality). Не должно существовать ограничений по отношению к графовой модели, т. е. гиперграфы, иерархические графы и т. д. должны быть выразимы с помощью одного и того же базисного формата.
- 3) Расширяемость (Extensibility). Должна существовать возможность расширять формат хорошо-определённым способом для представления дополнительных данных, требуемых произ-

вольными приложениями или более сложным использованием (например, посылая алгоритм раскладки вместе с графом).

- 4) Робастность (Robustness). Система, не способная обработать весь диапазон графовых моделей или дополнительной информации, должна быть в состоянии легко распознавать и извлекать то подмножество, которое она может обработать.

Разработанный язык описания графов GraphML базируется на XML [13]. Он позволяет описывать ориентированные, неориентированные и смешанные графы, гиперграфы и иерархические графы, а также любые специфичные для конкретных приложений атрибуты. В частности, язык GraphML полностью поддерживает атрибутированные иерархические графы. Благодаря XML синтаксису, GraphML может использоваться в комбинации с другими форматами, основанными на XML. С другой стороны, свой собственный механизм расширения позволяет прикреплять <data> метки со сложным содержимым (возможно требуемый для исполнения с другими моделями XML содержимого) элементов GraphML.

Возможности системы

Пользовательский интерфейс системы изображен на рис. 3. Он включает рабочий стол, миникарту, навигационную панель, а также атрибутную панель.

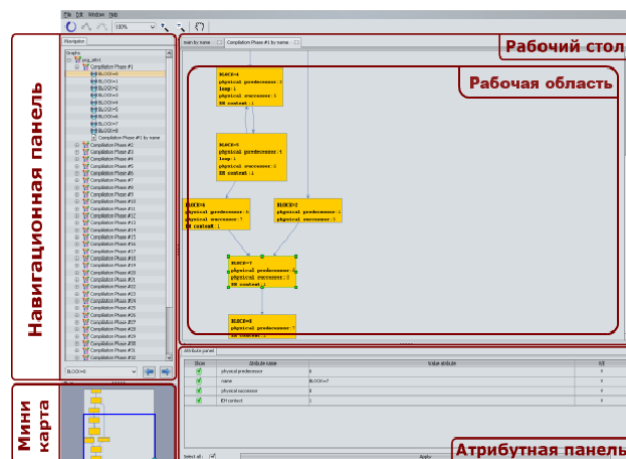


Рис. 3: Пользовательский интерфейс системы Visual Graph

Рабочий стол. Рабочий стол состоит из набора вкладок, открываемых пользователем для визуализации выбранной части графовой модели в виде ее изображения на плоскости. Для улучшения полученного автоматически изображения, пользователь может менять форму вершин и ребер, раскладку и ее параметры, отображаемые атрибуты, масштаб видимой области и многое другое.

Миникарта. Данный инструмент позволяет обозревать целиком весь граф, показанный в текущей вкладке, а также помогает перемещать и масштабировать видимую его область – ту часть графа, которая видна в текущей вкладке.

Навигационная панель. Навигационная панель предназначена для визуализации всех графов, с которыми работает пользователь, в виде изображения с помощью отступов деревьев вложенности фрагментов этих графов. Для быстрого поиска по деревьям реализована строка поиска, которая позволяет пользователю без труда найти интересующие его элементы, используя регулярные выражения. После чего пользователь может выделить интересующие его элементы и открыть их в новой вкладке.

Атрибутная панель. Атрибутная панель – инструмент, который позволяет управлять визуализацией атрибутов для выбранных вершин и ребер в текущей вкладке. Для этого пользователю необходимо выделить у графа из текущей вкладки те вершины и ребра, которые ему интересны, после чего отметить в атрибутной панели галочками те атрибуты, которые он хочет визуализировать у этих элементов.

Так же с помощью данного инструмента можно задать набор атрибутов, которые будут видны у элементов графа, открытых в новой вкладке.

Фильтр. Фильтр – инструмент, который поддерживает поиск в текущей вкладке элементов (вершин и ребер) графовой модели по их атрибутам. Условия поиска задаются пользователем с использованием имен атрибутов и их значений (Рис. 4). Заданные пользователем условия могут быть объединены в выражения с помощью логических операций и скобок. Результатом работы фильтра является набор всех тех элементов графовой модели, которые удовлетворяют заданным условиям (Рис. 5).

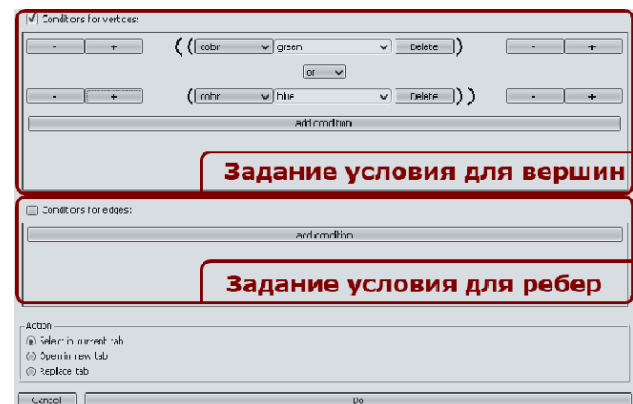


Рис. 4: Фильтр

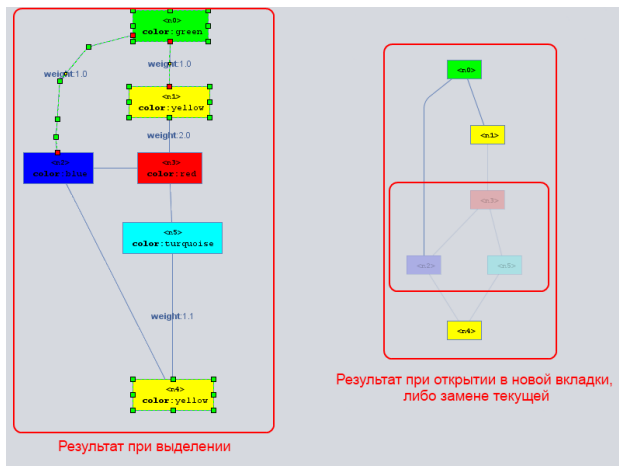


Рис. 5: Результат работы фильтра

Блокнот. Блокнот – инструмент, позволяющий загружать дополнительную информацию в виде текстовых файлов и связывать ее с графовой моделью. После этого пользователь может переходить от элементов графовой модели к связанной с ними дополнительной текстовой информации. Например, это можно использовать для связи синтаксического дерева с исходным кодом.

Средства структурного анализа. К этим средствам относятся различные алгоритмы работы с графовыми моделями, которые помогают пользователю выделять и визуализировать нужную ему информацию в их изображениях. К ним относятся, например, такие средства, как подсветка кратчайшего пути, циклов, обязательных предшественников и т. д., а также поиск максимального общего подграфа двух графов [14]. Результат работы такого поиска изображен на Рис. 6.

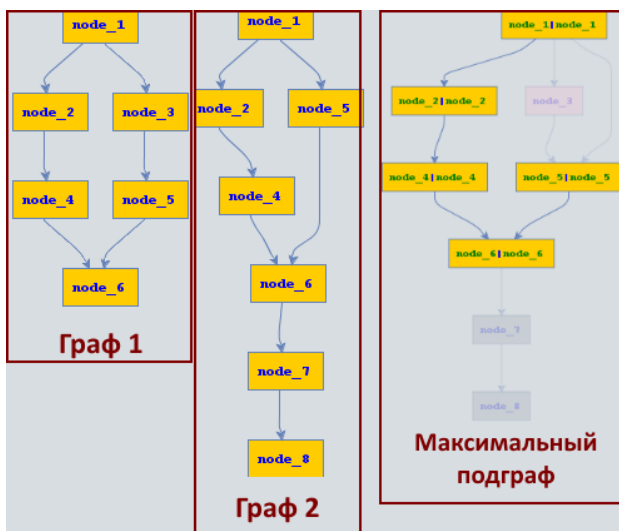


Рис. 6: Пример работы алгоритма поиска максимального общего подграфа двух графов

Заметим, что задача нахождения максимального общего подграфа двух графов является NP-трудной, но возникает перед разработчиком компилятора довольно часто. Например, когда он хочет найти разницу в поведении разных версий одного и того же компилятора. Эту разницу можно увидеть благодаря графам, возникающим внутри компилятора в момент компиляции. В частности, благодаря ним можно узнать какие оптимизации были сделаны или не сделаны в одной версии в отличие от оптимизаций, реализованные в другой версии.

При небольших графах (порядка 5 вершин), пользователь может визуально пытаться решить эту задачу, не прибегая к дополнительному инструментарию. Но стоит увеличить количество вершин до 15 – 20 и эта задача уже не будет столь тривиальной.

Раскладчики. Задача раскладки графа на плоскости не имеет и не может иметь оптимального решения в связи с тем, что наборы критериев для оценки качества раскладки различны для различных приложений [1 – 5]. Известно, что для тех или других приложений одни критерии могут вносить большую значимость в наглядность графа, чем другие. Тем не менее, для конкретных задач (и связанных с ними классов визуализируемых графов) обычно можно подобрать виды раскладок, их примерные параметры и алгоритмы.

Основные графы, с которыми работает компилятор, имеют относительно небольшое количество элементов при их представлении в виде иерархических графов, когда надо визуализировать в отдельных вкладках фрагменты, в которых вложенные фрагменты свернуты в вершины. Как правило, алгоритм поуровневой укладки (иерархический раскладчик) хорошо справляется с такими видами графов. В зависимости от размера графа в нем можно использовать различные эвристики для минимизации ширины или высоты получаемой раскладки, а также для уменьшения числа пересечений между ребрами. Система Visual Graph предоставляет пользователю возможность использования нескольких раскладчиков, основным из которых является иерархический раскладчик (Рис. 7). При включении в систему он был максимально адаптирован для работы с графами, используемыми в компиляторах. Но при желании этот (и любой другой) раскладчик системы может всегда быть изменен и/или настроен под другие типы задач.

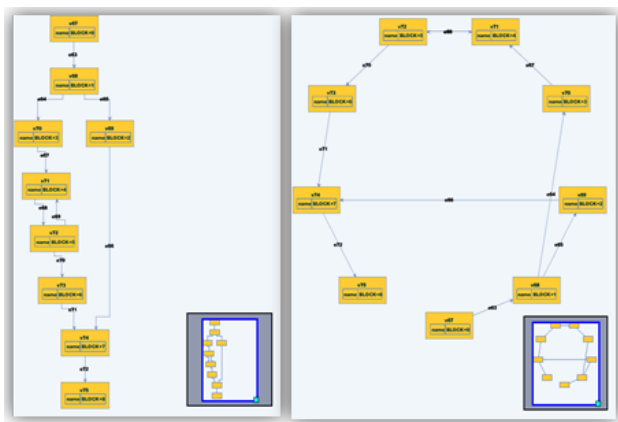


Рис. 7: Примеры раскладок управляющего графа

Особенности Реализации

Хранение графовых моделей внутри системы. Размер файла с входной графовой моделью может достигать сотни мегабайт, что не позволяет использовать оперативную память компьютера для хранения визуализируемых графов в системе. Поэтому было решено использовать кэширование данных на жесткий диск, используя для этого реляционную базу данных. В качестве реляционной базы данных была выбрана встраиваемая база данных SQLite [15]. В отличие от большинства популярных реляционных баз данных, для SQLite не требуется установка сервера, а вся клиент-серверная архитектура сводится к работе с файлами.

В системе Visual Graph часть данных расположена в базе данных, а часть в оперативной памяти, благодаря чему достигается высокая скорость при работе с графовыми моделями большого размера.

Расширяемость системы. Все возможности системы Visual Graph, в том числе по навигации, визуализации и структурному анализу, реализованы (и предоставляются пользователям системы) с помощью набора средств, который может расширяться как самими разработчиками системы Visual Graph, так и сторонними разработчиками.

Для достижения простой расширяемой системы было принято решение об использовании продукта Apache Felix [16], который в свою очередь является реализацией спецификации OSGi [17]. Данное решение является де-факто стандартным для подобного типа задач и позволяет разработчикам легко расширять систему за счет написания новых плагинов.

Основные аналоги системы

В настоящее время на рынке представлен достаточно широкий круг систем для визуализации графовых моделей. Наиболее известными из них являются: aiSee[8], yEd[9], Cytoscape[10] и Hlgres[6, 7].

Область применения, этих систем, частично совпадает с областью применения системы Visual Graph, но в целом несколько шире. Так многие из них рассчитаны на редактирование существующих графов и создание новых. В то время как система Visual Graph предназначена, исключительно, для просмотра уже существующих графовых моделей без возможности их редактирования.

Основными задачами, решаемыми системой Visual Graph, являются отображение графовых моделей и инструменты навигации по ним.

Поэтому в рамках этих двух задач и будем в основном рассматривать особенности каждой из перечисленных систем.

Система aiSee. Это – коммерческая система, которая автоматически строит раскладку исходного графа, описанного на специально разработанном языке GDL. Затем пользователь может интерактивно исследовать данный граф (без возможности редактирования), отпечатать его и сохранить в различных форматах. Первоначально система была разработана для визуализации структур данных, обрабатываемых компиляторами. На сегодняшний день ее используют десятки тысяч людей по всему миру в самых разных прикладных областях, в том числе:

- бизнес-менеджмент (структурные схемы предприятий, визуализация бизнес-процессов),
- генеалогия (генеалогические деревья, эволюционные диаграммы),
- разработка программного обеспечения (блок-схемы, управляющие графы, графы вызовов функций),
- веб-дизайн и оптимизация сайтов (карты сайтов, графы движения пользователей, визуализация графов на страницах вики),
- спорт (турнирные деревья) и др.

В aiSee присутствует несколько раскладчиков, и множество настроек для них, которые в разной степени влияют на получаемый результат. Качество раскладок определенных типов графов системой весьма высокое.

В качестве инструментов навигации, aiSee предоставляет:

1. Рабочий стол – инструмент, который визуализирует графовую модель целиком, выдавая пользователю статичную картинку, которую нельзя изменять, например, передвигая элементы.
2. Поисквик – инструмент, позволяющий пользователю искать элементы графовой модели с помощью их имен. Он поддерживает задание регулярных выражений, выбор категорий элементов для поиска и сохранение предыдущих поисковых запросов.

Система yEd. Данная система предназначена для быстрого создания и редактирования высоко-

качественных диаграмм. Создание диаграмм происходит вручную или с помощью импортирования из внешних данных. После чего к полученной диаграмме можно применить богатый набор алгоритмов для проведения анализа с последующим получением необходимой информации.

В системе *yEd* присутствует большое количество раскладчиков и опций для них, которые предоставляют возможность тонкой настройки визуализации каждого графа пользователем.

В качестве инструментов навигации система *yEd* предоставляет следующие:

1. Рабочий стол, схожий с рабочим столом, который предоставляет система *aiSee*. Но в отличие от *aiSee* здесь есть возможность работать с несколькими графами одновременно в одном экземпляре программы (создается вкладка для каждого графа), а также есть возможность редактирования элементов графа, начиная от смены их положения и размеров до задания атрибутов, влияющих на их визуализацию.
2. Навигатор – инструмент, отображающий граф, находящийся в текущей вкладке.
3. Миникарта – инструмент, показывающий весь граф, находящийся в текущей вкладке, и выделяющий ту его часть, которая видна на вкладке.

Система Cytoscape. *Cytoscape* – свободная система с открытым исходным кодом для визуализации и анализа сетей. Основная область применения данной системы является биоинформатика. Данная система стала очень популярной благодаря тому, что легко расширяется, позволяя сторонним разработчикам писать для нее различные плагины. *Cytoscape* имеет как собственные алгоритмы раскладки, так и сторонние, среди которых можно отметить присутствие алгоритмов из *yFiles*. Стоит также отметить, что результаты, получаемые в системе *yEd* намного лучше, чем аналогичные результаты, получаемые в системе *Cytoscape*, с настройками раскладчиков по умолчанию. Выделим основные средства навигации, которые предоставляет система *Cytoscape* для решения данной задачи:

1. Рабочий стол, схожий с рабочим столом, который предоставляет система *yEd*. Отличие заключается в том, что система *Cytoscape* не умеет работать с иерархическими графами.
2. Миникарта, идентичная миникарте в системе *yEd*.
3. Атрибутная панель – инструмент, позволяющий отображать текущие атрибуты и задавать новые. Данный инструмент выглядит в виде таблицы, по горизонтали, которой расположены имена атрибутов, по вертикали список выделенных вершин, а в ячейках соответствующие значения того или иного атрибута для той или иной вершины.

4. Фильтр – инструмент, позволяющий осуществлять поиск вершин и ребер по заданным условиям на атрибутах.

Система Nigres. *Nigres* – первый отечественный универсальный визуализатор и редактор графовых моделей. Созданная в ИСИ СО РАН система *Nigres* ориентирована на многооконную работу с простыми иерархическими графами (Рис. 8). Каждому фрагменту иерархического графа соответствует некоторый прямоугольник плоскости, внутри которого располагаются все его вершины. Кроме того, для каждого фрагмента можно открыть отдельное окно, в котором видны только вершины данного фрагмента и его подфрагменты. При этом каждый подфрагмент можно объявить закрытым – тогда изображаются только его контуры, либо открытым – тогда изображаются все его вершины и инцидентные им дуги. Для изображения контуров фрагментов в системе используется прием создания эффекта тени. Закрытые фрагменты выглядят слегка выступающими вверх – как будто они закрыты крышками, открытые же слегка утоплены вниз.

Важным отличием системы *Nigres* от других универсальных систем визуализации является ее способность сохранять во внутреннем представлении и визуализировать не только сам граф, но и его семантику, представленную в виде системы типов атрибутированных вершин, дуг и фрагментов графа, а также библиотеки алгоритмов обработки – так называемых внешних модулей. Причем пользователь системы может легко управлять методами визуализации графовой модели, а также корректировать и доопределять ее семантику. Такой подход обеспечивает, с одной стороны, универсальность системы *Nigres*, с другой – возможность ее специализации. Он также позволяет использовать систему как платформу для исполнения и анимации алгоритмов работы с иерархическими графами. Запустив внешний модуль, пользователь может регулировать параметры обработки графовой модели, прерывать алгоритм на любом шаге, просматривать в любую сторону последовательность изображений промежуточных результатов шагов работы алгоритма как в форме анимации, так и в покадровом режиме.

В качестве инструментов навигации *Nigres* предоставляет:

1. Рабочий стол, который хотя и аналогичен рабочему столу системы *aiSee*, но в отличие от него дает возможность редактировать элементы визуализируемого иерархического графа, начиная от смены их положения, формы и размеров до задания типов и атрибутов элементов графа, влияющих на их визуализацию, а также закрывая или открывая те или другие фрагменты.

2. Миникарта, аналогичная миникарте в системе yEd.

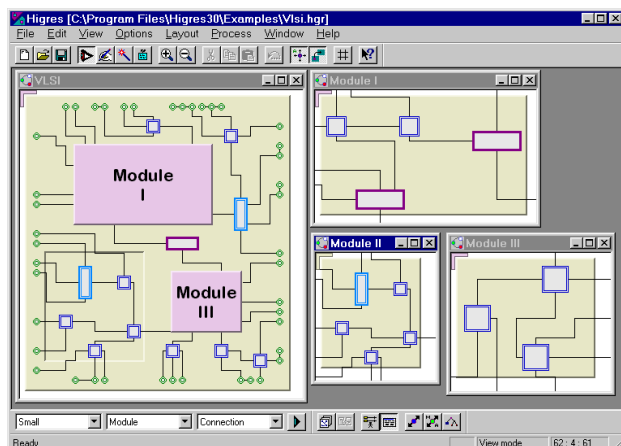


Рис. 8: Система HIGRES

Выводы

В докладе была рассмотрена система Visual Graph, которая создается в лаборатории конструирования и оптимизации программ ИСИ СО РАН для визуализации сложно структурированной информации большого объема на основе графовых моделей.

В отличие от зарубежных аналогов, таких как aiSee система Visual Graph поддерживает обработку произвольных атрибутированных иерархических графов (в том числе составных и кластерных графов) и использование для спецификации входного (визуализируемого) графа стандартного языка описания графов GraphML, ориентирована на пошаговое построение многооконного изображения графовой модели, состоящего из укрупненных изображений интересных пользователю фрагментов модели, и предоставляет богатые возможности для навигации по графовой модели и ее структурного анализа, работы с атрибутами ее элементов, а также простого расширения и настройки системы на нужды конкретного пользователя.

Созданная экспериментальная версия системы Visual Graph ориентирована на визуализацию структур данных, возникающих в компиляторах, позволяет одновременно работать с ними как в графовой, так и в текстовой форме, и обеспечивает плавность выполнения основных операций над графами, содержащими до 100000 элементов (вершин и дуг). Система проходит тестовое использование в компании Intel.

Ведется работа по встраиванию системы Visual Graph в визуальную среду параллельного программирования, создаваемую в рамках проекта по поддержке облачных супервычислений на базе языка Cloud Sisal [17, 18]. Цель проекта – дать возможность широкому кругу лиц, находящихся в уда-

ленных населенных пунктах или в местах с недостаточными вычислительными средствами, разрабатывать параллельные программы на доступных персональных компьютерах, имеющих выход в Интернет, и оперативно дистанционно использовать для их исполнения вычислительные мощности, сосредоточенные в крупных вычислительных центрах.

Создаваемая среда должна любому пользователю, имеющему выход в Интернет, позволить без установки дополнительного программного обеспечения на своем рабочем месте в визуальном стиле создавать и отлаживать переносимые параллельные программы на языке Cloud Sisal, а также в облаке осуществлять эффективное решение своих задач, исполняя на некотором супервычислителе, доступном ему по сети, созданные и отлаженные переносимые Cloud-Sisal-программы, предварительно адаптировав их под используемый супервычислитель с помощью облачного оптимизирующего кросс-компилятора, предоставляемого средой.

Среда использует внутреннее представление Cloud-Sisal-программ, которое имеет вид атрибутированных иерархических графов.

Предоставляемые системой возможности по визуализации внутреннего представления программ будут позволять пользователю получать изображение структуры программы и отслеживать процессы конструирования, отладки и преобразования Cloud-Sisal-программы с помощью графических образов.

Литература

- [1] Di Battista G., Eades P., Tamassia R., Tollis I. G. Graph Drawing: Algorithms for Visualization of Graphs. – PrenticeHall, 1999. – 397 p.
- [2] Herman I., Melancon G., Marshall M. S. Graph visualization and navigation in information visualization: a survey // IEEE Trans. on Visualization and Computer Graphics. – 2000. – Vol. 6. – pp. 24-43.
- [3] Касьянов В. Н., Евстигнеев В. А. Графы в программировании: обработка, визуализация и применение. – СПб.: БХВ-Петербург, 2003. – 1104 с.
- [4] Касьянов В. Н., Касьянова Е. В. Визуализация графов и графовых моделей. – Новосибирск: Сибирское Научное Издательство, 2010. – 123 с.
- [5] Касьянов В. Н., Касьянова Е. В. Визуализация информации на основе графовых моделей // Научная визуализация – 2014. – Т. 6, N 1. – С. 31 - 50.
- [6] Lisitsyn I. A., Kasyanov V. N. HIGRES – visualization system for clustered graphs and graph algorithms // Lecture Notes in Computer Science. – 1999. – Vol.1731. – pp. 82-89.
- [7] <http://pco.iis.nsk.su/higres> – Система HIGRES.
- [8] <http://www.aisee.com> – Система aiSee.
- [9] <http://www.yworks.com> – Система yEd.

- [10] <http://www.cytoscape.org> – Система Cytoscape.
- [11] *Kasyanov V. N.* Hierarchical graphs and visual processing // ICM 1998 International Congress of Mathematicians. Abstracts of Short Communications and Poster Sessions. – Berlin, 1998. – p. 292.
- [12] *Касьянов В. Н.* Иерархические графы и графовые модели: вопросы визуальной обработки // Проблемы систем информатики и программирования. – Новосибирск, ИСИ СО РАН, 1999. – С. 7-32.
- [13] <http://graphml.graphdrawing.org/specification.html> – Спецификация языка GraphML.
- [14] *Золотухин Т.* Алгоритм поиска максимального подграфа двух графов и его реализация в рамках системы VisualGraph // Инновационные технологии: теория, инструменты, практика (InnoTech 2013). Материалы V Международной Интернет-конференции молодых ученых, аспирантов и студентов. – Пермь: ПНИПУ, 2014. – С. 190-196.
- [15] <http://www.sqlite.org> – SQLite.
- [16] <http://felix.apache.org> – Apache Felix.
- [17] <http://www.osgi.org/Main/HomePage> – OSGi Alliance.
- [18] *Касьянов В.Н., Идрисов Р.И., Касьянова Е.В., Стасенко А.П.* Методы и средства параллельного программирования на основе языка Sisal // Материалы XV Международной конференции “Информатика: проблемы, методология, технология”. – Воронеж: ВГУ, 2015. – Том 3. – С. 166-170.
- [19] *Kasyanov V.N., Kasyanova E.V.* Cloud system of functional and parallel programming for computer science education // Proceedings of 2015 2nd International Conference on Creative Education (ICCE-2015), June 27-28, 2015, London, UK. – SMSSI, 2015. – pp. 270-275. – (Advances in Education Sciences, Vol. 10).

Компенсация линейного смаза цифровых изображений с помощью метода Люси-Ричардсона

К.В. Панфилова

panfilova_krista@mail.ru

Национальный исследовательский университет «МИЭТ», Москва, Зеленоград, Россия

В данной работе рассматривается задача восстановления изображений, а именно компенсация линейного смаза. Для восстановления используется итерационный алгоритм Люси-Ричардсона и его модификации. Узкими местами метода являются возникновение краевых артефактов в виде горизонтальных и вертикальных полос на изображении и отсутствие информации об оптимальном числе итераций. Предлагается расширять изображения за его пределы, монотонно сводя яркость к нулевому значению на новых границах, а также использовать определенный критерий останова. Последний основан на анализе изменения изображения на каждой итерации. В результате предложенных адаптаций удалось уменьшить уровень искажения изображения более чем на 50% в среднеквадратичной метрике, а также получить оценку числа итераций для корректной работы алгоритма.

Ключевые слова: восстановление изображений, метод Люси-Ричардсона, компенсация линейного смаза

Compensation of the linear blur of digital images using Lucy-Richardson method

K. V. Panfilova

National Research University of Electronic Technology, Moscow, Zelenograd, Russia

This paper considers the problem of image restoration, namely the compensation of linear blurring. To restore the image we use Lucy-Richardson iterative algorithm and its modifications. The disadvantage of the method is the emergence of edge artifacts in the form of horizontal and vertical stripes on the image and the lack of information about the optimal number of iterations. We propose to extend the image beyond its limits, monotonically reducing the brightness to zero at new borders, as well as use certain stopping criterion. The latter is based on analysis of the changes in the image at each iteration. As a result of the proposed adaptations we managed to reduce the level of image distortion by more than 50% of rms metric, as well as estimate the number of iterations for correct work of the algorithm.

Keywords: restoration image, Lucy-Richardson method, compensation of linear blur

Процесс формирования изображений несовершенен. Существуют различные типы искажений, которые зависят от многих факторов. В данной работе рассматриваются такие искажения как линейный смаз, возникающий при движении объекта относительно регистрирующей системы, и воздействие некоррелированного шума. Предлагается способ компенсации смаза (восстановления изображения), основанный на методе Люси-Ричардсона.

Формулировка задачи и обзор методов решения

Введём следующие обозначения: $f(x, y)$ – оригинальное (неискажённое) изображение, где значение функции определяет яркость изображения в пикселе с координатами (x, y) , $0 \leq x \leq M$, $0 \leq y \leq N$, а M , N – высота и ширина изображения; $g(x, y)$ – искажённое изображение, $h(x, y)$ – импульсная характеристика (ИХ) оператора искажения, $\hat{f}(x, y)$ – восстановленное изображение и $\eta(x, y)$ – шум. Обозначим $G(u, v)$, $H(u, v)$, $F(u, v)$, $N(u, v)$ – Фурье-образы функций $g(x, y)$, $h(x, y)$, $f(x, y)$ и $\eta(x, y)$ соответственно, полученные с помощью дискретного преобразования Фурье [2, стр. 284].

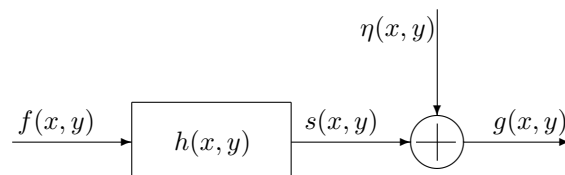


Рис. 1: Линейная система искажения изображения

Процесс искажения изображения (рис. 1) описывается с помощью свёртки [2, стр. 406] в пространственной области (1) и с помощью умножения в частотной (2):

$$g(x, y) = h(x, y) ** f(x, y) + \eta(x, y) \quad (1)$$

$$G(u, v) = H(u, v) \cdot F(u, v) + N(u, v) \quad (2)$$

где $< ** >$ – операция двумерной свёртки. Это представление описывает двумерную дискретную систему с импульсной характеристикой $h(x, y)$ [2, стр. 405].

Для получения оценки $\hat{f}(x, y)$ решается задача деконволюции. Методы её решения делятся на итерационные и неитерационные. Были рассмотрены некоторые из них. К первым относятся: инверсная фильтрация (3) [2, стр. 411], винеровская фильтрация (4) [2, стр. 414] и регуляризация по Тихонову (5) [2, стр. 418].

$$\hat{F}(u, v) = \frac{G(u, v)}{H(u, v)}, \quad (3)$$

$$\hat{F}(u, v) = \left(\frac{H^*(u, v)}{|H(u, v)|^2 + S_\eta(u, v)/S_f(u, v)} \right) G(u, v), \quad (4)$$

$$\hat{F}(u, v) = \left(\frac{H(u, v)}{|H(u, v)|^2 + \gamma|P(u, v)|^2} \right) G(u, v), \quad (5)$$

где $S_f(u, v) = \|F(u, v)\|^2$, $S_\eta = M[\|N(u, v)\|^2]$ – энергетические спектры изображения и шума соответственно, γ – параметр регуляризации, $P(u, v)$ – двумерное ДПФ дискретного аналога оператора Лапласа.

Наиболее распространенными итерационными являются метод Люси-Ричардсона (6) [5, 7] и методы слепой деконволюции [4, стр. 337].

$$\hat{f}_{k+1}(x, y) = \hat{f}_k(x, y) \left[h(-x, -y) * * \frac{g(x, y)}{h(x, y) * * \hat{f}_k(x, y)} \right], \quad (6)$$

Для корректной работы перечисленных методов необходимо получить оценки для оператора смаза и параметров шума.

Оценка параметров искажения

Для оценки параметров смаза (величины r и угла φ) предлагается использовать кепстральное представление изображения [1]:

$$\hat{g}(x, y) = F^{-1}\{\log |G(u, v)|\},$$

где обозначение $F^{-1}\{\cdot\}$ использовано для обратного преобразования Фурье. Кепстр ИХ оператора искажения имеет большой отрицательный пик, расположенный на расстоянии величины смаза от начала координат [1]. При свёртке двух изображений их кепстры складываются. Таким образом, кепстр искажённого изображения также обладает отрицательным пиком, сообщающим о наличии искажения и его расположении. Найдя его координаты, можно рассчитать величину смаза r и угол смаза φ . Этот способ определения параметров смаза позволяет найти параметры искажения достаточно точно: ошибка определения составляет не больше одной-двух единиц (пикселей для r и градусов для φ).

Шум, искажающий изображение, считаем аддитивным белым и не коррелированным с изображением. Известно, что основная информация изображения сосредоточена в области нижних частот. В области высоких частот значения спектра изображения

близки к 0 и вкладом первого слагаемого в (2) можно пренебречь. Таким образом, в некоторой высокочастотной области спектр изображения близок к спектру дискретного белого шума $S_\eta(u, v) \approx \sigma_\eta^2$. Для оценки среднеквадратичного отклонения (ско) σ_η используем формулу:

$$\sigma_\eta \approx \sqrt{\frac{1}{|\Omega|} \sum_{(i,j) \in \Omega} |G(u, v)|^2}, \quad (7)$$

где Ω – рассматриваемая высокочастотная область, $G(u, v)$ – спектр искажённого изображения.

Сравнение характеристик методов компенсации линейного смаза

Неитерационные методы. В инверсной фильтрации не учитывается влияние шума, хотя все реальные изображения являются зашумлёнными из-за несовершенства регистрирующих систем или условий съёмки. С помощью формулы (3) не удастся корректно решить задачу восстановления. Винеровская фильтрация позволяет учитывать шумовую составляющую, но для ее корректной работы необходима оценка энергетического спектра шума и изображения. Первую оценку можно получить с помощью формулы (7), а вторая нам недоступна. Регуляризация по Тихонову не требует знания о неискажённом изображении, но проблема использования этого метода состоит в подборе параметра регуляризации γ .

Основным недостатком неитерационных методов является то, что они чувствительны даже к небольшим погрешностям определения параметров искажения.

Итерационные методы. Метод Люси-Ричардсона был предложен независимо друг от друга Л. Б. Люси (1972) [5] и В. Х. Ричардсоном (1974) [7]. Он формулируется в терминах максимального правдоподобия, в котором изображение моделируется в виде статистик Пуассона. Алгоритм является итерационным и на каждом шаге улучшает оценку восстановленного изображения.

Методы слепой деконволюции [4, стр. 337], в отличие от алгоритма Люси-Ричардсона, не требуют явных знаний ИХ оператора искажения. Алгоритмы из этой группы также основаны на методе максимального правдоподобия, где целевой функцией является модель оригинального изображения.

Выводы. Для дальнейших исследований и компенсации линейного смаза на изображениях было решено использовать метод Люси-Ричардсона. Этот вывод сделан из следующих соображений:

- Алгоритм итерационный, и это позволяет получить более точную оценку изображения.

- Решение, полученное с помощью этого метода, всегда сходится к изображению, близкому к оригинальному. При использовании методов слепой дековолюции есть вероятность, что восстановленное изображение будет состоять только из нулевых пикселей.
- Вычислительные затраты у метода Люси-Ричардсона могут быть на порядок меньше, чем у методов слепой дековолюции. Несмотря на это, при корректном выборе входных параметров алгоритма, результаты восстановления часто идентичны.

Существующие варианты адапций метода Люси-Ричардсона

Важной характеристикой алгоритма являются требуемые для его работы вычислительные затраты. Поэтому нам хотелось бы получать хороший результат восстановления за меньшее число итераций. Существует ряд адаптаций алгоритма Люси-Ричардсона, направленных на увеличение его скорости сходимости к решению. В дальнейшем вместо (6) будем использовать следующий итерационный процесс Люси-Ричардсона [3]:

$$\hat{f}_k = u_k + \alpha_k h_k, \quad (8)$$

где

$$h_k = u_k - u_{k-1}, \quad u_{k+1} = \hat{f}_k + g_k, \\ g_k = \varphi(\hat{f}_k) - \hat{f}_k,$$

а $\varphi(\hat{f}_k)$ – результат $(k+1)$ -ой итерации алгоритма Люси-Ричардсона.

Таким образом, во время работы алгоритма Люси-Ричардсона выполняется экстраполяция: предсказывается значение на следующей итерации, основанное на предыдущих. Длина шага α_k определяется на каждой итерации следующим образом [3]:

$$\alpha_k = \frac{\sum g_{k-1} \cdot g_{k-2}}{\sum g_{k-2} \cdot g_{k-2}}, \quad 0 < \alpha_k < 1,$$

где операция « \cdot » – поэлементное умножение, \sum – суммирование по всем элементам массива.

Алгоритм Люси-Ричардсона, как и другие методы, основанные на методе максимального правдоподобия, влечет усиление шума в восстановленном изображении. Избежать дополнительных искажений восстановленного изображения можно несколькими способами: остановить итерационный процесс, когда уровень шума станет недопустимым; выполнить размытие полученного изображения; адаптировать метод, учитывая воздействие шума на процесс восстановления. Более надёжным является последний способ [4, стр. 328], который сводится к процедуре:

$$\hat{f}_{k+1} = \hat{f}_k \cdot C, \quad (9)$$

где C – ядро модифицированного алгоритма Люси-Ричардсона. Ядро C находится с помощью следующего выражения [4, стр. 331]:

$$C = g(x, y)^{\gamma-1} \cdot (\gamma - (\gamma - 1)g(x, y)),$$

где

$$g(x, y) = -\frac{2}{t^2} \left(G(x, y) \ln \frac{G(x, y)}{J(x, y)} - J(x, y) + G(x, y) \right),$$

а t – пороговое значение, определяющее уровень затухания (на практике удобно выбирать равным $\sigma/10$, где σ – ско шума), $J(x, y)$ – оценка искажённого изображения, $G(x, y)$ – искажённое изображение, поступающее на вход алгоритма. Из эмпирических соображений предлагается принимать $\gamma = 10$ [4, стр. 332].

Рассмотренные модификации позволяют получить более корректную оценку изображения по сравнению со стандартным алгоритмом Люси-Ричардсона (6). Далее для итерационной процедуры восстановления будем использовать (8) и (9).

Определение оптимальных параметров использования метода Люси-Ричардсона

Уменьшение влияния краевых артефактов.

Метод Люси-Ричардсона (так же, как и метод слепой дековолюции) обладает следующим недостатком: при величине смаза r больше 20 – 25 пикселей и большом числе проделанных итераций метода появляются артефакты в виде горизонтальных и вертикальных полос вблизи границ изображения. Данные искажения можно ослабить, если расширить изображение за его пределы, монотонно сводя яркость к нулевому значению на новых границах. Ширина расширяемой области выбирается в зависимости от величины смаза r и структуры изображения. Рекомендуется увеличивать изображение с каждой стороны на величину $2r$. Благодаря расширению уровень влияния артефактов уменьшается, но время восстановления увеличивается. В некоторых приложениях это может быть критично. Тогда стоит выбирать размер зоны расширения в зависимости от структуры изображения: если на краях изображения нет важных деталей, то ширина равна r , иначе – $2r$. Чтобы свести яркость монотонно к нулю, предлагается использовать следующую функцию:

$$W(x, y) = \begin{cases} 1, & 2S_X + 1 \leq x \leq M - 2S_X - 1 \\ & 2S_Y + 1 \leq y \leq N - 2S_Y - 1 \\ w(n), & S_X \leq x \leq 2S_X, S_Y \leq y \leq 2S_Y \\ 0, & \text{в остальных случаях} \end{cases}, \quad (10)$$

где M, N – размеры изображения, $S_X = \text{round}(m/4)$, $S_Y = \text{round}(n/4)$, m, n – размеры

ИХ оператора искажения, функция *round* – округление в большую сторону, а $w(n)$ – весовая функция Хэмминга:

$$w(n) = 0.54 - 0.46 \cos \frac{2\pi n}{N-1}.$$

В случае использования (10), результат восстановления получается заметно лучше, чем при использовании прямоугольного окна.

Таким образом, для предотвращения появления артефактов следует расширять изображение, выбрав параметры в соответствии с поставленной задачей и назначить каждому пикселю вес с помощью (10).

Выбор оптимального количества итераций для алгоритма. Если оригинальное изображение известно, то оптимальное количество итераций можно определить по норме разности между ним и восстановленным изображением:

$$\|I - J\| = \sqrt{\sum_{(i,j)} (I(i,j) - J(i,j))^2}.$$

Когда значение нормы будет минимально, необходимо завершить работу алгоритма. На практике оригинальное изображение не известно, будем использовать норму разности восстановленных изображений на текущей (k -ой) и предыдущих итерациях ($S_k = \|\hat{f}_k - \hat{f}_{k-1}\|^2$). Чтобы краевые артефакты оказывали меньшее влияние на работу критерия останова, будем рассматривать только 1/10 пикселей центральной части изображения. Эксперименты показали, что чем ближе количество проделанных итераций (9) к оптимальному числу итераций, тем более гладко изменяется значение нормы S_k . Поэтому, установив ту итерацию, которая принесёт меньшее изменение этой нормы, чем другие, после неё возможно закончить итерационный процесс восстановления. Для этого предлагается дополнение алгоритма Люси-Ричардсона (алгоритм 1) (далее в обозначениях I_{blur} – искажённое изображение размером $M \times N$).

Экспериментальные результаты и выводы

Рассмотрим результаты применения предложенных адаптаций метода: расширение изображения и критерий останова процесса восстановления на нескольких примерах.

Покажем, как работают предложенные адаптации на искусственно искажённом изображении. На вход алгоритма подается искажённое изображение (600×450) со смазом в 30 пикселей под углом 28° (рис. 2б).

Алгоритм 1 Дополнение алгоритма Люси-Ричардсона.

- 1: Найдём нормировочный коэффициент (ско) для области Ω , состоящей из центральных пикселей изображения:

$$avr = \frac{1}{MN} \sum_{(i,j)} I_{blur}(i,j)$$

$$J = I_{blur} - avr$$

$$n = \sum_{(i,j) \in \Omega} J(i,j).$$
- 2: Выполняем 5 итераций алгоритма Люси-Ричардсона (9). Полагаем $k = 6$.
- 3: Переходим к следующей (k -ой) итерации и находим оценку изображения \hat{f}_k в соответствии с (9).
- 4: Находим норму разности $S_k = \|\hat{f}_k - \hat{f}_{k-1}\|^2$.
- 5: Чтобы избежать резких изменений S_k и ложных срабатываний критерия, проведём сглаживание:

$$S_k := 1/4 \cdot S_{k-2} + 1/2 \cdot S_{k-1} + 1/4 \cdot S_k.$$
- 6: **если** $\max(S_{k-4}, S_{k-3}, S_{k-2}, S_{k-1}, S_k) < \sigma_\Omega \cdot 10^{-3}$
то
- 7: **если** величина смаза $r < 15$ **то**
- 8: **останавливаемся** на $k + 26$ итерации.
- 9: **иначе**
- 10: **останавливаемся** на $k + 1$ итерации.
- 11: **иначе**
- 12: **увеличиваем** k на 1 и переходим к шагу 3.



Рис. 2: а) Оригинальное изображение, б) Искажённое изображение (смаз – 30 пикселей под углом 28°)

Так как мы имеем в наличии неискажённое изображение (рис. 2), то можем оценить оптимальное количество итераций для восстановления изображения с помощью нормы разности между восстановленным и оригинальным изображением. Окажется, что нужно остановиться на 119-й итерации.

В соответствии с предложенным выше критерием останова, процесс восстановления прекращается на 77 итерации. Результат рассмотрим на рис. 3.

Уровень искажения DL (Distortion Level) определяется следующим образом:

$$DL = -10 \lg \frac{\sum_{(i,j) \in \Omega} I(i,j)^2}{\sum_{(i,j) \in \Omega} (I(i,j) - J(i,j))^2},$$

где $I(i,j)$ – значения яркости в пикселе (i,j) оригинального изображения, $J(i,j)$ – яркость в отсчё-



Рис. 3: Результат восстановления искажённого изображения 2 с помощью алгоритма Люси-Ричардсона: а) 119 итераций, б) 77 итераций

Таблица 1: Результаты работы алгоритма для искажённого изображения 2. Величина смаза 30 пикселей, угол – 28° . Время указано для компьютера с процессором Intel Core 2 Duo, 2,20GHz.

Уровень искажения, дБ	Искажённого изображения	-21,0
		После 119 итераций
	После 77 итераций	-27,5
Время выполнения, с	119 итераций	19,8
	77 итераций	11,5

тах искажённого изображения, Ω – рассматриваемая область, состоящая из 1/10 пикселей центральной части изображения.

Из таблицы 1 и рис. 3 видно, что результат восстановления после 77 итераций по качеству не уступает результату после 119 итераций. Разница в 0,2 дБ является несущественной. Кроме того, вычислительные затраты стали меньше. Значит, использование предложенного критерия для данного искажённого изображения корректно.

Рассмотрим пример с изображением, природа искажения которого нам неизвестна.



Рис. 4: Пример работы метода: а) Искажённое изображение (600×482), б) Восстановленное изображение

Предложенный алгоритм определил параметры искажения изображения на рисунке 4а: величина смаза равна 22 пикселям, а угол – 45° . Количество итераций, необходимых для восстановления, составляет 45 (время выполнения 11,2 с). На изображении 4б можно прочитать автомобильный номер и рассмотреть мелкие детали изображения, которые не видны на рисунке 4а, поэтому можно утверждать, что восстановление прошло успешно.

Для того чтобы получить описанный выше критерий останова алгоритма была проведена серия экспериментов на изображениях с различной структурой и различными смазами. В ходе опытов проводился анализ изменения изображений на каждой итерации. Были сделаны выводы о том, что с помощью нормы разности $S_k = \|\hat{f}_k - \hat{f}_{k-1}\|$ на k -ой итерации можно определить номер итерации, на которой алгоритм должен прекратить восстановление без потери в качестве изображения.

В дальнейшем планируются исследования в использовании алгоритма Люси-Ричардсона и его адаптаций для компенсации нелинейных смазов.

Литература

- [1] Бъемон Ж., Лагендейк Л., Мерсеро Р.М. Итерационные методы улучшения изображений // ТИИЭР, 1990. – № 5.
- [2] Гонсалес Р.С., Вудс Р.Е. Цифровая обработка изображений. – Москва: Техносфера, 2012. – 1104 с.
- [3] Biggs D.S.C., Andrews M. Acceleration of iterative image restoration algorithms // Applied Optics, 1997. – Vol. 36 (No. 8).
- [4] Jansson P.A. Deconvolution of images and spectra // New York: Academic Press, 1997. – 2nd ed.
- [5] Lucy L.B. An iterative technique for the rectification of observed distributions // The Astronomical journal, 1974. – vol. 79 (No. 6).
- [6] Meinel E.S. Optical of linear and non-linear recursive restoration algorithms // Journal of the Optical Society of America, June 1986. – vol. 3 (No. 6). – pp.787-799.
- [7] Richardson W.H. Bayesian-Based Iterative Method of Image Restoration // Journal of the optical society of America, 1972. – vol. 62 (No. 6).

Интерференция света, изотропные прозрачные объекты, трассировка лучей*

В.А. Дебелов

debelov@oapmg.sscs.ru

Новосибирск, Россия, Институт вычислительной математики и математической геофизики СО РАН;
Новосибирский государственный университет

В данной работе рассматривается задача визуализации явления интерференции при расчете фотореалистических изображений. Изложение материала ведется на основе компьютерного моделирования известных опытов: опыт Юнга (разделение фронта волны) и двулучевая интерференция (разделение амплитуды). Приводятся все формулы необходимые для проведения расчетов. Материал достаточно иллюстрирован и объясняется на основе рассчитанных изображений. Также предложено, как можно модифицировать обратную рекурсивную трассировку лучей для учета интерференции.

Ключевые слова: фотореалистический рендеринг, спектральный рендеринг, трассировка лучей, изотропные прозрачные объекты, линейно поляризованный свет, интерференция, опыт Юнга.

Light interference, isotropic transparent objects, ray tracing*

V.A. Debelov

Novosibirsk, Russia, Institute of Computational Mathematics and Mathematical Geophysics SB RAS;
Novosibirsk State University

In this paper, we consider the problem of visualization of interference phenomena in the calculation of photo-realistic images. The presentation material is based on computer modeling of the known experiments: Young's experiment (separation of the wave front) and two-beam interference (amplitude division). Material is illustrated enough and explained on the basis of the calculated images. We give all the necessary formulas for calculations. It is also suggested a possible modification of the inverse recursive ray tracing [?] to account for interference.

Keywords: Photorealistic spectral rendering, physically based rendering, ray tracing, linear polarized light, isotropic transparent objects, light interference, double-slit (Young's) experiment.

1. Введение

Существующие программы фотореалистического рендеринга позволяют получать изображения для сцен, в которых оптические свойства (коэффициенты отражения, преломления и т.п.) объектов и источников задаются при помощи спектров в диапазоне волн видимого света. При помощи так называемого спектрального рендеринга возможен учет явления оптической дисперсии, а, следовательно, можно запрограммировать иллюстрацию школьного опыта Ньютона с призмой по разложению белого света в спектр, более того, стало возможным получение физически корректных изображений бриллиантов, стразов и стеклянных предметов.

В литературе [1,4] описано много натуральных опытов со светом, показывающих его волновую и спектральную природу, например, опыт Юнга [1]. Одно из таких явлений – интерференция, которую мы пока не можем продемонстрировать, используя существующие коммерческие программы фотореалистического рендеринга.

Метод трассировки лучей является, пожалуй, основным в фотореалистическом рендеринге, поэтому в данной работе мы рассмотрим каким обра-

зом можно модернизировать фотореалистический спектральный трейсер для того, чтобы правильно моделировать явление интерференции света. Как правило, учет интерференции невозможен без рассмотрения поляризации света и, таким образом, нам придется модернизировать представление луча света, введя параметры состояния его поляризации.

В целом содержание работы посвящено компьютерной реализации пары опытов на основе применения лучевой трассировки и спектрального рендеринга. Практическое применение учета интерференции в реалистическом рендеринге, возможно, не столь интересно, кроме случаев визуализации сцен с тонкими пленками на объектах либо тонкими дисками, а также для компьютерной постановки ряда физических опытов.

2. Существующие работы

С теоретической точки зрения вопрос расчета интерференции физиками проработан в достаточной степени. Данная работа посвящена несколько другому ракурсу проблемы, а именно: интерференции в фотореалистическом рендеринге. Достаточно обширный обзор работ, посвященных *рендерингу лучом поляризованного света* и рендерингу *оптически анизотропных прозрачных* материалов дан в

Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347

работе [5]. В предлагаемой работе будут рассматриваться только *оптически изотропные прозрачные* объекты. Вторая особенность почему явлению интерференции не уделялось достаточного внимания в компьютерной графике: интерферируют только лучи когерентного света. А в обычных 3D сценах свет, приходящий в исследуемую точку сцены разными путями, как правило, некогерентный. Оказывалось достаточно только энергетических расчетов для отдельных лучей. Таким образом, в исследуемой точке сцены энергии лучей только складывались, и не нужно было учитывать сдвиги фаз световых волн, «приносимых» в исследуемую точку сцены разными лучами.

С позиции компьютерной графики мы отметим работу Sun с соавторами [6], в которой были получены изображения, демонстрирующие интерференцию в тонких пленках. Также отметим работу Козлова [3], в которой он рассчитывает интерференцию для изотропных и анизотропных прозрачных объектов в специфической 3D сцене, являющейся моделью поляризационного (петрографического) микроскопа. В этих работах рассчитаны замечательные интерференционные картины, тем не менее, в них приводится недостаточно информации, чтобы повторить разработки авторов. Имеется еще ряд работ, но без интересующих нас деталей реализации.

В данной работе мы решили на основе компьютерных экспериментов рассмотреть построение изображений простейших сцен с учетом интерференции.

3. Поляризация света, интерференция

Все рассуждения проводятся в предположении, что свет в сцене неполяризованный или линейно поляризованный. В случае частично поляризованного света его можно рассматривать как сумму его неполяризованной и поляризованной частей.

В свою очередь луч неполяризованного света часто представляется суммой двух лучей света, линейно поляризованных во взаимно перпендикулярных плоскостях.

Все формулы рассматриваем для монохроматического света, длина которого в вакууме λ .

Введем луч линейно поляризованного света $R = \{P_0, \mathbf{dir}, \mathbf{X}, \mathbf{Y}\}$ – это математический луч $\{P_0, \mathbf{dir}\}$ и связанная с ним правосторонняя система координат $\{\mathbf{X}, \mathbf{Y}, \mathbf{dir}\}$, здесь P_0 – начало луча, \mathbf{dir} – направление.

Считаем, что вектор \mathbf{E} осуществляет колебания вдоль оси \mathbf{X} . Для фотографии важна, в основном, электрическая компонента электромагнитной волны, и приходящая интенсивность пропорциональна квадрату амплитуды этой волны, поэтому мы будем рассматривать только эту компоненту.

Пусть в исследуемую точку сцены P приходят две волны, их электрические векторы \mathbf{E}_1 и \mathbf{E}_2 , соответственно. Эти волны можно записать в виде $\mathbf{E} = \mathbf{E}_0 e^{i\phi}$, где ϕ – фаза, а \mathbf{E}_0 – амплитуда [1,4]. Следуя [1], результирующее электрическое поле – вектор $\mathbf{E}_\Sigma = \mathbf{E}_1 + \mathbf{E}_2$ отсюда

$$\mathbf{E}_\sigma^2 = \mathbf{E}_1^2 + \mathbf{E}_2^2 + 2\mathbf{E}_1\mathbf{E}_2.$$

Суммарная интенсивность I выражается через интенсивности обеих волн как

$$I = I_1 + I_2 + J_{12},$$

где последнее слагаемое – это *интерференционный член*. После этого авторы известных монографий переходят к случаю, когда лучи когерентного света приходят в P по *одному (или близким) направлению*. Возможно даже с *совпадающими плоскостями поляризации*. По нашему разумению, это связано с практическим аспектом, что при конструировании оптических устройств важен именно такой случай. В этих условиях интенсивность в точке P считается по формуле из [1]

$$I = I_1 + I_2 + 2\sqrt{I_1 I_2} \cos \delta \quad (1)$$

δ – это разность фаз двух электрических полей в P . Для простоты считаем, что оба луча покинули источник света с нулевой фазой. Пусть луч (точнее путь) прошел от источника до точки P расстояние длиной L_1 в среде с показателем рефракции n_1, \dots , расстояние длиной L_k в среде с показателем рефракции n_k , то пройденный им оптический путь вычисляется по формуле: $s = \sum_{i=1}^k n_i L_i$, а фаза при этом равна $\phi = 2\pi s/\lambda$. Два рассматриваемых луча прошли оптические пути s_1 и s_2 . Тогда разность фаз только за счет разности оптических путей равна $\delta' = \frac{2\pi}{\lambda}(s_1 - s_2)$.

Для того чтобы программировать трейсер, этой информации недостаточно. Необходимо рассмотреть также изменения фазы при отражении и преломлении света на границе Γ двух прозрачных сред с индексами рефракции n_1 и n_2 .

В [1,6] говорится, что при отражении от среды с большим индексом рефракции фаза сдвигается на π , т.е. не взирая на поляризацию, при $n_1 < n_2$

$$\phi_{new} = \phi_{old} - \pi \quad (2)$$

Рассмотрим более тонкий подход. Для описания взаимодействия луча с Γ , как правило, применяются формулы Френеля, см. [1,4], для чего у падающего луча даже уже поляризованного света электрическое поле в общем случае надо разложить на две компоненты, поляризованные перпендикулярно \mathbf{E}_\perp и параллельно \mathbf{E}_\parallel плоскости падения.

Кроме того, необходимо знать углы падения α и Брюстера β , $\text{tg } \beta = n_2/n_1$. На границе Γ может

быть скорректирована фаза для отраженного и преломленного лучей [4]:

- При преломлении падающий и преломленный лучи софазны.
- E_{\perp} : при отражении фаза всегда меняется на $-\pi$.
- E_{\parallel} : при отражении фаза меняется на $-\pi$, если:

$$(n_1 < n_2 \wedge \alpha < \beta) \vee (n_1 > n_2 \wedge \alpha > \beta). \quad (3)$$

На этом остановимся заниматься дальнейшими уточнениями математической модели.

4. Опыт Юнга

Этот опыт достаточно известен [1]. Параллельный пучок света падает на экран с небольшим отверстием. Пройдя через отверстие, свет доходит до второго экрана, в котором проделаны две щели. Когерентные пучки, излучаемые каждой из щелей, интерферируют на третьем экране.

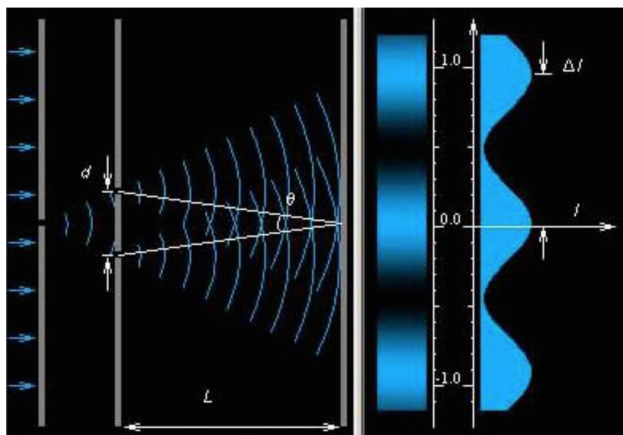


Рис. 1: Опыт Юнга, иллюстрация из [8].

На рис. 2 показана схема опыта. Опишем его как 3D сцену. Учитывая тот факт, что наш инструмент – это трассировка лучей, сцена будет очень проста: а) монохромные когерентные источники света S_1 и S_2 в виде тонких вертикальных параллельных линий, б) экран Э, на котором получается изображение. s_1 и s_2 – пути лучей до точки P . Отметим, что $l \gg d$. Полагаем индекс рефракции среды равным 1.

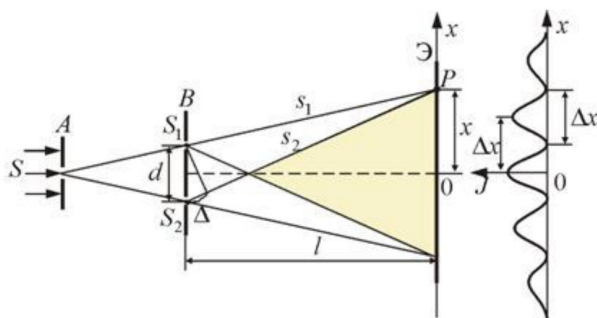


Рис. 2: Схема опыта Юнга, иллюстрация из [8].

Изображение на рис. 4 рассчитано на основе формулы (1) при следующих параметрах: $550 \cdot e^{-9}$; $d = 500 \lambda$; $l = 1000 d$

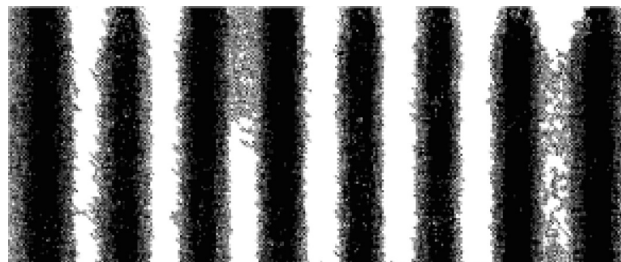


Рис. 3: Фото результата опыта Юнга [1].

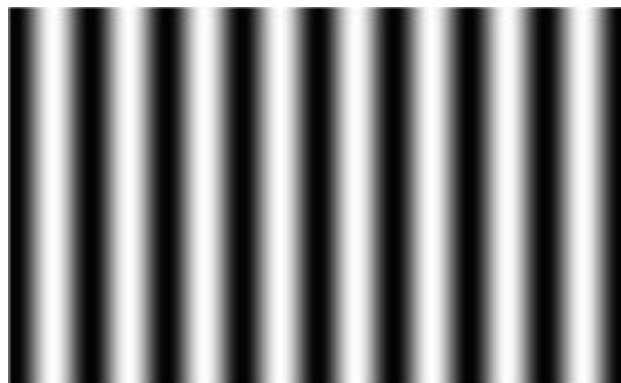


Рис. 4: Результат эксперимента.

5. Двухлучевая интерференция

Рассмотрим еще один опыт из [1]: прозрачная изотропная плоскопараллельная пластинка, освещенная точечным источником света. На рис. 5 представлена очередная сцена: точечный источник со спектральными характеристиками стандартного $D65$ находится на высоте h над пластинкой, толщина пластинки d , плоскость вертикального экрана Э находится на расстоянии a от источника. Индексы рефракции: среды – 1, пластинки – 1.5.

Проводим серию экспериментов. Рассматриваем источник неполяризованного света. На рис. 6 представлено изображение, которое получилось бы при спектральном рендеринге большинством существующих рендеров, т.к. в расчет берется только сумма интенсивностей, приходящих по двум лучам $I = I_1 + I_2$. Теперь добавим учет когерентности и разности фаз за счет разности оптических путей – рис. 7. На рис. 8 результат, полученный при учете сдвига фазы при отражении луча от более плотной среды. Видно, что картинка изменилась: цветные полосы как-бы сдвинулись. Теперь рассчитаем изображение, используя формулы Френеля, предварительно представив луч неполяризованного света в виде суммы лучей поляризованных перпендикулярно плоскости падения и параллельно ей $R =$

$= R_{\perp} + R_{\parallel}$. Т.е. делается учет всех формул: (1), (2) и (3). Согласно теории, интенсивность в точке P вычисляется простым суммированием двух интенсивностей $I = I_{\perp} + I_{\parallel}$, см. рис. 9.

Разность между изображениями рис. 8 и 9 имеет следующие показатели в формате RGB: total pixels: 810000; total different pixels: 79041; max difference ($r + g + b$): 60 at (12, 42); max component difference ($r|g|b$): 31 at (10, 49). Т.е. почти черное изображение. Получается, что более грубая модель рис. 8 дает вполне приемлемое изображение. На рис. 10 приведена усиленная разность, чтобы видеть места, где максимальные несовпадения. Такое несовпадение, скорее всего, связано с тем, что для рис. 8 все лучи обрабатывались по правилам для лучей с перпендикулярной поляризацией R_{\perp} , и влияние угла Брюстера на R_{\parallel} не учитывалось, см. формулу (3).

В обоснованиях опыта Юнга [1] основное предположение, что $l \gg d$ и у обоих лучей (верхний и нижний пути) направления колебаний \mathbf{E}_{\perp} (или \mathbf{E}_{\parallel}) в P совпадают (или «почти»), поэтому применяется формула (1). Такое же предположение мы применяли и в данном опыте, хотя по разным путям направления приходящих в точку P лучей различались.

Требуется рассчитывать интерференционный член с учетом этого факта, например, как в [1]. На рис. 11 приводится рассчитанное изображение. Не очень заметно, что оно немного бледнее, чем на рис. 9. Действительно, сравнение дает: total different pixels: 54578; max difference : 8 at (102, 137); max component difference: 4 at (98, 134). Значит расстояние до экрана уже достаточно большое. В расчете следующие параметры: $M = 550 \cdot e^{-9}$; $h = 10 M$; $d = 50 M$; $a = 100 M$; сторона квадратного экрана $300 M$.

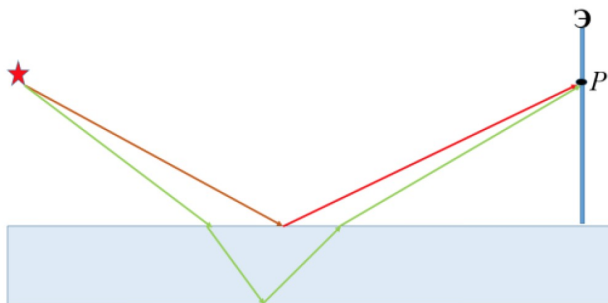


Рис. 5: Плоскопараллельная пластинка, освещенная точечным источником света.



Рис. 6: Только энергетический расчет, т.е. без учета когерентности.

6. Организация трассировки

Рассмотрим в качестве алгоритма рендера обратную рекурсивную трассировку Виттеда [7] в условиях спектрального рендеринга. Для каждой конкретной длины волны и каждого пикселя строится дерево трассировки (прямой проход), а при сборе интенсивности при обратном проходе дерева для полной физической корректности в смысле формул Френеля можно применить следующий прием, который без учета возможных оптимизаций выглядит так:

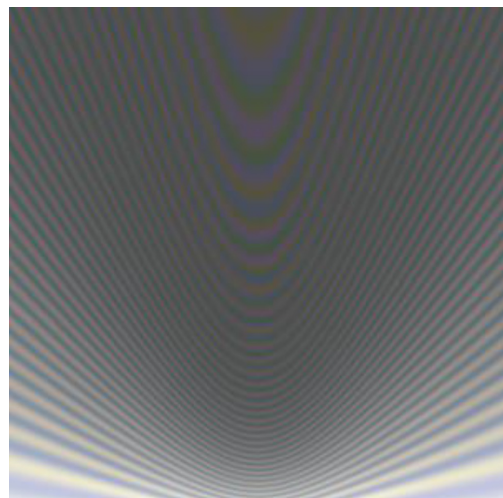


Рис. 7: Учтена когерентность и разность фаз из-за разности оптических путей, расчет по формуле (1).

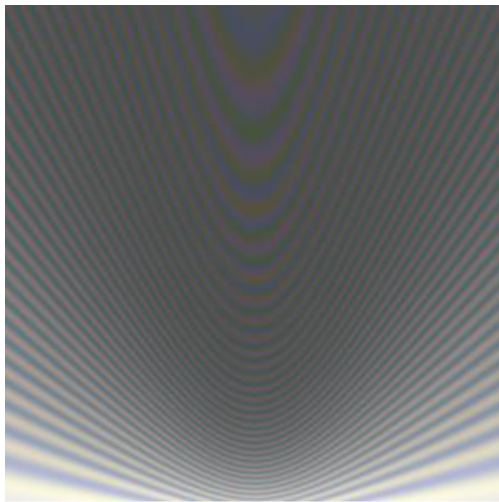


Рис. 8: Расчет по формуле (1) с учетом формулы (2).

1. При подъеме по дереву от листового узла (источник) к отцовскому узлу обратный луч раскладывается по правилу $R = R_{\perp} + R_{\parallel}$. Таким образом, на предыдущий уровень приходит уже два поляризованных луча. При этом подсчитывается фаза каждого порожденного луча.
2. При подъеме по дереву от узла k к отцовскому узлу $k - 1$ каждый обратный луч также раскладывается по правилу $R = R_{\perp} + R_{\parallel}$. Поскольку, в отличие от сцены нашего эксперимента, плоскость падения в узле может не быть параллельной плоскости падения в предыдущем узле. Таким образом, на уровень $k - 1$ приходит уже вдвое больше лучей. При этом подсчитывается фаза каждого порожденного луча.
3. В корне дерева осуществляется расчет результирующей интенсивности с учетом всех параметров пришедших лучей: интенсивности, состояния поляризации, фазы.

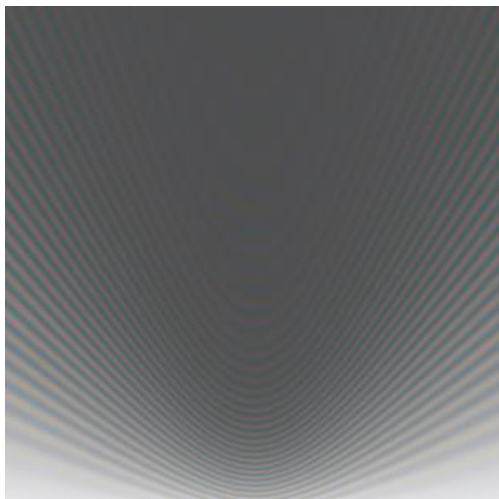


Рис. 9: Подход с представлением $R = R_{\perp} + R_{\parallel}$.

Очевидно, что в произвольной сцене нельзя надеяться на почти совпадение направлений интерферирующих лучей и необходимо делать более дорогой расчет.

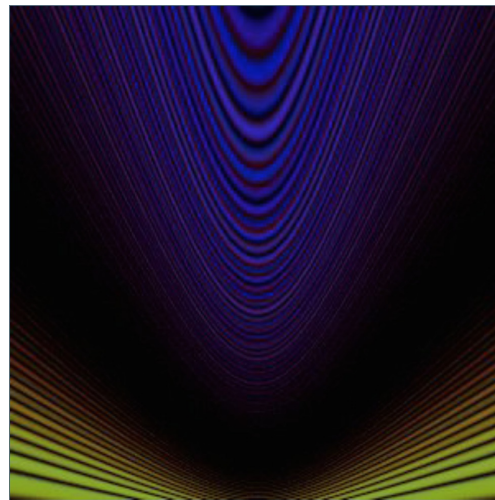


Рис. 10: Усиленная разность рис. 8 и рис. 9.

7. Заключение

Для большинства опытов (как показано на компьютерной симуляции опыта Юнга) и при моделировании оптических устройств подход без учета угла Брюстера вполне достаточен, что видно и по изображениям. Особенно, если учесть, что углы падения в основном около нуля или отражение вообще игнорируется. В наших экспериментах оказалось также, что перпендикулярная компонента дает более весомый вклад в изображение по сравнению с параллельной: на рис. 12 приведено изображение, полученное для источника, который испускает поляризованные лучи R_{\perp} с интенсивностью 0.5, а на рис. 13 – аналогичное для R_{\parallel} . На рис. 13 также видно, что около угла Брюстера почти нет отражения. Можно сказать (грубо), что

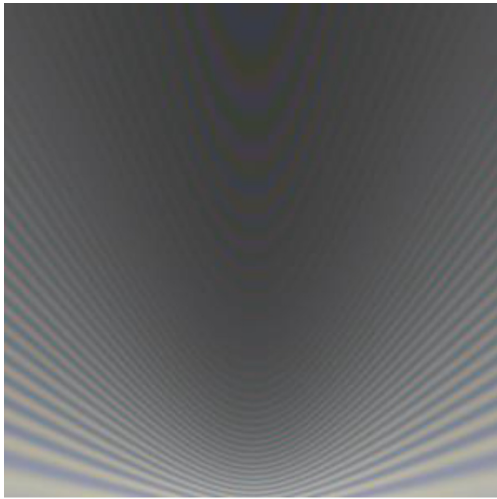
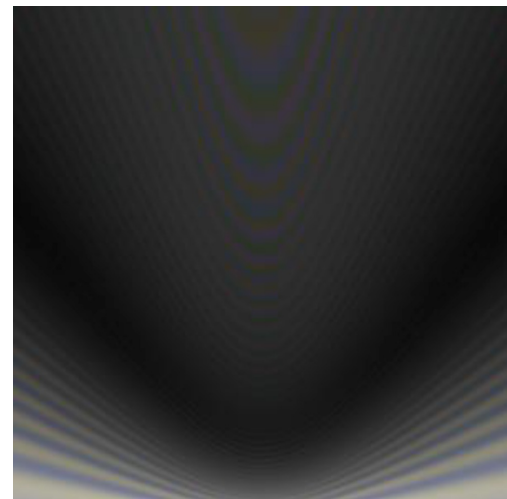
$$\text{рис. 9} = \text{рис. 12} + \text{рис. 13},$$

поскольку перпендикулярные поляризации не интерферируют.

Для полноты экспериментов необходимо также учитывать, что интенсивность луча от точечного источника убывает как квадрат расстояния. В работе это опущено, поскольку основное внимание уделяется тому, как учет поляризации или правильного подсчета сдвига фазы сказывается на изображении интерференции, и, в конечном счете, пользователь может выбрать подходящий ему вариант.



Рис. 11: Окончательное изображение.

Рис. 12: Освещение источником лучей R_{\perp} .Рис. 13: Освещение источником лучей R_{\parallel} .

- [4] Матвеев А.Н. Оптика. Москва, «Высшая школа», 1985.
- [5] Debelov V.A., Kozlov D.S. STAR: Rendering of transparent optically anisotropic objects // Proceed. of Graphicon-2013, Vladivostok, Russia, 16-20 Sept. 2013, pp.50–55.
- [6] Sun Y., Fracchia F. D., Calvert T.W., Drew M.S. Deriving Spectra from Colors and Rendering Light Interference // IEEE Computer Graphics and Applications. - 1999. - No. 4. - pp.1-8.
- [7] Whitted, T. An Improved Illumination Model for Shaded Display // Commun. ACM, Vol. 23, No. 6, 1980. P. 343-349.
- [8] Кузнецов С.И. Колебания и волны. Геометрическая и волновая оптика // Презентация <http://900igr.net/prezentatsii/fizika/Interferentsija-voln/019-Klassicheskij-interferentsionnyj-opyt-JUnga.html>

Литература

- [1] Борн М., Вольф Э. // Основы Оптики. Москва, «Наука», 1973.
- [2] Физически корректный алгоритм рендеринга анизотропных поглощающих кристаллов // Тр. 23-й Международной конференции по компьютерной графике и зрению ГрафиКон-2013, 16-20 сентября 2013, ИАиПУ ДВО РАН, ДВФУ, Владивосток, Россия, - С.191–194.
- [3] Козлов Д. Алгоритм рендеринга по методу Монте-Карло петрографических шлифов одноосных кристаллов // Тр. Школы молодых ученых. Графикон-2009. Москва.

Об авторе. Виктор Алексеевич Дебелов - доктор технических наук, ведущий научный сотрудник лаборатории численного анализа и машинной графики Института вычислительной математики и математической геофизики СО РАН, он также профессор на факультете информационных технологий Новосибирского государственного университета.

About the author. Victor A. Debelov holds a position of leading researcher at Computer Graphics Lab of the Institute of Computational Mathematics and Mathematical Geophysics SB RAS. Also, he is a part time professor of Novosibirsk State University, Dept. Information Technologies.

Автоматическое приближение набора точек поверхностями второго порядка в задачах анализа многомерных данных*

Д.А. Сумин, А.Е. Бондарев, А.Г. Волобой

da.sumin@gmail.com | bond@keldysh.ru | voloboy@gin.keldysh.ru

Институт прикладной математики им. М.В. Келдыша РАН Москва, Россия

В статье рассматривается задача выявления закономерностей между переменными многомерных объёмов данных. Используется подход, основанный на понижении размерности исходных многомерных данных, позволяющий сводить исходную задачу к анализу набора точек в трёхмерном пространстве. Предполагается автоматическая аппроксимация трёхмерного набора данных семейством поверхностей второго порядка; в случае нахождения таких поверхностей становится возможным сделать выводы о взаимосвязи соответствующих исходных переменных.

Рассмотрен метод обработки набора точек в трёхмерном пространстве, состоящий из разделения набора точек на поднаборы и последующего определения приближающих поверхностей второго порядка. Исследованы и приведены ограничения, которые рассматриваемый подход накладывает на входные данные.

Ключевые слова: обработка многомерных данных, метод главных компонент, сегментация облаков точек, методы подгонки.

Automatic fitting of 3D point clouds with quadric surfaces in multidimensional data analysis*

D.A. Sumin, A.E. Bondarev, A.G. Voloboy

Keldysh Institute of Applied Mathematics, RAS, Moscow, Russian Federation

In this article, we address a problem of finding relationships between variables of multidimensional datasets. A PCA-based approach is used for reducing problem's dimension to perform processing in a 3D space. An automated fitting of 3D-data to quadric surfaces is used to reveal hidden relationships between variables of original multidimensional data.

Proposed approach of multidimensional data processing bases on dimensional-reduction methods and 3D-data processing algorithms: segmentation of point clouds and fitting of point clouds to quadric surfaces. The latter methods are reviewed and examined.

Keywords: multidimensional-data processing, PCA, point-cloud segmentation, fitting methods.

Введение

Необходимость работы с многомерными данными возникает сегодня в большом количестве прикладных и теоретических областей науки. Актуальными являются задачи анализа и визуализации многомерных данных, задачи определения взаимного расположения точек в многомерных облаках данных. Интересными являются вопросы выявления определяющих факторов и скрытых взаимосвязей между ними.

Существует большое количество практических задач, в которых необходимо оптимизировать состояние системы, зависящей от большого (более двух) количества параметров. Задача в этом случае является многомерной, и к ней оказывается невозможным применить классические методы визуализации с целью определить влияние каждого из параметров и их комбинации на систему. Зачастую при этом проведение эксперимента с каждой новой комбинацией параметров может быть сопряжено с различными трудностями и затратами; пере-

бор всех необходимых вариантов может оказаться очень сложным или невозможным. В этой ситуации было бы логично применить математический аппарат для предположения результатов экспериментов без их фактического проведения.

В [1] рассмотрен подход к анализу многомерных данных, возникающих в задачах вычислительной газовой динамики. Целью анализа набора точек $A_i(x_1, \dots, x_n), i = 1, \dots, m$ являлось изучение зависимости $x_n = F(x_1, \dots, x_{n-1})$ и поиск её явного представления. Задачу в такой постановке можно рассмотреть достаточно широко и применить методы её решения к описанным выше прикладным задачам предсказания состояния многопараметрических систем и нахождения скрытых зависимостей между переменными многомерных данных, поэтому развитие методов решения таких задач представляется актуальным. В этой статье описано развитие предложенного в [1] подхода.

Описание базового подхода

В этом разделе будет кратко освещен подход, изложенный в [1]. Авторы утверждают, что визуальное представление исследуемой зависимости $x_n = F(x_1, \dots, x_{n-1})$ — наиболее эффективный способ

Работа выполнена при финансовой поддержке РФФИ, грант 14-01-00769 и опубликована при финансовой поддержке РФФИ, грант 15-07-20347.

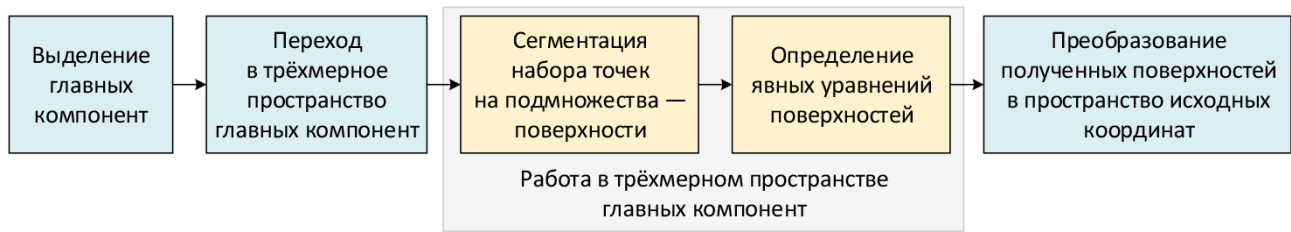


Рис. 1: Общая схема предлагаемого алгоритма.

анализа многомерных данных. Первым шагом обработки входных многомерных данных является понижение их размерности до трёх переменных, так как в настоящее время отсутствует [2] надежный и адекватный способ визуализации данных, имеющих размерность, превышающую 3.

Для понижения размерности входных данных $A_i(x_1, \dots, x_n), i = 1, \dots, m$ с целью дальнейшего поиска зависимости $x_n = F(x_1, \dots, x_{n-1})$ предлагается применить метод главных компонент (РСА), который является одним из основных методов уменьшить размерность данных, потеряв при этом минимальное количество информации. Реализации метода главных компонент и обобщения на нелинейные случаи подробно описаны в [2]; применение метода главных компонент в визуализации данных описано в [3].

После применения метода главных компонент будут получены новые переменные Y_1, Y_2, Y_3 , которые являются линейными комбинациями исходных переменных: $Y_j(x_1, \dots, x_n) = \sum B_j x_i, j = 1, 2, 3$. Это позволяет выразить точки исходного набора данных в трёхмерном пространстве главных компонент: $A_i(x_1, \dots, x_n) = A_i(Y_1(x_1, \dots, x_n), Y_2(x_1, \dots, x_n), Y_3(x_1, \dots, x_n))$.

После этого предполагается визуализация набора точек в трёхмерном пространстве и её изучение на предмет возможности аппроксимировать набор точек с помощью функций, имеющих аналитическое выражение. В [1] авторы применяют аппроксимацию с помощью параметрически заданных плоскостей вида $E_1 Y_1 + E_2 Y_2 + E_3 Y_3 = C_y$. Вследствие того, что плоскость при обратном переходе к исходным переменным сохраняет свои свойства, в результате преобразования получается следующая зависимость: $E'_1 Y_1 + E'_2 Y_2 + \dots + E'_n Y_n = C_x$. Это выражение можно рассматривать как искомую квазианалитическую зависимость $x_n = F(x_1, \dots, x_{n-1})$.

Предлагаемый алгоритм

Из описания базового подхода следуют первоочередные направления его развития.

Их два:

- более точное приближение набора точек в пространстве главных компонент;
- автоматизация процесса аппроксимации.

Увеличение точности приближения разумно достичь, используя вместо плоскостей поверхности второго порядка. Так, представляется разумным использование эллиптических параболоидов в качестве приближающих поверхностей для данных, полученных в [1] после перехода в пространство главных компонент.

Автоматизация процесса подгонки семейства поверхностей к исследуемому набору точек становится тем более актуальной при использовании поверхностей второго порядка в качестве приближающих поверхностей, так как визуально-ручной подбор подходящей квадратичной поверхности представляется затруднительным.

Таким образом можно сформулировать следующую задачу для развития исходного подхода к анализу многомерных данных: необходимо по набору точек в трёхмерном пространстве, принадлежащих одной или нескольким поверхностям, определить явные уравнения этих поверхностей. При этом полезным может быть полуавтоматический режим, когда пользователь выбирает вид приближающей поверхности; в этом случае следует произвести наигонку точек поверхностями выбранного типа.

Поставленную задачу разумно разбить на два этапа:

1. сегментация набора точек на множества точек, принадлежащих различным поверхностям;
2. определение уравнения для каждой из поверхностей.

Эти этапы будут подробно рассмотрены далее.

Последовательность шагов предлагаемого алгоритма представлена на рис. 1.

Сегментация набора точек.

Сегментация набора точек в трёхмерном пространстве является первым этапом обработки полученных после понижения размерности данных. Входными данными для алгоритма сегментации является набор точек в трёхмерном пространстве. О наборе точек можно сделать следующие предположения:

- точки образуют набор поверхностей, находящихся на некотором расстоянии друг от друга;

— поверхности не пересекаются.

Рассматриваемый в [1] класс задач предполагает, что поверхности оказываются похожими на эллиптический параболоид; кроме того поверхности имеют небольшую выпуклость. Пример входных данных показан на рис. 2.

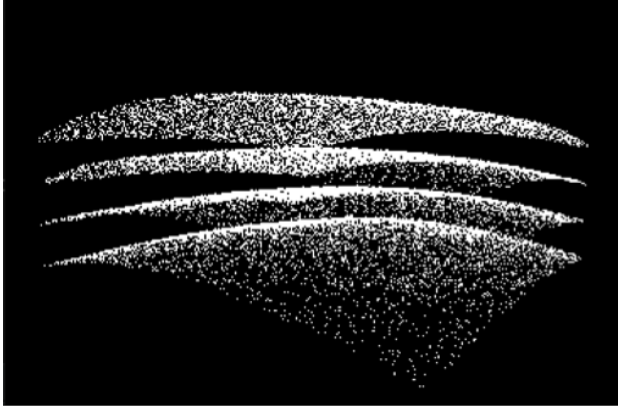


Рис. 2: Пример входных данных для алгоритма сегментации.

В случае соответствия входных данных описанным предположениям целью этапа сегментации является разделение набора точек на подмножества (кластеры), включающие в себя точки только одной поверхности. При этом оценка качества сегментации и возможности дальнейшего использования полученных результатов производится визуально. Предполагается, что несоответствие входных данных описанным предположениям устанавливается вручную; задача автоматического определения несоответствия в данной работе не рассматривалась.

Из описанных предположений о входных данных следует, что сегментацию произвести тем сложнее, чем ближе находятся поверхности друг к другу. Предельным является расстояние d_{min} между поверхностями: алгоритм выделяет правильно все поверхности, если расстояние между ними равно $d \geq d_{min}$ и не может определить поверхности, если расстояние между ними равно $d < d_{min}$. Тестирование алгоритма сегментации происходит на синтетических данных, состоящих из нескольких (например, 4) эллиптических параболоидов, заданных одной формулой и смещенных относительно друг друга на расстояние d по оси OZ (для задания параболоидов используется формула $z = \frac{x^2}{a^2} + \frac{y^2}{b^2}$).

Разумной метрикой для сегментации набора точек на множество поверхностей является гладкость локальной окрестности каждой из точек исходных данных. При построении алгоритма сегментирования необходимо найти компромисс между излишним и недостаточным сегментированием, выража-

ющихся в слишком большом и слишком маленьком количестве сегментов соответственно. Излишнее сегментирование характерно для алгоритмов, основанных на анализе кривизны поверхности и производных высокого порядка.

В данной работе для кластеризации набора точек был применён модифицированный алгоритм выделения областей путём их наращивания [5]. Этот алгоритм основан на анализе значений кривизны и векторов нормалей к точкам рассматриваемого набора данных.

Вместе с набором точек, значениями локальной кривизны в точках набора и векторами нормалей в этих точках алгоритм принимает на вход два пороговых значения: максимальная разница углов нормалей и максимальная разница значений локальной кривизны. Точки рассматриваемого набора обрабатываются в определённом порядке, связанном со значениями локальной кривизны в точках набора. После добавления новой точки в текущий кластер рассматриваются «соседи» этой точки; соседние точки, удовлетворяющие обоим пороговым значениям, также добавляются в кластер.

Пример результата работы алгоритма сегментации показан на рис. 3. Алгоритм успешно разделил поверхности на 4 набора. Незначительное количество точек (помечены красным цветом) не были отнесены ни к одному из кластеров.

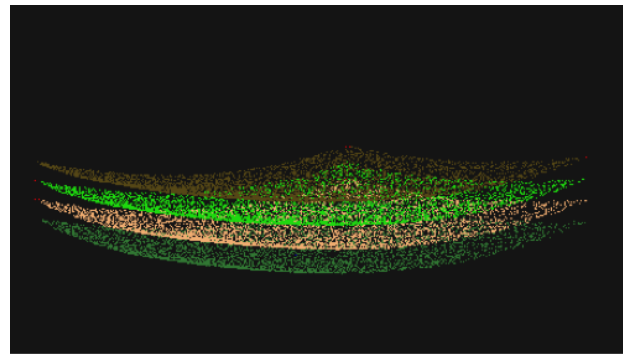


Рис. 3: Визуализация результата применения алгоритма сегментирования к набору точек, принадлежащих четырём параболоидам.

Апробация описанного алгоритма позволила сделать вывод об ограничениях, которым должны удовлетворять входные данные. Как было описано выше, интерес представляет минимальное расстояние d_{min} между подобными поверхностями, при котором алгоритм сегментирования корректно разделяет данные на кластеры. В результате экспериментов было установлено, что для данных, похожих на показанный на рис. 2 пример (несколько эллиптических параболоидов на некотором расстоянии друг от друга), расстояние между поверхностями связано с расстоянием между точками одной

поверхности. Было получено следующее соотношение: $d_{min} = 2.6r$, где r — расстояние между точками поверхностей, которые нужно отделить друг от друга. Алгоритм сегментации успешно выделяет поверхности при $d \geq d_{min}$.

Кроме нахождения расстояния d_{min} была проверена устойчивость процедуры сегментации к зашумленности входных данных. Было установлено, что максимальная амплитуда шума вдоль оси OZ, при котором возможно успешное разделение набора точек на поверхности, составляет $A_{max} = 0.35d$.

Нахождение уравнения поверхности второго порядка.

После того, как исходное множество точек было разбито на подмножества точек, принадлежащих разным поверхностям, необходимо найти явные выражения для каждой из поверхностей. Фактически, необходимо найти коэффициенты c_0, \dots, c_9 в общей записи уравнения поверхности второго порядка: $f(c, p) = c_0 + c_1p_x + c_2p_y + c_3p_z + c_4p_x^2 + c_5p_y^2 + c_6p_z^2 + c_7p_xp_y + c_8p_xp_z + c_9p_y p_z = 0$

Важно, что множество точек, составляющих каждую из поверхностей, могут не принадлежать одной поверхности второго порядка; при этом необходимым является нахождение поверхности, описывающей набор точек наилучшим образом. Найденная поверхность не должна обязательно содержать каждую из точек, но хорошо описывать набор точек в целом.

При построении и тестировании алгоритма определения уравнения поверхности второго порядка к исходным данным можно применить шум, сместив точки относительно исходной поверхности. Ответ при этом не должен значительно отличаться от случая, определения уравнения поверхности без шума.

Необходимо также обратить внимание на то, что рассматриваемые наборы точек составляют лишь сегмент поверхности. Как показано в [6], один и тот же сегмент поверхности может быть корректно отнесён к разным поверхностям второго порядка. Поэтому может быть полезной возможность явного выбора вида поверхности, которой будет приближаться рассматриваемый набор точек.

Алгебраические методы подгонки часто используются для подгонки поверхностей второго порядка [4]. Общая идея заключается в приближении нелинейной функции ошибки аппроксимации выражением вида: $\sum_{i=0}^n \frac{\|f(c, p_i)\|^2}{q(c)}$, где $q(c)$ — нормализующая функция. В данной работе была выбрана функция $q(c) = \sum_{i=0}^n \|\nabla_p f(c, p_i)\|^2$, предложенная в методе Таубина [7]. Метод Таубина является эффективным алгоритмом подгонки квадратичных поверхностей [4]; кроме того, в [6] предложен

способ определять с его помощью уравнения поверхностей второго порядка определённого вида, что может быть полезно в рассматриваемой задаче.

В ходе исследования была проверена устойчивость выбранного метода определения коэффициентов c_0, \dots, c_9 к шуму, который был добавлен к искусственно сгенерированным тестовым данным. Допустимая амплитуда шума зависит от площади сегмента, для которого происходит аппроксимация и приблизительно составляет $A = 0.03S$, где S — площадь рассматриваемого сегмента.

Заключение

В статье было рассмотрено развитие подхода к обработке многомерных данных, предложенного в [1]. Было предложено увеличить точность анализа входных многомерных данных путём использования квадратичных поверхностей при анализе главных компонент рассматриваемых многомерных данных и автоматизировать процесс обработки. Были рассмотрены и опробованы алгоритмы, позволяющие реализовать предложенные идеи.

В ближайшее время планируется провести эксперименты с целью проверить практическую применимость предлагаемого подхода и рассмотренных алгоритмов для решения многомерных задач вычислительной газовой динамики и, возможно, других практических задач.

Литература

- [1] Бондарев А.Е. Анализ многомерных данных в задачах вычислительной газовой динамики // Научная визуализация. – 2014. – Т.6, № 5, – С.59-66.
- [2] Gorban A., Kegl B., Wunsch D., Zinovyev A. Principal Manifolds for Data Visualisation and Dimension Reduction LNCSE 58, Springer, Berlin – Heidelberg – New York, 2007.
- [3] Зинovieв А.Ю. Визуализация многомерных данных – Красноярск: изд. КГТУ, 2000. – 180 с.
- [4] Chernov N., Ma H. Least squares fitting of quadratic curves and surfaces // Computer Vision, Editor S. R. Yoshida, Nova Science Publishers, 2011. – pp.285-302.
- [5] T. Rabbani, F. van den Heuvel, G. Vosselman Segmentation of point clouds using smoothness constraint // Proceedings of the ISPRS Commission V Symposium Image Engineering and Vision Metrology, 2006. – Vol.XXXVI Part 5. – pp.248-253.
- [6] Andrews J., Séquin C. Type-Constrained Direct Fitting of Quadric Surfaces // Computer-Aided Design and Applications, September 2013. – Vol.11, Issue 1 – pp.107-119.
- [7] Taubin, G. Estimation Of Planar Curves, Surfaces And Nonplanar Space Curves Defined By Implicit Equations, With Applications To Edge And Range Image Segmentation // IEEE Transactions on Pattern Analysis and Machine Intelligence. 13, 1991. – pp.1115-1138.

Технология и система визуализации больших объемов синоптических данных*

S.V. Мельман¹, В.А. Бобков², А.С. Черкашин²

melman.sv@dvmfu.ru | bobkov@iacp.dvo.ru | alexandr@cherkashin.ws

¹Дальневосточный федеральный университет, Инженерная школа, Владивосток, Россия;

²Институт автоматизации и процессов управления, Владивосток, Россия

Представлена технология создания системы визуализации больших объемов синоптических данных с использованием распределенных и гетерогенных (параллельных/гибридных) вычислений.

Ключевые слова: *научная визуализация, распределенные вычисления, клиент-сервер, синоптические данные.*

Technology and system visualization of large amounts of synoptic data*

S.V. Melman¹, V.A. Bobkov², A.S. Cherkashin²

¹Far Eastern Federal University, School of Engineering, Vladivostok, Russia;

²Institute of Automation and Control Processes, Vladivostok, Russia

The technology of creating a system of visualization of large amounts of synoptic data with distributed and heterogeneous (parallel/hybrid) computing.

Keywords: *scientific visualization, distributed computing, client-server, synoptic data.*

1. Введение

Исследования динамики атмосферных явлений и океана, выполняемые метеорологами, подразумевают работу с большими объемами пространственных данных. Поэтому при разработке программных систем, обеспечивающих автоматизацию научных исследований в этой области, возникают две основные задачи, от результативности решения которых зависит эффективность системы в целом: быстрая обработка больших объемов данных и визуализация 3D данных, облегчающая пространственное восприятие синоптических данных и анализ погодных явлений.

Ранее разработанная авторами система визуализации синоптических данных предназначалась для работы на обычном персональном компьютере. Система представляла собой набор программных приложений и отдельную программу визуализации. При этом обеспечивалась приемлемая скорость визуализации для результатов предобработки данных, сохраняемых на жесткий диск. Однако последующее развитие системы по нескольким направлениям: расширение функциональности визуального анализа 3D объектов с более развитым интерфейсом и использованием динамически подключаемых модулей нескольких типов, многоплановая и темпоральная анимация, возросшие объемы спутниковых данных – потребовало принципиального

повышения вычислительной производительности. Поэтому новая версия системы ориентирована на использование многопроцессорных систем с организацией параллельных вычислений.

С развитием технологий гетерогенного параллелизма появились работы с использованием гибридно-распределенных вычислений, как правило, учитывающих специфику конкретной тематики. Например, в работе [1] авторы предлагают использовать гибридно-распределенный подход с использованием MPI и многопроцессорных вычислительных архитектур для визуализации результатов математического моделирования векторных полей. Вопросы одновременного использования многоуровневого параллелизма и распределенных вычислений и связанные с этим проблемы рассматривались авторами [4, 5]. В работе [6] рассматриваются вопросы визуализации результатов моделирования физических полей на примере моделирования распространения акустического сигнала.

В настоящей работе предлагается технология и система визуализации больших объемов синоптических данных с реализацией многоуровневого гибридного (GPU+CPU) параллелизма и распределенных вычислений. Новая архитектура программного комплекса построена на использовании клиент-серверной технологии с централизованной БД и возможностью подключения в качестве вычислительных узлов, как персональных компьютеров, так и супер-ЭВМ. Используемый модульный подход обеспечил возможность расширения алгоритмической и функциональной базы за счет подключения дополнительных модулей по технологии plug-in.

Работа выполнена при финансовой поддержке Программы РАН по стратегическим направлениям развития науки «Фундаментальные проблемы математического моделирования» и Программы «Дальний Восток» (проект 15-I-4-025) и опубликована при финансовой поддержке РФФИ, грант 15-07-20347.

2. Технология параллельно-распределенной обработки

Предлагаемая технология параллельно-распределенной обработки больших объемов данных на гетерогенных вычислительных архитектурах применительно к обработке и визуализации спутниковых синоптических данных основывается на следующих принципах:

1. распределенные вычисления – для обеспечения вычислений в реальном времени;
2. клиент-серверная модель – для централизованного управления;
3. распределенная база данных – широко применяется принцип кэширования данных, каждый узел содержит только актуальные данные и только необходимые для работы модулей подключенных к этому узлу;
4. plug-in технология;
5. гибридный параллелизм (GPGPU+CPU) в подключаемых модулях.

Эти принципы реализованы в новой архитектуре системы визуализации. В архитектуре также учитывается специфика работы с синоптическими данными. Под синоптическими данными подразумеваются скалярные и векторные поля распределения погодных характеристик. Данные могут быть одномерными (показания зондов, стационарных радиолокационных станций), двумерными (погодные карты, спутниковые снимки облачности, льдов и т.д.) и пространственными (поля ветра, давления, температуры и т.д.). Кроме того, система визуализации должна работать со стандартными, используемыми синоптиками, форматами данных. При этом данные могут быть как статическими, так и изменяемыми во времени, т.е. динамическими.

Также предлагаемая архитектура дает пользователю совершенно прозрачную схему работы с данными, которая охватывает все стадии работы с данными от запроса данных со специализированных БД до визуализации. Каждая из описанных стадий подготовки данных выполняется с помощью подключаемого модуля (plug-in), что позволяет значительно расширять функционал системы в целом.

Процесс визуализации в системе выполняется как простая циклическая последовательность: получение данных, обработка, подготовка к визуализации, визуализация. Наибольшая вычислительная нагрузка ложится на этап обработки и подготовки к визуализации. Сюда включается моделирование, восполнение, фильтрация, разрежение данных. Эта работа возлагается на распределенную вычислительную систему. Непосредственно визуализация осуществляется полностью на ПК пользователя оснащенного современными мощными графическими ускорителями.

В основу предыдущей версии системы был заложен модульный принцип, согласно которому все функциональные блоки были реализованы в качестве отдельных программных динамических модулей. Необходимые динамические модули подключались автоматически. Такой подход позволил наращивать алгоритмическую базу без изменений основного приложения СВ. Указанные решения позволили в значительной степени использовать предыдущие наработки, включая модули загрузки, фильтрации и визуализации.

Клиент-серверная часть распределенных вычислений выполняет следующие функции:

1. Серверная часть – обеспечивает централизованное управление процессом сбора и обработки данных;
2. Клиентское вычислительное приложение – обеспечивает подключение отдельных вычислительных модулей (plug-in) на отдельном узле (ПК);
3. Клиентское интерфейсное приложение – предоставляет пользователю интерфейс управления всей системой и необходимый инструментарий для визуального анализа.

Реализованные протоколы обмена данных позволяют запускать клиентские вычислительные приложения на кластере. При этом остается необходимым портирование приложения и вычислительных модулей на операционную систему кластера.

2.1 Конвейер обработки данных

Процесс обработки и визуализации данных представлен на рис. 1 в виде разветвленного конвейера.

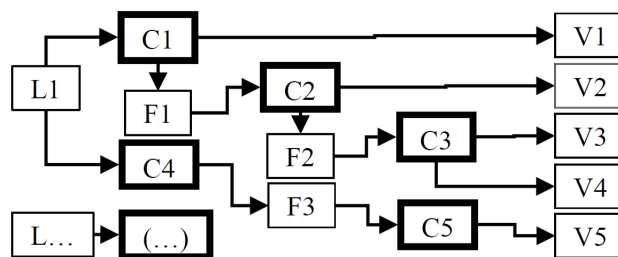


Рис. 1: Конвейер обработки данных.

В начале работы с системой пользователю предлагается список модулей для работы с данными (модуль загрузки – L). Пользователь может выбрать несколько модулей загрузки одновременно. Один модуль может работать с несколькими выборками данных заданными пользователем, которые образуют контейнеры данных (контейнеры – C). Контейнер с данными – это набор данных с идентификатором, определяющим тип данных. Далее для каждого контейнера, при необходимости, определяется набор фильтров (фильтры – F), результатом работы которых, так же является контейнер с

данными. Так же для контейнеров с данными, согласно их типу, пользователю предлагается набор модулей визуализации (модуль визуализации – V). Для одного контейнера можно добавить несколько модулей визуализации, а на этапе выбора из всего набора модулей обработки пользователю доступны только те, которые предназначены для типа данных из контейнера.

В клиентском интерфейсном приложении конвейер обработки данных удобно представлять в виде древовидной структуры (рис. 2).

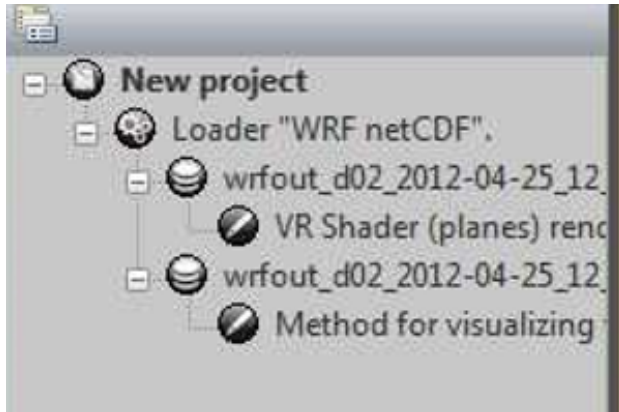


Рис. 2: Древовидное представление конвейера обработки данных в пользовательском приложении.

3. Система визуализации синоптических объектов с параллельно-распределенной обработкой

3.1 Обработка входных синоптических данных и функциональность системы

Данные перед визуализацией проходят несколько стадий:

1. Сбор данных от источников (искусственные спутники земли, зонды, метеостанции);
2. Пересылка к удаленному хранилищу (уБД) – удаленная база данных, не являющаяся частью описываемой системы визуализации;
3. Запрос пользователя к уБД для выбора интересующих данных и получение данных с удаленного сервера: профилей спутникового мониторинга по протоколу HTTP/FTP/Samba и др., считывание данных из файловых контейнеров дата-центра (NETCDF погодной модели WRF) или хранящихся локально на жестких или сетевых дисках;
4. Конвертация, распаковка (данные могут храниться в текстовом или компактном бинарном виде). Спутниковые данные, данные зондов и метеостанций имеют узкоспециализированные форматы данных;

5. Предобработка, согласно заданным пользователем параметрам (применение моделей преобразования);
6. Пространственное и темпоральное восполнение (применение моделей восполнения), регуляризация, фильтрация;
7. Подготовка к визуализации – загрузка актуальных для текущего ракурса и момента времени данных в оперативную память ПК;
8. Визуализация.

3.2 Сервер

Серверная часть системы реализована в программном приложении, объединяющем в себе базу данных и супервизор. Супервизор координирует работу распределенной системы, формируя пакеты с заданиями для вычислительных узлов и их модулей. База данных используется для централизованного сбора и хранения результатов обработки данных с вычислительных узлов. В БД хранятся только актуальные данные для текущих настроек конвейера. Если пользователь поменял параметры какого-либо вычислительного модуля, то все контейнеры данных, которые от него зависят, очищаются и пересчитываются заново, после чего актуализируются на стороне клиентского интерфейсного приложения для визуализации.

Протокол обмена данными в базе данных реализован на TCP, который обеспечивает доставку данных без потерь. Порядок работы:

1. Система запускается на одном или нескольких ПК, объединенных ЛВС, на каждом ПК с клиентским вычислительным приложением должны быть одинаковые наборы вычислительных модулей. Для функционирования системы должны быть запущен один сервер, как минимум одно вычислительное приложение и одно интерфейсное приложение;
2. Пользователь задает конфигурацию конвейера в клиентском приложении;
3. Супервизор запускает конвейер расчета актуальных данных – рассылает пакет с заданиями для старта расчетов вычислительными приложениями;
4. Супервизор сообщает интерфейсному приложению об окончании расчетов;
5. Интерфейсное приложение забирает те контейнеры с данными, которые согласно заданной конфигурации участвуют в визуализации;
6. Интерфейсное приложение запускает процесс рендеринга;
7. Если пользователь меняет конфигурацию конвейера или настройки вычислительных модулей, то супервизор начинает процесс с пункта 3, иначе вычислительный процесс останавливается, работает только интерфейсное приложение.

Схема распределения вычислений исключает возможность запроса на запись одного и того же набора данных из разных вычислительных приложений, так как между вычислительными узлами распределяется работа по расчету данных на разные моменты «времени» (имеется в виду дата/время съемки данных). Параллелизм в рамках расчета одного момента «времени» идет на уровне вычислительного модуля. Таким образом, удалось избежать схемы с блокировкой базы данных и, соответственно, сбор данных от разных вычислительных приложений может происходить независимо в параллельном режиме.

3.3 Клиентское интерфейсное приложение

Актуальными считаются данные только для выбранной временной точки динамических данных. Таким образом, для темпоральной анимации на каждом шаге временной шкалы необходима работа вычислительного конвейера. В этом случае узким местом будет пересылка данных по ЛВС. Чтобы избежать замедления темпоральной визуализации в клиентском интерфейсном приложении предусмотрен кэш данных. При первом проходе по темпоральной шкале, все полученные данные в ключевых моментах сохраняются на жестком диске. После первого прохода, когда все данные собраны локально в бинарном виде, визуализация выполняется на большой скорости. Ключевые моменты – это моменты времени, в которые были сделаны замеры набора данных. Промежуточные сетки данных, между ключевыми моментами, вычисляются на лету на клиентской стороне. При завершении работы приложения все временные файлы удаляются. Эксперименты показали, что скорости работы современных ЦП и жестких дисков достаточно для плавной темпоральной анимации данных из кэша, так как основная вычислительная нагрузка идет именно в конвейере обработки данных.

Если пользователь изменил параметры фильтров или других вычислительных модулей, то, как было сказано выше, происходит пересчет данных, но порядок расчета ключевых моментов происходит не от самого раннего к самому позднему, а в обе стороны от текущего положения темпорального ползунка. Это сделано для того, чтобы пользователь как можно быстрее мог продолжать визуальные исследования данных в окрестности текущего «времени». Функциональные возможности и инструментарий визуального анализа клиентского интерфейсного приложения описывается в предыдущих работах авторского коллектива [2,3]. Примеры визуализации данных демонстрируются на рис. 3 и 4.

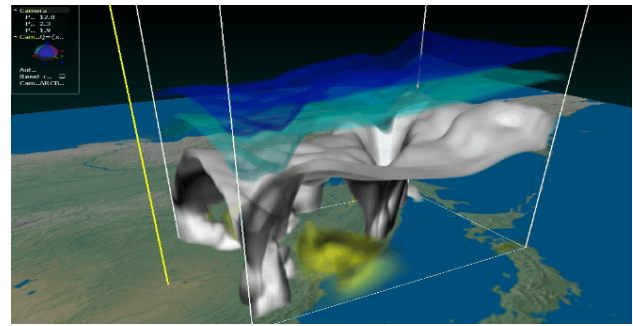


Рис. 3: Визуализация температурных слоев, полученных математической мезомасштабной прогнозной моделью WRF-ARW рассчитанных для прогноза погоды.

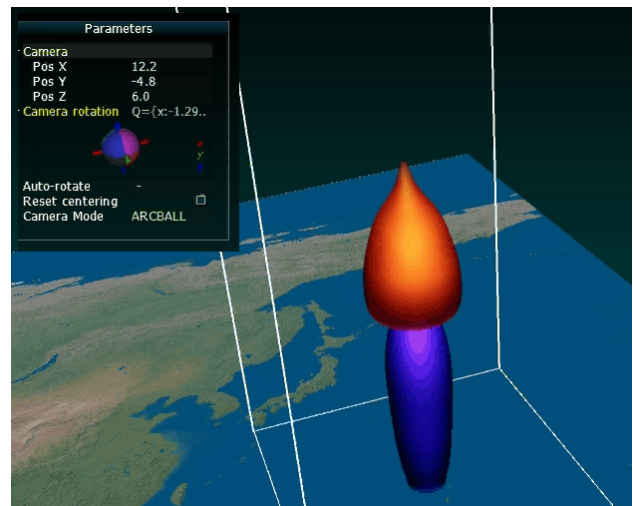


Рис. 4: Визуализация температурных аномалий в ядре тропического циклона.

3.4. Клиентское вычислительное приложение

Клиентское вычислительное приложение является интерфейсом, связывающим вычислительные модули с распределенной БД. Обеспечивает кэш (копии только необходимых данных из БД), занимается пересылкой данных и поддержанием их в актуальном состоянии.

Для обеспечения темпоральной анимации вычислительный конвейер работает только для ключевых моментов времени. Пакет с заданиями, рассылаемый диспетчером для вычислительного узла содержит описание конвейера, со всеми параметрами вычислительных модулей, заданными пользователем в интерфейсном приложении, количество темпоральных срезов (измерений) и порядковый номер, присвоенный узлу. Каждый вычислительный узел, получив пакет, выполняет полный проход по конвейеру для своей части данных и отправляет результат в базу данных. Зная свой порядковый но-

мер и количество темпоральных срезов, узел обрабатывает только те, для которых остаток от деления номера среза на количество узлов дает порядковый номер этого узла. При этом нумерация узлов и срезов начинается с нуля. Пример: есть температурные профили, получаемые раз в день на протяжении 5 дней, то есть 5 темпоральных срезов, так же известно, что вычислительных узлов 3, тогда нулевой узел обрабатывает срезы 0 и 3, первый узел 1 и 4, второй узел обрабатывает 2 срез. Так как в типичной задаче количество срезов исчисляется десятками, а количество узлов значительно меньше, то эффективность схемы не страдает.

4. Заключение

Вычислительные эксперименты проводились с данными тропических циклонов и спутниковых профилей с применением разных алгоритмов визуализации: объемный рендеринг на шейдерах и 3D текстурах, визуализация изоповерхностей методом «марширующих кубиков», визуализация движения центра ТЦ. Использовались вычислительные модули конвертации профилей спутникового мониторинга в регулярную решетку, восполнение недостающих данных и расчет аномалий в центрах тропического циклона.

Автоматическая загрузка данных о спутниковых профилях осуществлялась по протоколу FTP из центра спутникового мониторинга ИАПУ ДВО РАН.

Также были проведены эксперименты с данными прогнозной WRF модели, соответствующих области Приморского края и временному периоду с 25.04.2012 по 29.04.2012.

Эксперименты подтвердили эффективность системы построенной по предложенной технологии. Разработанная система визуализации позволила полностью в автоматическом режиме делать запросы к удаленному хранилищу данных и визуализировать полученные данные в рамках одного пользовательского интерфейса.

В перспективе планируется расширение множества модулей загрузки синоптических данных из мест общего пользования (через интернет), создание модулей визуализации узкоспециализированных данных (наложение погодных карт, карт ледяного покрова и т.д.), разработка механизма расширения измерительного инструментария с помощью модулей по технологии plug-in.

Литература

- [1] David Camp, Christoph Garth, Hank Childs, Dave Pugmire, and Kenneth I. Joy. Streamline Integration Using MPI-Hybrid Parallelism on a Large Multicore Architecture. *IEEE Transactions on Visualization and Computer Graphics*, 17(11):1702–1713, November 2011.
- [2] Бобков В. А., Мельман С. В., Май В. П. Визуализация динамики циклонов. // *Информационные технологии* №2, 2012, стр. 49-54.
- [3] Бобков В.А., Мельман С.В. и др. Программный комплекс визуализации синоптических данных. // *Информатика и системы управления*. 2014. № 1 (39). С. 141-150.
- [4] Конышин И.Н., Сушко Г.Б., Харченко С.А., Трёхуровневая MPI+TBV+CUDA параллельная реализация блочного итерационного алгоритма решения СЛАУ для мелкоблочных неструктурированных разреженных матриц, *Научный сервис в сети Интернет: поиск новых решений: Труды Международной суперкомпьютерной конференции*, Москва, Издательство МГУ, 2012, с. 522 — 528.
- [5] Кривов М.А., Притула М.Н., Гризан С.А., Иванов П.С., Оптимизация приложений для гетерогенных архитектур. Проблемы и варианты решения, // *Информационные технологии и вычислительные системы*, 2012, № 3.
- [6] Чусов А.А., Стаценко Л.Г. Разработка адаптируемых распределенных систем параллельного моделирования, анализа и визуализации физических полей: монография [Электронный ресурс] / Инженерная школа ДВФУ. – Владивосток: Дальневост. федерал. ун-т, 2014. [166 с.]. (Серия «Автоматика, электроника и средства связи», ISSN 2307-8006), ISBN 978-5-7444-3324-6.

Моделирование и визуализация работы энергетической установки сложной формы в режиме стереоанимации*

С.В. Андреев¹, А.Е. Бондарев¹, А.В. Бондаренко², Ю.В. Визильтер²,
 В.А. Галактионов¹, А.В. Гудков², С.Ю. Желтов², В.Т. Жуков¹,
 Е.Б. Иловайская², В.А. Князь², К.В. Мануковский¹, Н.Д. Новикова¹,
 М.В. Ососков², Н.Ж. Силаев², О.Б. Феодоритова¹, Н.А. Бондарева¹
 bond@keldysh.ru

¹Институт прикладной математики им. М.В.Келдыша РАН, Москва, Россия;

²ФГУП ГосНИИАС, Москва, Россия

Работа посвящена вопросам моделирования и визуального представления физических характеристик энергетической установки (ЭУ), которая обладает сложной формой. Основной задачей работы является моделирование рабочего режима установки под нагрузкой. Рассматриваются вопросы построения математической модели и трехмерной расчетной сетки, как статической, так и динамической, а также вопросы организации визуального представления, позволяющего верификацию построенной модели и полученных результатов. Реализована полноценная технологическая цепочка алгоритмов, с помощью которой проводится моделирование ЭУ от построения исходной CAD модели до организации визуального представления результатов в стереорежиме. Итоги численного моделирования являются базовыми для последующего решения задачи глобальной оптимизации формы ЭУ с целью выработки оптимальных рабочих параметров.

Ключевые слова: визуализация многомерных данных, энергетическая установка, вычислительная газовая динамика, OpenFOAM

Simulation and visualisation of power plant with complicated shape in stereoanimation mode*

S. V. Andreev¹, A. E. Bondarev¹, A. V. Bondarenko², Yu. V. Vizilter²,
 V. A. Galaktionov¹, A. V. Gudkov², S. Yu. Zheltov², V. T. Zhukov¹,
 E. B. Ilovayaskaya², V. A. Knyaz², K. V. Manukovsky¹, N. D. Novikova¹,
 M. V. Ososkov², N. Zh. Silaev², O. B. Feodoritova¹, N. A. Bondareva¹

¹KIAM RAS, Moscow, Russia;

²GOSNIAS, Moscow, Russia

The work is devoted to simulation and visualization of power plant (PP), which is the installation of vertical type and has a very complicated shape. The main goal is simulation of operating mode under load. Constructing of mathematical model and 3D calculation grid (both static and dynamic ones) as well as organization of the visual presentation to verify constructing model and obtained results are considered. Complete technological pipeline that allows simulating PP from constructing original CAD model up to visualization of results in stereo mode is implemented. The results of numerical simulation are the base to solve further the problem of global optimization of the PP shape in order to develop the optimal operating parameters.

Keywords: multidimensional data visualization, power plant, CFD problems, OpenFOAM

Введение

Основной задачей предлагаемой работы являлось создание полной технологической цепочки для аэродинамического моделирования работы объекта сложной формы (в нашем случае – энергетической установки), а также последующего анализа и оптимизации основных важнейших параметров, модификации формы объекта с целью улучшения его производственных характеристик.

Поставленная задача включает в себя несколько основных этапов, а именно, разработка приемлемой

CAD модели, подбор адекватной математической модели, построение сетки на поверхности объекта и в окружающей его области, выбор вычислительной модели, проведение серии вычислительных экспериментов и их верификация, и наконец, визуализация и анализ полученных данных. Ниже мы остановимся на каждом из этих этапов подробнее.

Следует заметить, что описанный цикл может быть повторен многократно, что диктует необходимость использования мощных компьютеров. Все расчеты проведены в ИПМ им. М.В. Келдыша РАН на вычислительном кластере К-100 [1].

Работа выполнена при поддержке грантов РФФИ (проекты 13-0100367а и 14-01-00769а) и опубликована при финансовой поддержке РФФИ, грант 15-07-20347.

Построение CAD модели и расчетных сеток

Аэродинамический расчет реального трехмерного объекта может быть успешным только при наличии геометрической сетки приемлемого качества, и подготовка такой сетки является одной из самых трудоемких частей вычислительного (равно как и всего технологического) процесса.

Сначала создается сетка на поверхности, ограничивающей рассматриваемое трехмерное тело. Как правило, поверхность импортируется из CAD-пакета (полностью или частично) и в случае необходимости может быть отредактирована. При этом набор предлагаемых форматов описания геометрии достаточно широк. Таким образом, основой для построения объемной сетки вне тела является одна или несколько триангулированных поверхностей. Основные требования, предъявляемые к поверхностным сеткам: отсутствие самопересечений, замкнутость и некоторые другие. При работе с реальными промышленными объектами получаемые CAD-поверхности удовлетворяют названным критериям в редких случаях. Как правило, для получения объемной сетки нужного качества необходима дополнительная подготовка поверхности. Для достижения указанной цели используется ряд автоматических инструментов с широким набором средств контроля качества. Пример триангулированной сетки на поверхности приведен на рис. 1.

Объемная сетка для моделирования течений жидкостей и газов, как правило, состоит из двух основных частей: призматической вблизи обтекаемых поверхностей и произвольной многогранной на удалении от поверхностей. Для создания пристеночного призматического слоя с заданным числом ячеек в направлении от стенки, с заданным законом роста размеров ячеек и полной толщиной слоя используются два возможных подхода. В первом строится поверхностная сетка, отстоящая от исходной на полную толщину слоя. Получившийся таким образом пристеночный объем разрезается на заданное число призматических слоев. Такой подход обычно не гарантирует создание призматического слоя с заданными параметрами и требуемой полной толщиной в областях со сложной геометрией. В таких случаях предпочтительней использовать метод последовательного создания призматических слоев, допускающий модификацию поверхностной сетки текущего слоя (коллапс ребер, разрезание ячеек, оптимизация положения вершин и т.д.), что позволяет создать слой призматических ячеек, проникающий значительно дальше от стенок вглубь расчетной области.

Для заполнения внепристеночного объема также могут использоваться различные подходы: тетраэдры, усеченные ячейки (гексагональные во внут-

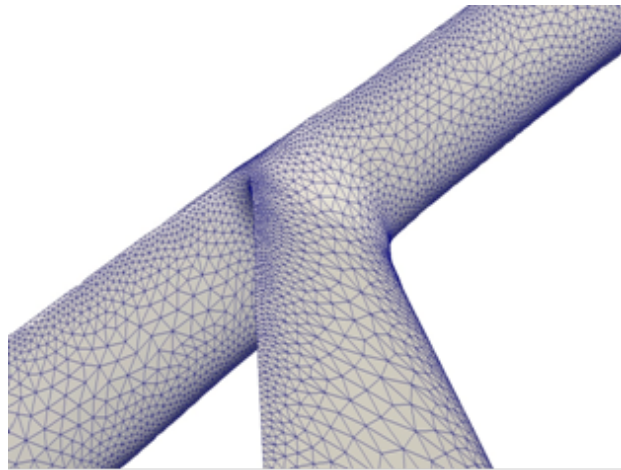


Рис. 1: Пример построенной поверхностной сетки.

ренней части и многогранные на границе расчетной области), а также многогранные ячейки во всей счетной области. Программный пакет OpenFOAM, который использовался при численном моделировании, позволяет проводить расчеты на сетках, состоящих из произвольных многогранных ячеек, ограниченных произвольным числом многоугольных граней.

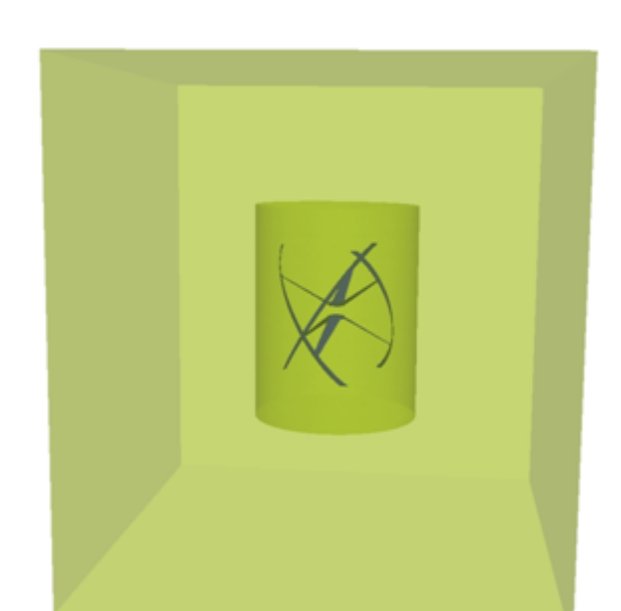


Рис. 2: Пример двухобластной задачи.

Моделирование физических процессов при движении объекта требует построения как минимум двухобластной объемной сетки при условии их стыковки «узел в узел» на границе областей. Пример такой расчетной области, состоящей из двух подобластей, и примененной нами для расчета физических характеристик движущегося объекта, показан на рис. 2.

Вычислительный эксперимент

Математическая постановка.

В настоящее время течения газовой сплошной среды рассчитываются на основе осредненных уравнений Навье-Стокса, в которые введены дополнительные члены, учитывающие эффекты турбулентности [2]. Система уравнений дополняется уравнением состояния: в простейшем случае – уравнением состояния идеального газа, который, однако, может служить хорошим первоначальным приближением. Вид турбулентных компонент не является универсальным и зависит от модели турбулентности, выбираемой с учетом свойства реальных физических течений в некотором диапазоне параметров. Вопрос выбора модели турбулентности вынесен за рамки данной работы. На первоначальном этапе рассматриваются ламинарные течения.

Вычислительная модель.

Процесс математического моделирования предполагает выбор вычислительной модели, определение расчетных областей и подобластей, граничных и начальных условий, а также проведение цикла методических и параметрических расчетов. В современных расчетах необходимо также учитывать возможность хорошего распараллеливания алгоритма.

Для проведения расчетов мы воспользовались открытой интегрируемой платформой для численного моделирования задач механики сплошной среды OpenFOAM. Элементы OpenFOAM активно используются в промышленности, в академической сфере и экспертном сообществе, в частности для расчета энергетических характеристик ветроустановок [3]. Пакет представляет собой множество готовых моделей, но будучи открытым, позволяет вносить изменения и дополнения. В качестве базовой мы использовали модель rhoCentralFoam, которая реализует так называемую «центрально-противопотоковую» схему и, в полном соответствии с названием, представляет собой комбинацию центрально-разностной и противопотоковой схем [4].

Преимущество указанных схем состоит в том, что, применяя соответствующую технику уменьшения численной вязкости, можно добиться хорошей разрешимости и для разрывных решений (ударных волн в газовой динамике) и для решений, где основную роль играют вязкие явления. Суть центрально-противопотоковых схем состоит в специальном выборе контрольного объема. За прошедшее десятилетие этот метод интенсивно развивался и продемонстрировал свои достоинства при проведении предсказательных вычислений.

Для анализа движущейся установки использовалась модель rimpleDyMFoam, в основе которой лежит алгоритм PIMPLE. Внутренняя подобласть (в

нашем случае это цилиндр) вращалась с заданной скоростью и на каждом шаге производилась корректировка потоков на интерфейсе.

Была проведена серия специальных методических расчетов, которая позволила сформировать топологию, форму и размеры расчетной области, наиболее соответствующие целям проводимого компьютерного моделирования и принятой постановке задачи. При этом первоначально рассчитывалась и анализировалась только часть ЭУ, а именно, одна неподвижная лопасть [5].

Практические результаты.

Среди аэродинамических характеристик нас интересует прежде всего полная аэродинамическая сила и крутящий момент [6]. При рассмотрении одной неподвижной лопасти мы изучали влияние на эти характеристики скорости набегающего потока, его направления, температуры, гравитации. В методических целях проведены расчеты с измельчением расчетной сетки для ориентировочной оценки уровня измельчения сетки, необходимого для масштабных расчетов полной компоновки ЭУ.

Полученные в расчетах силовые характеристики удовлетворительно согласуются с приближенной эмпирической оценкой. Предварительные выводы, которые можно сделать из проведенных серий расчетов обтекания ламинарным потоком лопасти ЭУ в неподвижном состоянии, следующие: рост скорости обтекания лопасти приводит к увеличению силы и момента, воздействующих на лопасть, что соответствует физической природе моделируемого процесса. Изменение температуры потока практически не оказывает влияния на силовые характеристики. Введение учета гравитации в расчетах приводит к небольшому изменению в силовых характеристиках.

При переходе к полномасштабным расчетам ЭУ в полной компоновке основной интерес представляет влияние на газодинамические характеристики изменение геометрии объекта: расположение лопастей относительно центральной оси установки, ширины и конструктивных особенностей самих лопастей и т.д.

Визуализация. Построение стерео-модели

Основным источником представления о физических процессах, происходящих при работе энергетической установки является визуальное представление. В данной работе использовались различные формы визуального представления: статическая форма для тщательного изучения распределения газодинамических функций по поверхности установки и в расчетной области, динамическая форма для изучения процессов перераспределения газодинамических функций в процессе движения и

объемная динамическая форма в виде стереопредставления.

Для статического визуального представления использовалась концепция прозрачной счетной области и рендеринга распределения газодинамических функций (давление, температура) по поверхности лопастей установки (рис. 3). Для одновременной оценки процессов распределения газодинамических функций на поверхности лопастей и в счетной области использовалось поперечное сечение (рис. 4).

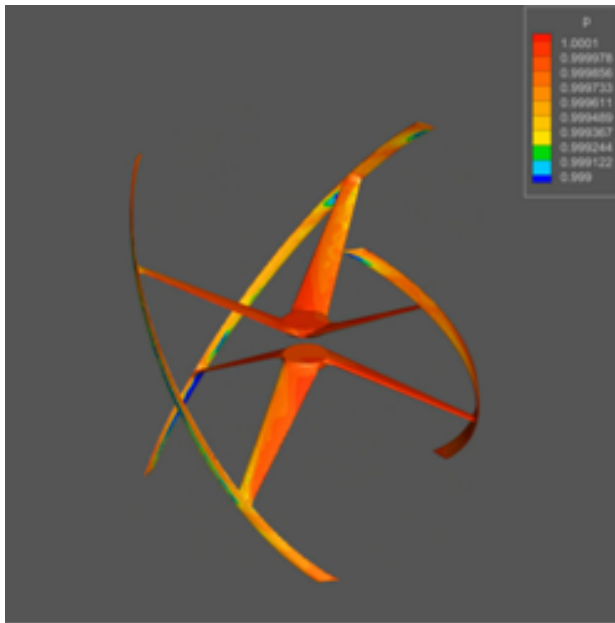


Рис. 3: Распределение давления по поверхности энергоустановки.

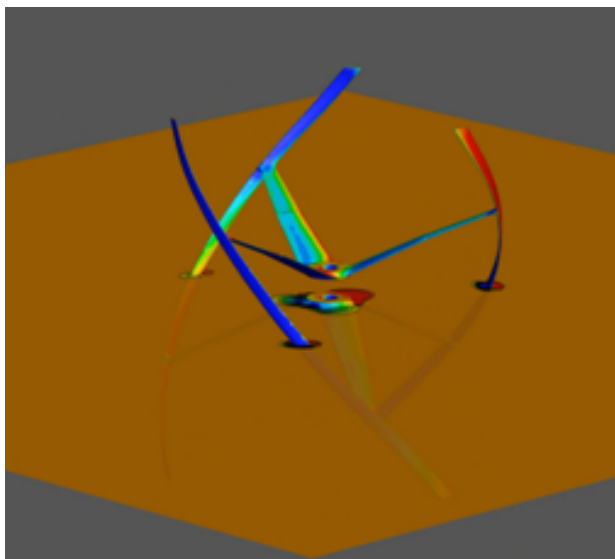


Рис. 4: Распределение давления по поверхности энергоустановки и в поперечном сечении.

Анимированное представление реализовывалось с помощью двух подходов. В первом подходе камера фиксировалась в определенном положении, и вращение узла лопастей происходило перед камерой. Это позволяло наблюдать процесс изменения газодинамических характеристик на всем узле в целом во время движения узла. Второй подход заключался в фиксации камеры в определенном положении относительно узла лопастей и вращении камеры вместе с узлом. Этот подход обеспечивал возможность контроля изменения газодинамических характеристик на просматриваемой лопасти в процессе движения.

Для визуализации результатов расчетов в объемном виде (построение стерео-модели) необходимо создание пары изображений, показывающие объект с различных ракурсов для правого и левого глаза соответственно, называемые стереопарой. То есть стереопара – это два плоских изображения одного и того же объекта (сюжета), имеющая различия между изображениями, призванные создать эффект объёма. В основном используются два метода создания стереопар: это смещение камеры параллельно на некоторое расстояние (линейная стереобаза), либо поворот камеры вокруг демонстрируемого объекта на определенный угол (угловая стереобаза). Нашей задачей являлось создание стерео-фильма, позволяющего продемонстрировать результаты расчетов (создание последовательности стереопар, каждая из которых является стереокадром фильма), поэтому для этих целей наиболее подходящим является метод "облета" камеры вокруг объекта, то есть угловая стереобаза. Практически любая САД-система позволяет визуализировать трехмерный объект с заданными координатами положения по отношению к пользователю. Поэтому задача сводилась к построению последовательности изображений САД модели с зафиксированной осью Z, проходящей через ее геометрический центр, и смещением на определенный угол. После каждого такого смещения проводился рендеринг модели, позволяющей достичь фотореалистичности, и сохранение результата в виде графического файла в высоком разрешении. Для левого стереоканала устанавливалась начальная позиция, отличающаяся от правого стереоканала на угол, определяющий параллакс, и позволяющая добиться объемности объекта в результирующем стереофильме (рис. 5).

Данный метод привлекателен еще и тем, что для статической модели, то есть с неизменной геометрией и цветовым распределением, возможно использование уже рассчитанных кадров, например, если для левого стереоканала мы используем 1,2,3,4,... кадры, то для правого стереоканала достаточно начать последовательность со второго или с третьего кадра (2,3,4,5...), автоматич-

чески обеспечивая сдвиг объекта на необходимый угол стереобазы [7].

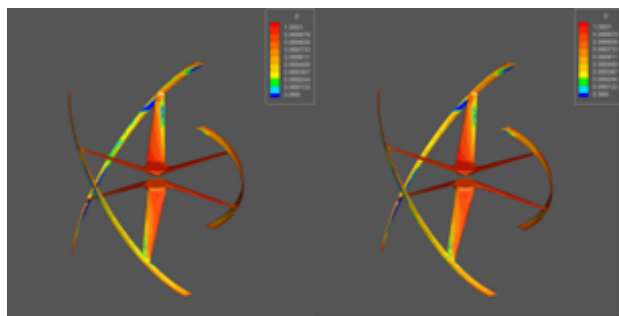


Рис. 5: Стереокادر с угловой стереобазой, демонстрирующий распределение давления по поверхности энергоустановки.

Таким образом, был создан ряд стерео-фильмов, визуализирующий вращение модели в объемном представлении при задании различных внешних условий.

Выводы

Разработанная технологическая цепочка позволяет проводить полноценное вычислительное исследование и последующий анализ работы энергетической установки, начиная от создания САД модели и

заканчивая стереометрической визуализацией. Это дает возможность перейти к задачам оптимизации для уменьшения энергетических потерь и увеличения КПД установки.

Литература

- [1] www.kiam.ru/MVS/resources – Гибридный вычислительный кластер К-100.
- [2] *Л.Д.Ландау, Е.М. Лившиц.* Гидродинамика. – Москва: Наука, 1986
- [3] 1st Symposium on OpenFOAM in Wind Energy 20th-21st of March 2013 – Oldenburg.
- [4] *A.Kurganov, E.Tadmor.* New high resolution central schemes for nonlinear conservation laws and convection-diffusion equations, *J.Comput. Phys.*, Vol.160, 2000, pp.241-282
- [5] *Бондарев А.Е., Жуков В.Т., Мануковский К.В., Новикова Н.Д., Феодоритова О.Б.* Разработка и организация математического моделирования обтекания неподвижной лопатки энергетической установки. – Препринт N 60 ИПМ им. Келдыша, 2014
- [6] *Краснов Н.Ф.* Аэродинамика, Высшая школа. – Москва, 1980.
- [7] *Андреев С.В., Бондарев А.Е., Михайлова Т.Н., Рыжова И.Г.* Организация стереопредставлений в задачах синтеза фотореалистичных изображений и научной визуализации / Препринт ИПМ им. М.В. Келдыша РАН, N 61, 2010, 14 с.

Визуальный анализ кластерных структур в многомерных объемах данных*

А.Е. Бондарев, В.А. Галактионов

bond@keldysh.ru | vlgal@gin.keldysh.ru

Институт прикладной математики им. М.В. Келдыша РАН, Москва, Россия

Работа посвящена вопросам построения алгоритмов для визуального анализа кластерных структур в многомерных объемах данных. Целью работы является построение комплекса алгоритмов визуализации и визуальной аналитики, позволяющего изучение кластерных структур в многомерных объемах данных без применения алгоритмов кластеризации, вносящих изменения в исходные данные. Для анализа кластерных структур в многомерном объеме данных предлагается использовать методы отображения точек исходного многомерного пространства на вложенные в это пространство многообразия меньшей размерности. Данный подход базируется на построении самоорганизующихся карт SOM, применении метода главных компонент PCA и построении упругих карт Elastic Maps с последующей реализацией процедуры отжига для этих карт. Для реализации полной и последовательной обработки многомерного массива данных вышеупомянутые методы и подходы выстраиваются в последовательность применяемых методов и алгоритмов, образуя единую технологическую цепочку обработки данных. Применение подобной цепочки позволяет получить информацию о кластерной структуре исследуемого объема многомерных данных на разных уровнях глубины анализа и детализации информации.

Ключевые слова: многомерные данные, кластерные структуры, визуальный анализ

Visual Analysis of Cluster Structures for Multidimensional Data*

A.E. Bondarev, V.A. Galaktionov

Keldysh Institute of Applied Mathematics RAS, Moscow, Russia

The paper considers design of algorithms intended for visual analysis of clusters in multidimensional data volumes. The paper is aimed to design of a set of visualization and visual analytics methods for cluster structure studies without applying of clusterization methods influencing at original data. To analyze clusters in original data volume we propose to use the methods of original data points mapping to enclosed manifolds having less dimensionality. This approach is based on self-organized maps (SOM) design, principal components analysis (PCA) and application of elastic maps with further varying of elasticity parameters for the last ones. To provide complete processing of original data volume all mentioned above methods should be organized as a pipeline. The applying of such pipeline allows one to get insight of cluster structures at the different levels of details.

Keywords: multidimensional data, cluster structures, visual analysis

Введение

Одной из основных современных задач практически во всех областях человеческой деятельности на сегодняшний день является анализ многомерных данных. Многомерные данные являются результатами численных исследований, технических показателей, обобщением экономической и финансовой информации и т.д. Необходимость обработки, анализа и адекватной трактовки этих данных породила такую интенсивно развивающуюся научную дисциплину, как анализ многомерных данных (Data Analysis). Одной из важнейших составляющих это направление дисциплин является кластерный анализ [1, 2], рассматривающий различные способы группировки объектов внутри облака многомерных данных. Методов и алгоритмов кластерного анализа на современном этапе существует очень много, они постоянно развиваются и отли-

чаются большим разнообразием. Это могут быть, например, алгоритмы, реализующие полный перебор сочетаний объектов или осуществляющие случайные разбиения множества объектов. Многообразие алгоритмов кластерного анализа обусловлено также множеством различных критериев, выражающих те или иные аспекты качества автоматического группирования. Надо заметить, что ряд источников [2, 3] указывает на ряд специфических особенностей методов, алгоритмов и подходов кластерного анализа, которые исследователь должен учитывать в обязательном порядке:

- А) Многие методы кластерного анализа – довольно простые процедуры, которые, как правило, не имеют достаточного статистического обоснования. Они – не более чем правдоподобные алгоритмы, используемые для создания кластеров объектов.
- Б) Методы кластерного анализа разрабатывались для многих научных дисциплин, а потому несут на себе отпечатки специфики этих дисциплин. Это важно отметить, потому что каждая дис-

Работа выполнена при финансовой поддержке РФФИ, гранты 13-01-00367а, 14-0100769а и опубликована при финансовой поддержке РФФИ, грант 15-07-20347.

циплина предъявляет свои требования к отбору данных, к форме их представления, к предполагаемой структуре классификации. Так как кластерные методы порой не более чем правила для создания групп, то пользователь должен знать особенности области происхождения облака данных [2].

- В) Разные кластерные методы могут порождать и порождают различные решения для одних и тех же данных. Это обычное явление в большинстве прикладных исследований. Одной из причин неодинаковых решений является использование различных правил формирования групп.
- Г) Цель кластерного анализа заключается в поиске существующих структур. В то же время его действие состоит в привнесении структуры в анализируемые данные. Хотя цель кластеризации и заключается в нахождении структуры, на деле кластерный метод привносит структуру в данные и эта структура может не совпадать с искомой, «реальной». Кластерный метод всегда размещает объекты по группам, которые могут радикально различаться по составу, если применяются различные методы кластеризации. Ключом к использованию кластерного анализа является умение отличать «реальные» группировки от навязанных методом кластеризации данных.

Эти обстоятельства вызывают вполне естественное желание обойтись по возможности без вышеперечисленных сложностей. Возникает вопрос – нельзя ли получить представление о кластерной структуре рассматриваемого многомерного объема данных более простыми способами? При этом естественно хотелось бы оставить рассматриваемую исходную кластерную структуру в многомерном облаке данных без изменений, порождаемых применением алгоритмов кластеризации. Положительный ответ на этот вопрос дают алгоритмы понижения размерности и визуального представления многомерных данных во вложенных в исходный объем многообразиях меньшей размерности. К числу таких семейств алгоритмов можно отнести метод главных компонент и отображение исходного многомерного объема в главных компонентах (PCA) [2, 5], построение самоорганизующихся карт (SOM) [4], построение упругих карт (Elastic Maps) [5, 6] с разными свойствами упругости или эластичности и отображение исходного многомерного объема в этих картах.

Все эти методы позволяют тем или иным образом выделить из исходного многомерного объема данных содержащуюся в нем кластерную структуру, не внося практически изменений в исходные данные. Общим свойством всех трех вышеперечисленных подходов является реализация визуально-

го представления многомерного объема данных в виде проекции данных на вложенное многообразие меньшей размерности, обладающее следующими свойствами:

- размерность данного многообразия меньше либо равна трем, что дает возможность визуального представления на уровне человеческого восприятия;
- наличие устойчивых связей основных координатных направлений многообразия со всеми координатными направлениями в изучаемом многомерном объеме;
- возможность выделения наиболее информативных определяющих факторов в изучаемом объеме и отбрасывании малоценной информации.

Отметим, что общей идеей всех вышеперечисленных подходов является отображение многомерных данных в представимую человеком размерность, например, на плоскость так, чтобы точки данных, близкие на плоскости (на карте), были близки и в исходном пространстве. С помощью визуализации мы можем получать большое количество информации о данных сразу, без какой-либо обработки. Становятся видимыми области группировки данных и разреженные области. Упрощается решение задач классификации. Видно количество кластеров, их форма, взаимное расположение и т.д. Обратим внимание, что это естественная классификация данных, не требующая каких-либо специальных действий над исходными данными. Объединяя эти подходы в единую последовательность, можно обеспечить необходимый уровень представления о кластерной структуре данных с целью выделения отдельных кластеров и последующего выявления в них скрытых зависимостей между ключевыми параметрами. Последняя задача в этом случае решается в гораздо меньшем многомерном объеме, а не в облаке в целом. С точки зрения практического применения построение подобной последовательности предполагается к применению в задачах поиска пространственно-временных структур при обработке многомерных решений задач вычислительной газовой динамики [7, 8, 9], в задачах анализа многомерных баллистических данных и при обработке и анализе больших объемов текстовой и числовой информации.

Используемые методы

Данный раздел рассматривает общие основные подходы, применяемые для визуального выделения кластеров в многомерном объеме данных. К этим основным подходам относятся: построение самоорганизующихся карт (SOM), построение визуального представления многомерного облака данных в пространстве главных компонент (PCA), построение упругих карт (Elastic Maps). Все три вышеперечисленных подхода обладают рядом общих

свойств. Их применение не вносит изменений в исходную структуру данных, а значит, не создает искусственных кластерных структур и артефактов, подобно многим методам кластеризации. Все три основных подхода относятся к методам визуального представления, что позволяет провести оценку кластерной структуры облака данных максимально быстро и эффективно. Все подходы универсальны и позволяют параллельное применение к рассматриваемому облаку данных или последовательное в любых комбинациях. Также характерной чертой всех трех подходов является возможность для исследователя работать с данными в пространствах естественной для человеческого восприятия размерности – двумерных и трехмерных, а не с облаком данных абстрактной размерности. Приведем кратко основные черты и свойства алгоритмов и методов, составляющих три вышеперечисленных подхода.

Одним из первых и наиболее известных подходов подобного рода стал алгоритм построения самоорганизующихся карт SOM (Self-Organised Maps), предложенный Кохоненом [4]. В современном представлении карты SOM трактуются как двумерные сетки узлов, размещенные в изучаемом многомерном пространстве.

В упрощенном виде алгоритм построения карт SOM можно представить следующим образом. В пространство данных размещается двумерная решетка из элементов, которые способны сближаться или отдаляться друг от друга. Запускается итерационный алгоритм сближающий элементы решетки, соответствующие близким точкам в исходном многомерном пространстве данных, и отдаляющий элементы решетки, соответствующие далеким в исходном пространстве точкам. В результате получается картина, отражающая основные свойства изучаемого облака данных, в том числе и наличие кластерной структуры. Подчеркнем, применение карт SOM позволяет нам получить первичное представление о наличии кластеров в изучаемом объеме данных. Для получения более подробной информации о кластерной структуре изучаемого облака данных следует применять более совершенные подходы, такие, например, как метод главных компонент (PCA) и построение упругих карт.

Метод главных компонент (PCA) [2, 5] позволяет уменьшить размерность исследуемого многомерного объема данных с наименьшей потерей информации. Главные компоненты представляют собой ортогональную систему координат, в которой дисперсии компонент характеризуют их статистические свойства.

Суть метода состоит в переходе к новому ортогональному базису в рассматриваемом многомерном пространстве, оси которого ориентированы по на-

правлениям максимальной дисперсии набора входных данных, и ранжированию этого базиса в порядке убывания по признаку максимальной дисперсии вдоль осей базиса. Практическая реализация метода PCA сводится к выделению основных направлений (на практике двух или трех), понижению размерности многомерного облака данных до числа новых направлений, и проецированию всех данных на получившееся из новых направлений линейное многообразие. Это позволяет представить многомерный объем данных в проекции на плоскость или трехмерную область визуально. Визуальное представление, в свою очередь дает исследователю возможность понять структуру и суть многомерных данных, в том числе и кластерную структуру.

Другим важным подходом к нелинейному сокращению размерности данных является построение упругих карт (Elastic Map). Идеология и алгоритмы реализации этого подхода подробно представлены в работах [5, 6]. По построению, она представляет собой систему упругих пружин, вложенную в многомерное пространство данных. Данный подход основывается на аналогии с механикой: главное многообразие, проходящее через «сердечину» данных, может быть представлено как упругая мембрана или пластинка. В отличие от карт SOM, метод упругих карт изначально формулируется как оптимизационная задача, предполагающая оптимизацию заданного функционала от взаимного расположения карты и данных. При создании критерия оптимальности авторы включили в него среднее расстояние от точки данных до ближайшего узла карты. Варьирование параметров упругости (процедура отжига) заключается в построении упругих карт с последовательным уменьшением коэффициентов упругости, в силу чего карта становится более мягкой и гибкой, наиболее оптимальным образом подстраиваясь к точкам исходного многомерного объема данных. После построения упругую карту можно развернуть в плоскость для наблюдения кластерной структуры в изучаемом объеме данных. Построение упругих карт на сегодняшний день является широко распространенным методом анализа данных. Применение упругих карт позволяет более точно и четко определять кластерную структуру изучаемых многомерных объемов данных.

Построение технологической цепочки алгоритмов обработки многомерного объема данных

Представленные в предыдущем разделе основные подходы визуального отображения многомерного пространства на вложенную двумерную карту обладают разным уровнем сложности при реализации, дают разный уровень глубины и детализа-

ции при анализе кластерной структуры многомерного облака данных, имеют различные возможности «подстройки» к рассматриваемому облаку. В свою очередь при анализе кластерной структуры многомерного объема данных у исследователя имеются различные по глубине и детализации задачи. С этой точки зрения было бы весьма разумным выработать некоторый универсальный подход, позволяющий проведение анализа кластерной структуры в многомерном объеме данных на различном уровне информационной детализации. Подобный подход должен быть выстроен в виде некоторой последовательности применяемых методов и алгоритмов, обеспечивающих по мере применения все более детализированный и глубокий уровень анализа. Такая технологическая цепочка представляет собой конвейер обработки данных, где уровень глубины анализа и информационной детализации увеличивается по мере продвижения по цепочке. Пример цепочки такого рода, реализуемый на основе применения всех трех подходов, перечисленных в предыдущем разделе, представлен на рисунке 1.

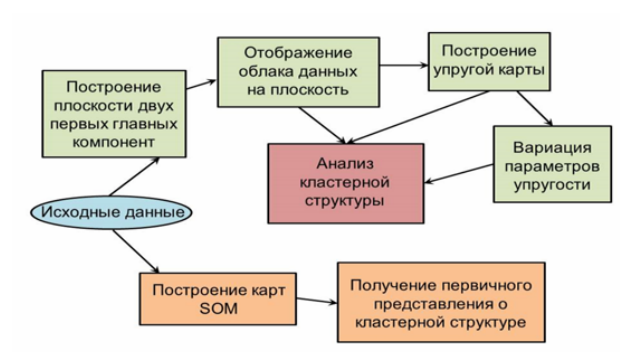


Рис. 1: Схема технологической цепочки исследования кластерной структуры.

Данная технологическая цепочка объединяет три подхода, обладающие общим и важнейшим для исследователя свойством – эти подходы не требуют применения алгоритмов кластеризации для анализа кластерной структуры в многомерном облаке данных и целиком основаны на визуальном представлении и анализе. На первом предварительном этапе используется самый грубый разведочный подход, состоящий в построении самоорганизующихся карт SOM. Он позволяет провести первичный, самый грубый анализ наличия кластерной структуры в облаке данных.

Для дальнейшего анализа часто бывает достаточно отобразить многомерное облако данных на плоскость двух первых главных компонент или в пространство первых трех главных компонент. Подобное визуальное представление обеспечивает следующий уровень анализа наличия и взаиморасположения кластеров.

Далее при необходимости обеспечить более глубокий уровень детализации и анализа, построенную плоскость двух первых главных компонент надо сделать гибкой, чтобы она могла наилучшим образом подстраиваться к исходному многомерному облаку данных. Для этого плоскость надо преобразовать в упругую карту. После построения упругой карты проводится ее развертка. Это обеспечивает более четкое (менее размытое) визуальное разделение кластеров. Далее следует вспомнить о том, что в отличие от предыдущих подходов, построение упругой карты является оптимизационной задачей, имеющей два внешних параметра – коэффициенты упругости карты λ и μ . Для того чтобы увеличить «резкость» изображения и обеспечить тем самым еще более четкое разделение кластеров, необходимо уменьшать коэффициенты упругости карты λ и μ . При этом карта становится более гибкой, лучше подстраивается к исходным многомерным данным, и при развертке обеспечивает максимально четкое разделение кластеров в рассматриваемой кластерной структуре.

Подытоживая вышесказанное, можно утверждать, что реализация подобной технологической цепочки в применении к практической задаче в подавляющем большинстве случаев позволит исследователю получить информацию о кластерной структуре многомерного облака данных на требуемом уровне. Причем, заметим, что все это происходит абсолютно без применения каких-либо простых или сложных алгоритмов кластеризации и без всякого искажения структуры исходных данных. К применению всего огромного аппарата алгоритмов кластеризации можно всегда перейти в том случае, если с помощью описанной технологической цепочки обработки, анализа и визуализации многомерных данных не удалось получить нужную информацию или достичь требуемого уровня глубины анализа.

Пример реализации

Покажем, как работает технологическая цепочка на примере конкретной задачи. В качестве тестовой задачи возьмем широко известный тестовый объем многомерных данных IRIS [5]. Данный объем представляет собой набор данных, основанных на измерениях характеристик растений – цветков ириса. Набор данных описывает три сорта ирисов и состоит из 150 точек в четырехмерном пространстве признаков.

Согласно описанию технологической цепочки, приведенному в предыдущем разделе, на предварительном этапе строится самоорганизующаяся карта SOM для рассматриваемого исходного набора данных. Это должно позволить получить следующую информацию – есть ли кластерная структура, как таковая, и сколько классов она в себе содержит. На

рисунке 2 представлены результаты построения самоорганизующейся карты SOM для рассматриваемого набора данных. Кластерная структура в объеме данных есть, и она содержит три класса.

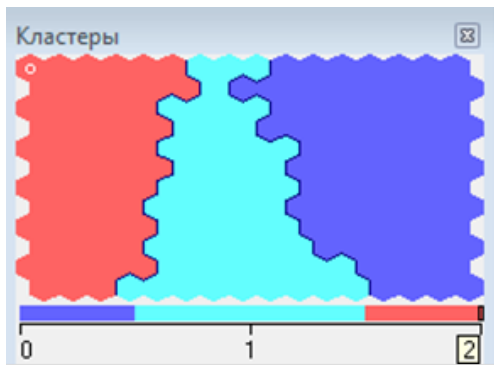


Рис. 2: Результат построения карт SOM.

Следуя технологической цепочке алгоритмов, далее необходимо применить метод главных компонент. На рисунке 3 приведено представление рассматриваемого объема данных в объеме, образованном тремя первыми главными компонентами. Различные классы выделены цветами – красным, синим и зеленым. Видно, что красные точки отделены от остальных достаточно четко, а синие и зеленые смешиваются. Применение метода главных компонент позволяет получить представление о взаиморасположении кластеров в многомерном пространстве.

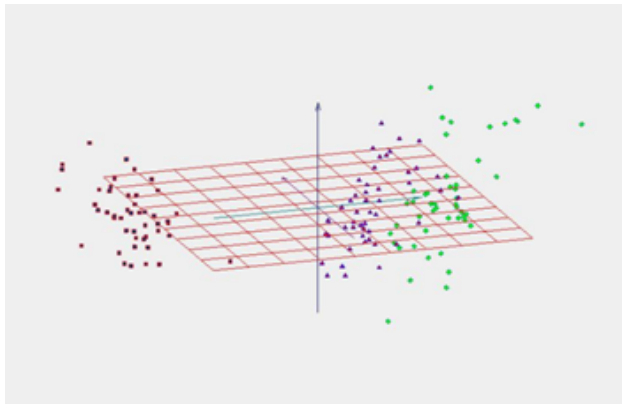


Рис. 3: Трехмерное представление исследуемого объема данных в пространстве главных компонент.

Двигаясь по цепочке дальше, мы ставим целью получить более четкое визуальное представление о разделении данных внутри исследуемого объема на кластеры. С этой целью проводится построение упругих карт и проецирование точек исследуемого объема на поверхности этих карт. На рисунке 4 представлена упругая карта в пространстве главных компонент с раскраской по значению плотно-

сти данных. Это так называемая «жесткая» упругая карта, построенная при значениях коэффициентов упругости $\lambda = 5$, $\mu = 5$.

Видно, как плоскость главных компонент изгибается, стараясь наилучшим образом подстроиться к многомерному объему данных. На рисунке 5 представлена развертка упругой карты без раскраски. Разделение синих и зеленых точек улучшилось, впрочем, как и разделение на классы в целом. Картина разделения как бы «проявляется», становясь все более четкой.

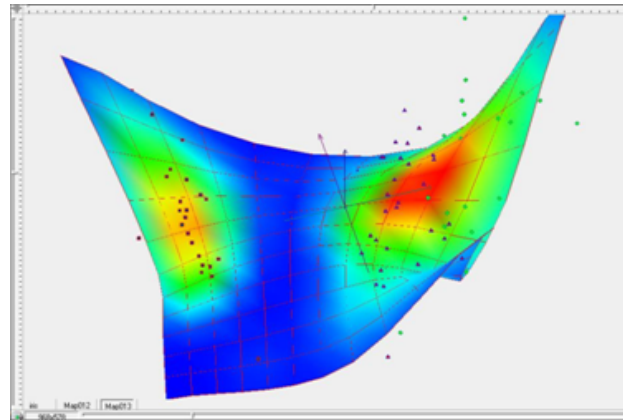


Рис. 4: Построение упругой карты при $\lambda = 5$, $\mu = 5$ в пространстве главных компонент.

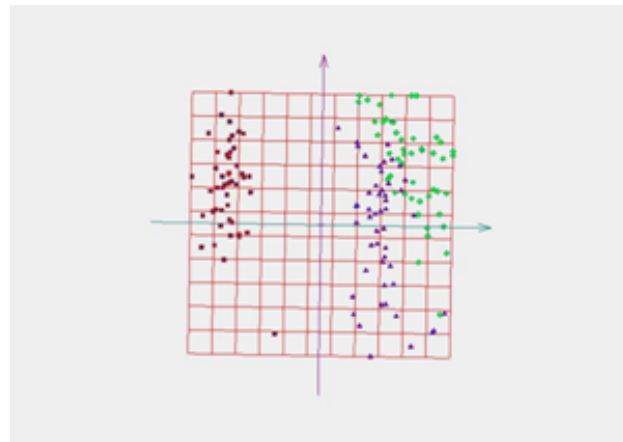


Рис. 5: Развертка «жесткой» упругой карты при $\lambda = 5$, $\mu = 5$.

Для дальнейшего улучшения четкости разделения на кластеры в исследуемом многомерном объеме данных применим так называемую процедуру отжига. Для этой цели будем уменьшать коэффициенты упругости λ и μ , делая карту более «мягкой». Это обеспечивает нам улучшенную адаптацию упругой карты к данным рассматриваемого многомерного объема. Зададим для дальнейшего построения их значения равными $\lambda = 0.01$, $\mu = 0.01$. Визуальные представления упругой карты

для этих параметров представлены на рисунках 6 и 7, представляющих поверхность упругой карты, раскрашенную в соответствии с плотностью данных, и развертку упругой карты в плоскость. Разделение на классы стало еще более четким и заметным (зеленые и синие точки на рисунке 7).

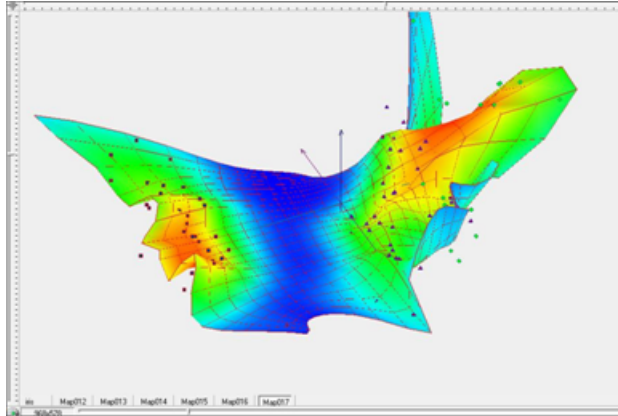


Рис. 6: Построение упругой карты при $\lambda = 0.01$, $\mu = 0.01$ в пространстве главных компонент.

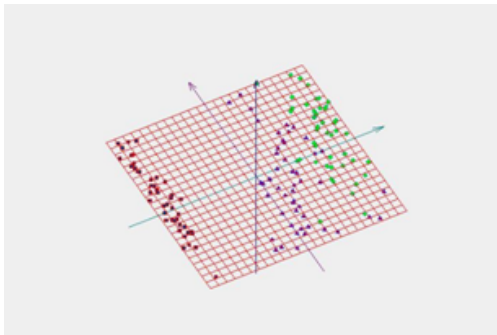


Рис. 7: Развертка упругой карты при $\lambda = 0.01$, $\mu = 0.01$.

Таким образом, на примере известного тестового объема многомерных данных проиллюстрирована вся технологическая цепочка, представленная в предыдущем разделе. Применение построения самоорганизующейся карты SOM позволило установить наличие кластерной структуры в исследуемом многомерном объеме данных и определить число классов в структуре. Отображение в пространство первых главных компонент позволило получить представление о форме и взаиморасположении кластеров в структуре. Применение построения упругих карт и реализация последующей процедуры отжига путем вариации коэффициентов упругости в сторону уменьшения позволили добиться четкого разделения данных на кластеры.

Выводы

Для анализа кластерных структур в многомерном объеме данных предлагается использовать методы

отображения точек исходного многомерного пространства на вложенные в это пространство многообразия меньшей размерности. Данный подход базируется на построении самоорганизующихся карт SOM, применении метода главных компонент PCA и построении упругих карт Elastic Maps с последующей реализацией последовательного уменьшения коэффициентов упругости, в силу чего карта становится более мягкой и гибкой, наиболее оптимальным образом подстраиваясь к точкам исходного многомерного объема данных. Для реализации полной и последовательной обработки многомерного массива данных эти методы и подходы выстраиваются в последовательность, образуя таким образом единую технологическую цепочку обработки данных. Применение подобной цепочки позволяет получить информацию о кластерной структуре исследуемого объема многомерных данных на разных уровнях глубины анализа и детализации информации. При этом не вносятся искажения в исходные данные.

Данная работа содержит описание построения подобной технологической цепочки и способов ее применения для анализа многомерных объемов информации.

Литература

- [1] *Мандель И.Д.* Кластерный анализ. – М.: Финансы и статистика, 1988. – 176с.
- [2] *Ким Дж., Мюллер Ч. и др.* Факторный, дискриминантный и кластерный анализ. – М.: Финансы и статистика, 1989. – 216с.
- [3] *Дюран Б., Одель П.* Кластерный анализ. – М.: Статистика, 1977. – 128с.
- [4] *Kohonen T.* Self-Organizing Maps. – Berlin – Heidelberg: Springer, 1997.
- [5] *Зинovieв А.Ю.* Визуализация многомерных данных. – Красноярск: Изд. КГТУ, 2000. – 180с.
- [6] *Gorban A., Kegl B., Wunsch D., Zinovyev A. (Eds.)* Principal Manifolds for Data Visualisation and Dimension Reduction. – Berlin – Heidelberg – New York: Springer, LNCSE 58, 2007.
- [7] *Бондарев А.Е., Галактионов В.А.* Современные направления развития визуализации данных в вычислительной механике жидкости и газа // Научная визуализация. – М.: НИЯУ МИФИ, 2013. – Т.5, №4. – С. 18-30.
- [8] *Бондарев А.Е.* Анализ многомерных данных в задачах вычислительной газовой динамики // Научная визуализация. – М.: НИЯУ МИФИ, 2014. – Т.6, №5. – С. 59-66.
- [9] *Bondarev A.E., Galaktionov V.A.* Analysis of Space-Time Structures Appearance for Non-Stationary CFD Problems // Proceedings of 15-th International Conference On Computational Science ICCS 2015 Reykjavik, Iceland, Procedia Computer Science 2015. – Vol.51, No.1. – pp.1801-1810.

Алгоритм тональной компрессии на основе восприятия человека*

К.С. Зипа, А.В. Игнатенко

kzipa@graphics.cs.msu.ru | ignatenko@graphics.cs.msu.ru

Факультет Вычислительной Математики и Кибернетики

Московский государственный университет им. М.В.Ломоносова

Алгоритмы тональной компрессии преобразуют изображения широкого динамического диапазона к изображениям узкого диапазона, представимому на экране. Основные недостатки существующих локальных алгоритмов – это большое число не интуитивных параметров и привнесение дефектов. В данной статье предлагается модификация алгоритма тональной компрессии на основе глобальной оптимизации линейных окон (*Globally Optimized Linear Windowed Tone Mapping [1]*). Модификация основана на устройстве зрительной системы человека и позволяет улучшить результат алгоритма, а также заменяет не интуитивные параметры на несколько интуитивных, вариация которых в достаточно широком диапазоне не приводит к визуальному искажению изображения.

Ключевые слова: тональная компрессия, восприятие.

Tone mapping algorithm based on human perception*

K.S. Zipa, A.V. Ignatenko

Faculty of Computational Mathematics and Cybernetics

Lomonosov Moscow State University, Moscow, Russia

Tone-mapping algorithms transform high dynamic range images to low dynamic range images that can be displayed on screen. The main shortcomings of the existing local algorithms are a large number of non-intuitive parameters and bringing in artifacts. In this paper, we propose a modification of tone-mapping algorithm based on global optimization of linear windows. The modification is relied on human visual system functioning and allows to improve both global lightness and local contrast of algorithm's result. We also replaced non-intuitive settings by few intuitive. Variation of these parameters over a wide range does not lead to any visible distortions in output image.

Keywords: tone mapping, perception.

1. Введение

Динамический диапазон яркости окружающего мира очень широк, в процессе эволюции человеческий глаз научился воспринимать его большую часть. В то же время, диапазоны современных устройств вывода ограничены всего двумя-тремя порядками. Это ограничение привело к тому, что большинство получаемых изображений содержат в себе лишь узкий диапазон яркости, что может приводить к потере информации в некоторых областях изображения (см. Рис.1).

В то же время, с помощью обычной бытовой камеры [2] можно получить изображение широкого динамического диапазона (*high dynamic range image – HDR image*), то есть такое изображение, количество градаций яркости в котором превышает 256 значений. В противопоставление, изображение с меньшим количеством градаций называют изображением узкого динамического диапазона (*low dynamic range image – LDR image*).



Рис. 1: Пример фотографий разных выдержек, в которых потеряна детализация в различных частях сцены.

Чтобы отобразить HDR-изображение на экране монитора, нужно осуществить сжатие яркости. Такое преобразование называют алгоритмом тональной компрессии. Качество алгоритма тональной компрессии определяется степенью схожести результата на результат восприятия наблюдателем сцены. Поэтому алгоритмы тональной компрессии связаны с изучением восприятия человека.

Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347

2. Постановка задачи

Общая цель заключается в создании алгоритма тональной компрессии, обеспечивающего сохранение информации, существенной с точки зрения человеческого восприятия.

$$O = TM(I), \|O - I\|_{human} \rightarrow 0$$

$$I = \{(p, i) \in [0, M], M < \infty, p \in P\} - \text{HDR-изображение}$$

$$O = \{O(p, i) \in [0, 1], \dots, N\}, p \in P\} - \text{LDR-изображение}$$

$$P - \text{множество пикселей изображения}$$

$$i = \{R, G, B\} - \text{цветовой канал изображения}$$

Для решения этой задачи из всех существующих алгоритмов был выбран базовый алгоритм, который был модифицирован для повышения качества результата и уменьшения сложности использования.

3. Обзор существующих методов

Существует множество алгоритмов тональной компрессии, обзоры которых можно найти в работах [3] и [4]. Различают две большие группы методов – глобальные и локальные. Алгоритмы из первой группы преобразуют яркость каждого пикселя изображения единообразно, используя общую для всех пикселей кривую сжатия. Достоинство этих методов в простоте, высокой скорости. Недостаток же в том, что при достаточно большом диапазоне входных значений яркости начинает пропадать детализация в некоторых областях.

С целью предотвратить это и начали использоваться локальные операторы, которые модифицировали значение яркости уже исходя из некоторой окрестности пикселя. Методы этого класса значительно сложнее, но позволяют обеспечивать лучшее визуальное качество результата. Однако общая проблема локальных методов состоит в привнесении искусственных искажений в результат. Иногда эти искажения могут быть настолько выраженными и значительными, что это сводит на нет их преимущество.

Для сохранения достоинств и устранения недостатков был предложен метод [1]. В отличие от других локальных методов алгоритм он не требует декомпозиции или сегментации входного изображения, что позволяет избежать артефактов. Обработка карты излучательности производится напрямую. Благодаря процессу глобальной оптимизации, обеспечивается локальная согласованность в каждом пикселе изображения.

Основные недостатки алгоритма – это, во-первых, необоснованный выбор цветового пространства для построения карты излучательности, что приводит к необходимости использовать эвристические способы нормировки, а во-вторых – необходимость

спецификации нескольких параметров, уникальных для каждой сцены.

Специфичность параметров можно проиллюстрировать на следующем примере (Рис. 2). Вверху дано изображение, полученное на стандартных параметрах, рекомендованных в статье ($[\beta_1 = 0.6, \beta_2 = 0.2, \beta_3 = 0.1]$), а внизу – изображение, полученное при индивидуальном наборе параметров ($[\beta_1 = 1.3, \beta_2 = 0.0, \beta_3 = 0.1]$). Видно, что использование стандартных параметров в данном случае привело к значительной яркостной деформации области неба.

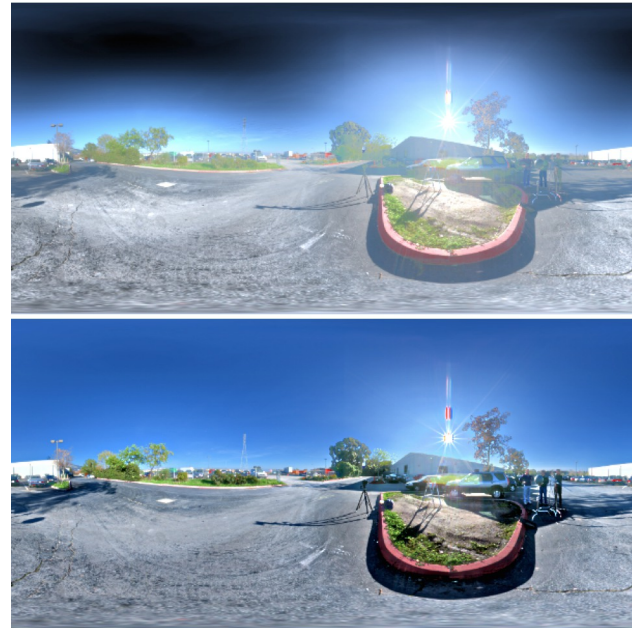


Рис. 2: Результаты работы базового метода (вверху – параметры $[0.6, 0.2, 0.1]$, внизу – параметры $[1.3, 0.0, 0.1]$).

4. Предложенный алгоритм

Перейдем к описанию алгоритма. Вначале будет дана общая схема алгоритма, представленная в статье [1] и указаны блоки, которые предлагается видоизменить. Затем будут более подробно разъяснены произведенные модификации.

4.1 Общая схема алгоритма GOLWTM

На вход алгоритму подается изображение широкого динамического диапазона в формате RGB. На выходе алгоритм возвращает изображение узкого динамического диапазона в формате RGB. Основные шаги алгоритма следующие:

1. Построение карты излучательности Q . В каждом пикселе карты излучательности хранится информация о яркости для этого пикселя. Карта излучательности получается эвристическим

алгоритмом, использующим цветовое пространство HSV и приводящим итоговую среднюю яркость к фиксированному значению.

- Генерация карты рекомендации C по карте излучательности Q . Качество этой карты имеет решающее значение в качестве итогового алгоритма тональной компрессии. Авторы предлагают рассчитывать её по следующей формуле:

$$C(p) = (\mu(p)^{\beta_1} \sigma(p)^{\beta_2} Q(p)^{\beta_3} + k)^{-1}$$

p – пиксель изображения

Q – карта излучательности

C – карта рекомендации

μ – локальное среднее (в окне 3×3)

σ – локальное стандартное отклонение (в окне 3×3)

$\beta_1, \beta_2, \beta_3, k$ – параметры алгоритма

- Всё пространство изображения разбивается на множество пересекающихся окон размером 3×3 , то есть каждый пиксель одновременно принадлежит 25 окнам. Внутри каждого локального окна предполагается наличие линейного преобразования яркости пикселя. Таким образом, требуется лишь найти параметры этого преобразования. Для этого предлагается в каждой точке минимизировать функционал разности:

$$f(q) = \sum_{p \in w(q)} (I(p) - u(p)Q - v(p))^2 + \varepsilon C^{-2} (u(p) - v(p))^2$$

q – рассматриваемый пиксель изображения

$w(q)$ – множество окон, в которое входит q

p – пиксель изображения, входящий в окно

I – выход этапа (излучательность узкого диапазона)

Q – карта излучательности

C – карта рекомендации

u, v – параметры линейного преобразования

ε – коэффициент регуляризации, позволяющий избежать тривиального решения

Минимизация функционала происходит путем рассмотрения системы линейных алгебраических уравнений и её решения. Детали этого процесса можно найти в статье [1].

Отметим, что из этого функционала виден смысл карты рекомендаций. Она показывает, насколько сильно можно сжать карту излучательности Q . Идея авторов статьи состояла в сжатии сильных перепадов яркости с сохранением детализации изображения, что обеспечивалось особенным построением карты, описанным в предыдущем пункте.

- Преобразование исходного HDR изображения в формате RGB по формуле Шлика [5]:

$$Output(p, i) = \left(\frac{Input(p, i)}{Q} \right)^s I$$

p – пиксель изображения

i – индекс цветового канала

I – излучательность узкого диапазона

Q – карта излучательности

Input – входные данные (HDR)

Output – выходные данные (LDR)

s – степень насыщенности (задается пользователем).

После этого выходное изображение нормализуется по гистограмме с отсечением верхних и нижних 0.5%, а затем семплируется на 256 яркостных уровней.

Для преодоления недостатков предлагается заменить в алгоритме шаги (1) и (2) и (4). Шаги (1) и (4) комплиментарны и определяются взаимосвязью между цветовыми и яркостными значениями в выбранной цветовой модели. Они будут рассмотрены в следующем пункте 4.2. Модификация шага (2) является ключевой модификацией в предложенном методе, поскольку позволяет сместить фокус карты рекомендации с локальной четкости (обеспеченной статистическими величинами внутри малого окна) в сторону объема передаваемой информации, связанной с человеческим восприятием. Описание шага 2 представлено в пункте 4.3.

4.2 Выбор цветового пространства

Для расчета яркости пикселя в базовой статье предлагалось использовать компоненту V из цветового пространства HSV. Хотя это цветовое пространство достаточно широко используется в силу своей интуитивности, оно не обладает такими важными чертами, как, скажем, линейность к человеческому зрению. Кроме того, оно не отражает адаптивность восприятия и другие важные феномены зрительной системы. Предлагается использовать другое стандартизированное цветовое пространство – CIECAM02 [6]. Это самая последняя модель, одобренная CIE. В настоящее время, она является компромиссом между сложностью настройки и точностью передачи процессов, происходящих в зрительной системе. CIECAM02 позволяет вычислить 6 атрибутов цвета: яркость, светлоту, цветность (colorfulness), хроматическую насыщенность (chroma), яркостную насыщенность (saturation) и оттенок. Именно яркость в этой модели связана с восприятием отражающей способности объектов. И именно её логично использовать в качестве Q в алгоритме. Основные шаги расчета величин исходя из цвета в XYZ следующие:

- Конвертация цвета в CAT02 LMS. При этом увеличивается спектральная четкость цвета.
- Применяется хроматическая адаптация на основе информации об окружении и фоне.
- Происходит конвертация в пространство LMS, близкое к откликами колбочек.
- Сжатие откликов колбочек.

Тонкий момент в применении модели к задаче тональной компрессии заключается в необходимости обеспечить корректное восстановление хро-

матических компонент после осуществления преобразования над яркостью (шаг 4 базового алгоритма). В исходном алгоритме это осуществлялось благодаря формуле Шлика [5]. Однако, эта формула неверно работает для выбранной цветовой модели. Вместо неё существует другой, однозначный способ восстановления. Если вынести в совокупные коэффициенты сложные параметры, то яркость рассчитывается по формуле:

$$Q = k * (aL + bM + cS)^\alpha,$$

где L, M, S – результат сжатия откликов колбочек. Предположим, что яркость преобразовалась в новое состояние Q' . Мы можем увидеть, что новые значения для колбочек можно рассчитать по следующим формулам:

$$L' = L * \left(\frac{I}{Q}\right)^\alpha, M' = M * \left(\frac{I}{Q}\right)^\alpha, S' = S * \left(\frac{I}{Q}\right)^\alpha$$

Здесь коэффициент α определяется исходя из параметров модели и не требует специального задания пользователем. Затем, произведем дополнительное выравнивание полученных откликов колбочек относительно исходных с помощью уравнивания значений яркости по гистограмме. Уравнивание проводится по отсечкам 0.5% относительно верхнего и нижнего пределов яркости в обоих изображениях. Это делается для того, чтобы избежать возможных ошибок в результате получения слишком больших значений после применения нелинейного коэффициента. После этого необходимо выполнить обратные преобразования хроматических компонент LMS в компоненты RGB. После этого нужно применить нормализацию по гистограмме, семплирование и гамма-коррекцию.

4.3 Расчет воспринимаемой яркости

Предлагается рассчитывать воспринимаемую яркость по модели BIWaM (Brightness Induction Wavelet Model) [7]. Затем эту воспринимаемую яркость можно использовать в качестве обратной карты рекомендаций C .

Модель описывает эффекты яркостного контраста, основываясь на известных свойствах зрительной системы. При применении модели моделируются многие иллюзии восприятия яркости. На Рис.3 приведены примеры иллюзий контраста. На примере слева иллюзия заключается в том, что серый квадрат выглядит более светлым на темном фоне и более темным на светлом. На примере справа серый ромб контрастирует с фоном, а не с примыкающей к нему решеткой.

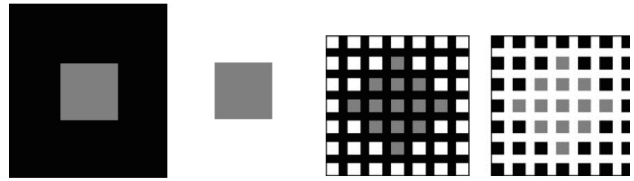


Рис. 3: Примеры иллюзий контраста.

В отличие от других аналогов, её применение не приносит сильных искажений в итоговое изображение, что делает её применимой в рассматриваемой задаче. У модели 2 параметра – базовая выраженность контраста и целевой масштаб, отражающий условия просмотра изображения. Каждый из этих параметров интуитивно понятен.

Эта модель осуществляет комбинацию яркости и контраста, самостоятельно устанавливая соотношения между ними на основе функции контрастной чувствительности. Модель использует декомпозицию изображения на набор частотно-ориентированных компонент.

Алгоритм, реализующий эту модель, принимает на вход одноканальное изображение, лежащее в диапазоне $[0,1]$. На выходе он возвращает другое изображение, вообще говоря, не ограниченное этим диапазоном. Поэтому для применения алгоритма необходимо нормировать полученную на первом шаге карту излучательности, а после применения алгоритма отрицательные значения можно ограничить небольшим положительным порогом, чтобы избежать деления на 0.

$$C = \max\left(\varepsilon, \text{BIWaM}\left(\frac{Q}{\max(Q)}\right)\right)^{-1}$$

Q – карта излучательности

$\varepsilon = 0.03$ – небольшой коэффициент

Внутри себя алгоритм BIWaM работает следующим образом:

1. Декомпозиция входного изображения Q на набор $3 * N + 1$ частотных компонент, где $N = \log_2(\max(W, H))$ – максимально доступный уровень детализации (W, H – ширина и высота изображения Q). На всех уровнях, кроме последнего присутствует по 3 разно-ориентированных компоненты, по смыслу отражающие контраст изображения. А на последнем уровне единственная компонента отражает общую яркость изображения.
2. На каждом уровне детализации (кроме последнего), для каждой ориентации, производится коррекция коэффициентов с использованием компоненты и двух нормирующих кривых:

$$w_{out}(s, o, p) = w_{in}(s, o, p) * (CSF_1(s) * Z_{s,o}(p) + CSF_2(s))$$

w_{in}, w_{out} – вход и выход шага соответственно
 s, o – текущий масштаб и ориентация

p – пиксель изображения

$Z_{s,o}(p)$ – контраст для текущей компоненты

$CSF_1(s), CSF_2(s)$ – кривые взвешивания для текущего масштаба

У кривых можно регулировать положение их пика (связан с целевым масштабом просмотра) и преобладание кривой 1 над кривой 2.

Контраст для текущей компоненты рассчитывается по формуле:

$$Z_{s,o}(p) = \frac{\sigma_{cen}^2(p)}{\sigma_{cen}^2(p) + \sigma_{sur}^2(p)},$$

p – пиксель изображения

$Z_{s,o}(p)$ – контраст для текущей компоненты

σ_{cen}^2 – дисперсия в окне 5×5

σ_{sur}^2 – дисперсия в окне 13×13 , из которого удалена сердцевина размером 5×5

3. Модифицированные компоненты собираются в результирующее изображение.

Использование алгоритма VIWaM позволяет сохранить на изображении детали, важные с точки зрения человеческого восприятия, и сжать детали, не важные с точки зрения человеческого восприятия.

5. Результаты алгоритма

Благодаря произведенным модификациям удалось улучшить качество алгоритма. На Рис.4 видно, что значительно повысилась детализация в области маски и платья. На Рис.5 видно, что исходный алгоритм потерял большую часть детализации в темных областях, а предложенный сохранил.



Рис. 4: Пример результатов исходного (слева) и модифицированного (справа) алгоритмов.



Рис. 5: Пример результатов исходного (слева) и модифицированного (справа) алгоритмов.

На Рис.6 витраж сохранил одновременно и высокую детализацию, и яркость относительно окружающих объектов. На Рис.7, наконец, показан результат на проблемном изображении, приведенном в начале статьи. Видно, что кроме отсутствия искажений, цвет на изображении стал более естественным.



Рис. 6: Пример результатов исходного (слева) и модифицированного (справа) алгоритмов.



Рис. 7: Пример результата модифицированного алгоритма.

6. Заключение

В статье предложена модификация алгоритма тональной компрессии на основе глобальной оптимизации линейных окон. Использование модели восприятия яркости позволяет отказаться от трудоемкой настройки параметров для каждого изображения. Модифицированный алгоритм демонстрирует улучшение с точки зрения количества различных деталей.

Литература

- [1] Shan Q., Jia J., Brown M.S. Globally Optimized Linear Windowed Tone Mapping. IEEE Transactions On Visualization And Computer Graphics, Vol. 16, No. 4, July/August 2010.
- [2] Debevec P., Jitendra Malik. Recovering high dynamic range radiance maps from photographs. // Proceeding SIGGRAPH '08 ACM SIGGRAPH 2008 classes Article No. 31. 2008.

- [3] Devlin K. A Review of Tone Reproduction Techniques, // Technical Report CSTR-02-005, Dept. of Computer Science, Univ. of Bristol. 2002.
- [4] Reinhard E., Ward G., Pattanaik S., and Debevec P., High Dynamic Range Imaging. Morgan Kaufman, 2005.
- [5] Schlick C. A Customizable Reflectance Model for Everyday Rendering, //Proc. Fourth Eurographics Workshop Rendering, pp. 73-83, 1993.
- [6] Moroney N. The CIECAM02 color appearance model. // Color and Imaging Conference, 10th Color and Imaging Conference Final Program and Proceedings, pp. 23-27(5)
- [7] Otazu X., Vanrell M., Párraga A.C. Multiresolution wavelet framework models brightness induction effects. // Vision Res. 2008 Feb; 48(5):733-51.

Об авторах

Кристина Зипа – аспирант Московского Государственного Университета, Факультет Вычислительной Математики и Кибернетики. Её контактные данные kzipa@graphics.cs.msu.ru

Алексей Игнатенко – к. ф.-м. н., с.н.с. Лаборатории Компьютерной Графики и Мультимедиа, Факультет Вычислительной Математики и Кибернетики. Его контактные данные ignatenko@graphics.cs.msu.ru

Комбинированный алгоритм локализации номеров железнодорожных цистерн

И.В. Копылов

ivv.kopylov@gmail.com

Кафедра математического и программного обеспечения ЭВМ (МПОЭВМ),
Череповецкий государственный университет, Череповец, Россия

Предложен новый подход к локализации номеров железнодорожных цистерн. Представлено гибридное решение, основанное на объединении алгоритма выделения границ контрастных объектов и алгоритма обнаружения номера с помощью обученного детектора, используя принцип скользящего окна. Первый этап возвращает набор зон на изображении, в которых предположительно может находиться номер вагона. На втором этапе происходит уточнение зоны, в которой находится номер.

Ключевые слова: локализация номеров, машинное обучение, скользящее окно, выделение границ контрастных объектов.

A combined algorithm for plate localization of railroad tank cars

I. V. Kopylov

Department of Mathematical Insure and Software of Computing Machines,
Cherepovets State University, Cherepovets, Russia

This paper shows a new approach to plate localization of railway tanks. Presented hybrid solution based on a combination of extraction algorithm boundaries contrast objects and plate detection algorithm by trained detector, using the principle of the sliding window. The first stage returns a set of areas in the image, which can be expected to be a plate of the tank car. The second step is the specification of a zone containing the plate.

Key words: plates localization, machine learning, sliding window, extraction boundaries contrast objects.

1. Введение

Задача локализации номеров цистерн в железнодорожном составе является актуальной для систем диагностирования подвижного состава, а также систем, предназначенных для автоматизации процессов коммерческого учета сырья и продукции, отгружаемых железнодорожным транспортом.

В данной статье предложен гибридный подход для локализации номеров вагонов, состоящий из двух этапов. На первом этапе выполняется выделение вертикальных перепадов яркости на изображении с дальнейшей обработкой полученной матрицы перепадов:

- выделение вертикальных перепадов с помощью фильтра Собеля [1], получение изображения границ G ;
- вычисление модулей перепадов яркости $|G|$ полученного изображения вертикальных перепадов G ;
- линейное контрастирование $|G|$ с удалением малоинформативных «хвостов» гистограммы;
- разбиение контрастированного изображения $|G|$ на столбцы с заданной шириной;
- выявление локальных минимумов проекции модуля градиента в столбцах;
- выбор потенциальных зон в каждом столбце;
- удаление пересекающихся зон;
- выбор лучших зон среди оставшихся после удаления пересекающихся областей;

- подавление областей, центрирование и формирование выхода с координатами зон с предполагаемым номером вагона.

На втором этапе используется обученный детектор (бинарный классификатор) номера. В найденных зонах производится попытка обнаружить номер с помощью детектора, используя принцип скользящего окна [2]. Каждое окно анализируется бинарным классификатором, который принимает решение - попадает часть номера в анализируемую область или нет. В результате выигрывает та зона, в которой было обнаружено большее количество положительных откликов детектора.

Алгоритм машинного обучения, используемый для обучения детектора, основан теории случайных функций [3,7]. В качестве признаков, которыми описываются зоны анализа детектора, были выбраны пирамидальные признаки [5].

В рамках эксперимента были подготовлены небольшая выборка для обучения детектора, состоящая из 500 положительных примеров (части номеров вагонов) и 500 шумовых примеров, а также тестовая выборка, состоящая из 4600 положительных примеров и 30000 шумовых. Обученный детектор показал следующие результаты на тестовой выборке: 89.93% правильного обнаружения номера, 4.44% ложного срабатывания на шумовой выборке.

Для проведения теста локализации номеров была составлена выборка из 78 изображений вагонов.

Эксперименты показали следующие результаты: на 76 вагонах номер был обнаружен, на 71 вагоне номер находился в первой из предложенных зон. Составленные выборки представлены в открытый доступ [8].

2. Локализация номера цистерны алгоритмом выделения границ

Для предварительной локализации зон с предполагаемым номером используется свертка исходного полутонового изображения I цистерны (рис. 1) с фильтром H , $I * H$. Результатом данной операции является изображение границ объектов G , над которым выполняется ряд операций, приводящих к нахождению зон с вероятным номером в них.

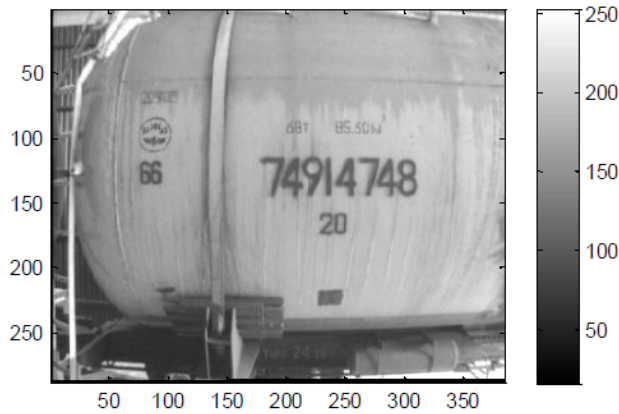


Рис. 1: Исходное изображение цистерны.

2.1. Вычисление перепадов яркости и линейное контрастирование

Выполняется выделение вертикальных перепадов яркости на изображении I с помощью фильтра Собеля с маской:

$$H = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}$$

В результате получается изображение границ $G = I * H$. Для дальнейшего анализа рассматриваются только абсолютные значения перепадов яркостей (рис. 2) в G , т.е. $|G|$ [1].

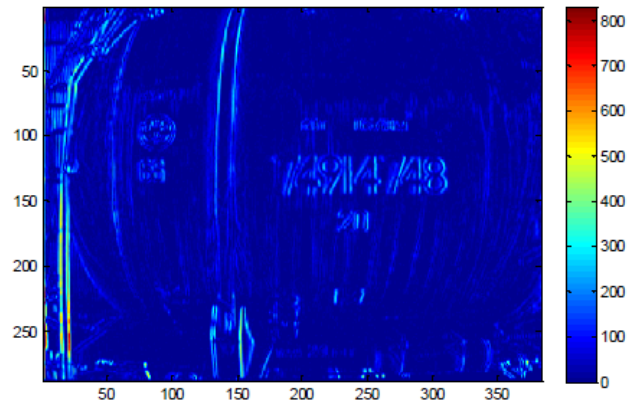


Рис. 2: Модуль вертикальных перепадов $|G|$.

Далее производится линейное контрастирование $|G|$. Для этого строится гистограмма по значениям из $|G|$, задаются доли самых больших и самых малых значений $|G|$ - $minGTh$, $maxGTh$. Минимальное значение контрастированного $|G|$ (обозначим как \bar{G}) станет равным 0, максимальное 255.

2.2. Локальные минимумы проекции модуля градиента в столбцах

Полученное после контрастирования изображение \bar{G} разбивается по горизонтали на n частей и в каждой из них рассчитывается средняя яркость элементов строки. Затем определяются локальные минимумы средней яркости. Количество частей, на которые необходимо разбить изображение определяется исходя из максимальной ширины номера $maxPlWidth$ и параметра k , указывающего, на сколько частей необходимо разбить номер максимальной ширины. $maxPlWidth$ и k являются настраиваемыми параметрами алгоритма.

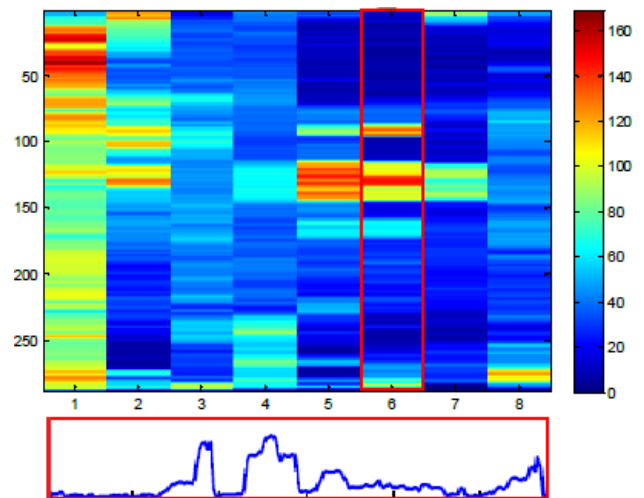


Рис. 3: Средняя интенсивность строк в каждой из зон. Профиль столбца 6.

При анализе средних яркостей b строк в каждом из столбцов, локальным минимумом считается строка i , если для нее выполняется одно из условий:

$$b(i) \leq b(i-1) \quad \text{and} \quad b(i) < b(i+1)$$

or

$$b(i) < b(i-1) \quad \text{and} \quad b(i) \leq b(i+1)$$

$(i, b(i))$ - локальный минимум.

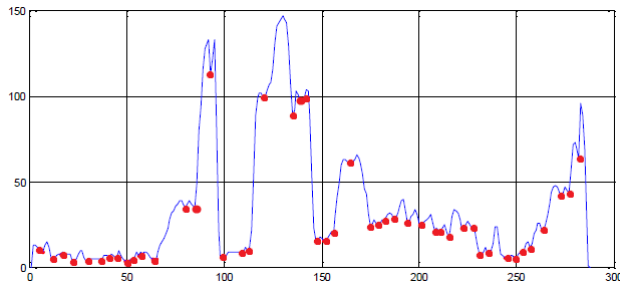


Рис. 4: Средняя интенсивность строк для столбца 6. Красными точками отмечены локальные минимумы.

2.3. Выбор потенциальных зон в каждом столбце

В каждом столбце анализируются всевозможные пары локальных минимумов с разницей координат от $minPlateHeight$ до $maxPlateHeight$, где $minPlateHeight$ - минимальная высота номера, $maxPlateHeight$ - максимальная высота номера. Через точки, соответствующие паре локальных минимумов, проводится прямая и выполняется оценка площади s (рис.5) между кривой средней яркости b и этой прямой. Площадь нормируется. Выбираются только те пары точек, у которых соответствующая нормированная площадь (вес) превышает порог $minS$.

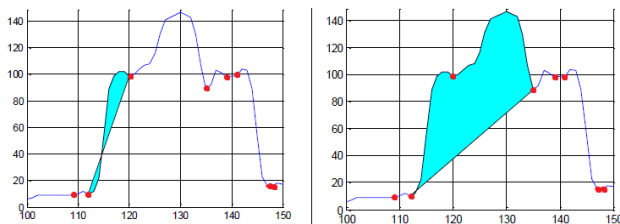


Рис. 5: Визуализация вычисляемых площадей.

Результатом выполнения этого этапа является набор потенциальных зон в каждом столбце. Для столбца 6 рассматриваемого примера было выделено 6 потенциальных зон (параметры $minPlateHeight = 15$, $maxPlateHeight = 32$).

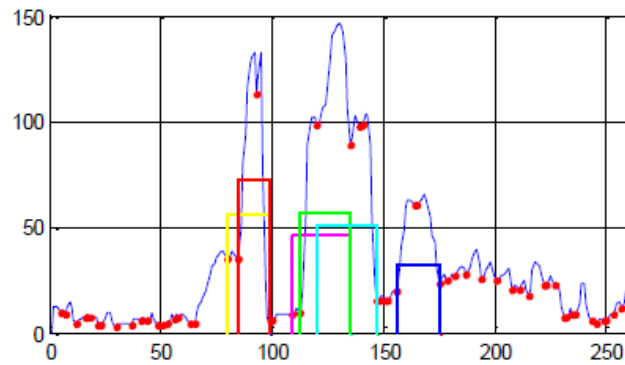


Рис. 6: Результат выделения потенциальных зон.

2.4. Удаление пересекающихся зон и выбор лучших среди оставшихся

Выделенная в каждом столбце зона проверяется на пересечение с другими зонами столбца, и если существует зона, которая пересекается с рассматриваемой и имеет большую площадь, то рассматриваемая зона удаляется (рис. 7).

Среди оставшихся зон выбирается заданное количество ($N_{kandidat}$) лучших в каждом столбце. Эти зоны по построению не могут пересекаться.

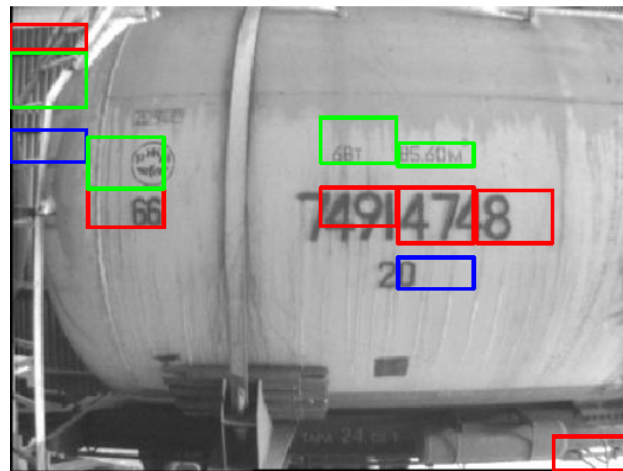


Рис. 7: В каждом столбце удалены пересекающиеся зоны и выделено по 3 зоны с максимальным весом среди оставшихся. Цвет прямоугольника соответствует порядковому номеру зоны: 1 - красный (зона с максимальным весом), 2 - зеленый, 3 - синий (зона с минимальным весом).

2.5. Формирование зон с номером

На данном этапе сравниваются выделенные зоны с зонами в соседних столбцах. Анализируется два соседних столбца слева и два соседних столбца справа. Если рассматриваемая зона пересекается по вертикали с зоной в одном из соседних столбцов и ее вес меньше соседа, то эта зона подавляется. Далее идет центрирование оставшихся обла-

стей, т.е. выбор предполагаемой средней зоны номера, при котором наибольший вес имеют оставшиеся зоны после процедуры подавления. У подавленных зон вес уменьшается. Опираясь на средние зоны, строятся зоны с предполагаемым номером цистерны в них (рис. 8). Высота зоны с номером равна высоте центрированной зоны столбца, ширина равна $maxPlWidth$.

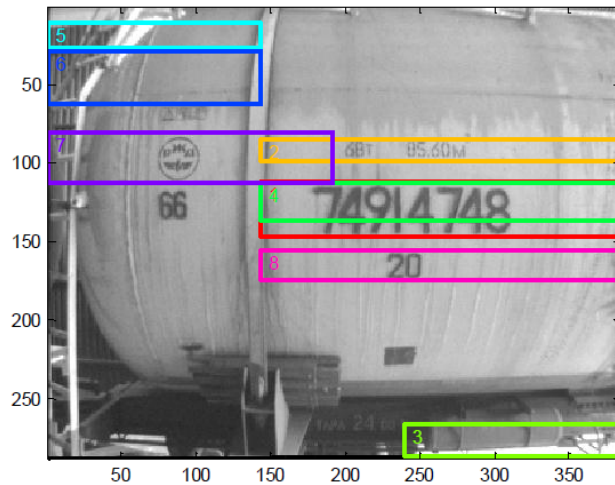


Рис. 8: Результат локализации алгоритмом выделения границ.

3. Уточнение зоны с номером

На данном этапе используется обученный детектор (бинарный классификатор) номера. В найденных зонах изображения I производится попытка обнаружить номер с помощью детектора, используя принцип скользящего окна. Каждое окно анализируется бинарным классификатором, который принимает решение - попадает часть номера в анализируемую область или нет. В результате выигрывает та зона, в которой было обнаружено большее количество положительных откликов детектора.

Алгоритм машинного обучения, используемый для обучения детектора, основан на теории случайных функций. В качестве признаков, которыми описываются зоны анализа детектора, были выбраны пирамидальные признаки [5].

3.1. Выбор признаков для изображения части номера

Рассмотрим этапы формирования признакового пространства для бинарного классификатора, более компактно описывающих изображения частей номеров цистерн [5]. Последовательность этапов формирования признаков выглядит следующим образом:

- Для представления изображения части номера цистерны выбран иерархический способ его

разбиения. Данный способ заключается в формировании пирамиды прямоугольных участков изображения A_0, A_1, \dots, A_n , интенсивности которых $I_{A_0}, I_{A_1}, \dots, I_{A_n}$. В итоге получается пирамида интенсивностей участков изображения, которая и является новым его представлением (рис. 9).

- Вид признаков, полученных из новой модели изображения, представляет собой всевозможные пары линейных комбинаций интенсивностей $I_{A_0}, I_{A_1}, \dots, I_{A_n}$:

$$f(I_{A_i}, I_{A_j}) = a_0 I_{A_i} + a_1 I_{A_j} \quad (1)$$

где $a_0 = 1/128$, $a_1 = -1/128$, а значения функции $f(I_{A_i}, I_{A_j}) \in (-2, 2)$.

- После преобразования изображения и получения признакового пространства выбирается подмножество наиболее информативных признаков, упрощающее задачу машинного обучения. Способ отбора значимых признаков основан на полном их переборе. В данном подходе используется критерий уменьшения признакового пространства с помощью линейного дискриминанта Фишера [6], определение значимости каждого признака с помощью критерия Джини [4]. Отбор производится жадным алгоритмом.

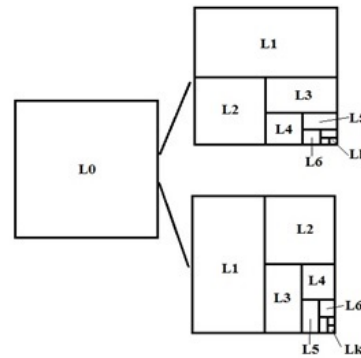


Рис. 9: Вид представления пирамиды участков изображения. L_0 – слой, которому соответствует само изображение, т.е. участок A_0 с интенсивностью I_{A_0} .

3.2. Обучение детектора номера цистерны

Для обучения детектора использовался алгоритм, основанный на теории случайных функций. Данный метод машинного обучения гарантирует получение оптимального результата с точки зрения рассматриваемого математического аппарата теории случайных функций. По эффективности не уступает иным популярным методам машинного обучения [3,7].

4. Экспериментальная часть

В рамках эксперимента были подготовлены небольшая выборка для обучения детектора, состоящая из 500 положительных примеров (части номеров вагонов) и 500 шумовых примеров, а также тестовая выборка, состоящая из 4600 положительных примеров и 30000 шумовых.

Каждое изображение представлялось в виде пирамиды с 5 уровнями. Всего возможных признаков (1) в данном представлении равно 820. Из них были отобраны 36 признаков. Т.е. каждому изображению из обучающей выборки соответствовал числовой вектор длиной 36. Обученный детектор показал следующие результаты на тестовой выборке: 89.93% правильного обнаружения номера, 4.44% ложного срабатывания на шумовой выборке.

Для проведения теста локализации номеров была составлена выборка из 78 изображений вагонов. Эксперименты показали следующие результаты: на 76 вагонах номер был обнаружен, на 71 вагоне номер находился в первой из предложенных зон (рис. 11). Составленные выборки представлены в открытом доступе [8].

Так, например, для рассматриваемого примера изображения с цистерной, которое использовалось в качестве иллюстраций в статье, распределение положительных откликов детектора на 8 зонах получилось следующим (рис. 10):

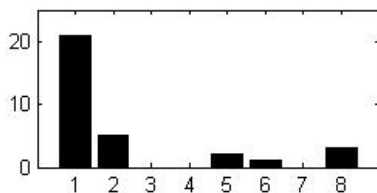


Рис. 10: В первой зоне детектор положительно сработал 21 раз, в результате чего алгоритм отдал предпочтение этой зоне.

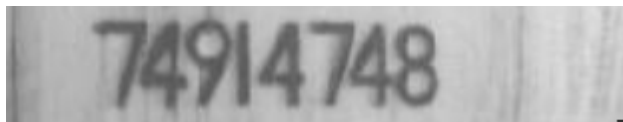


Рис. 11: Пример изображения первой зоны.

5. Заключение

Предложенный комбинированный алгоритм показал неплохие результаты на тестовых данных: 97% - номер будет в одной из найденных зон, 91% - номер будет содержаться в первой из найденных зон. Данный подход может показать лучшие результаты, если предоставить детектору номера больше дан-

ных для обучения, протестировать различные методы машинного обучения и иные способы признакового описания изображений частей номера цистерн.

Литература

- [1] *W. Kong, P. Chang, Z. Bi.* Real-time Sobel edge detector. // The 6th PSU-UNS International Conference on Engineering and Technology (ICET-2013), Novi Sad, May 15-17, 2013. University of Novi Sad, Faculty of Technical Sciences. pp.1-2.
- [2] *P. Viola, M. Jones.* Robust Real-Time Face Detection. // International Journal of Computer Vision 57(2), 2004, pp.137-154.
- [3] *Бахвалов Ю.Н., Малыгин Л.Л., Черкас П.С.* Метод машинного обучения на основе алгоритма многомерной интерполяции и аппроксимации случайных функций. // Череповец, Вестник Череповецкого государственного университета. 2012. Т. 2. № 39-2. С.7-10.
- [4] www.basegroup.ru/library/analysis/tree/math_cart_part1 – Критерий Джини.
- [5] *Копылов И.В., Малыгин Л.Л.* Построение модели изображения и ее применение в задачах детектирования объектов. // Череповец, Вестник Череповецкого государственного университета, Технические науки, №8, 2014.
- [6] *Дж.Ту, Р. Гонсалес.* Принципы распознавания образов. // Пер. с англ. – М.: Мир, 1974 –411 с. стр.281.
- [7] *Ширабакина Т.А., Бахвалов Ю.Н., Зуев А.Н.* Метод распознавания образов на основе теории случайных функций. // Известия высших учебных заведений. Приборостроение. СПГИ точной механики и оптики. 2005 г. Т.48, №2, С.5-8.
- [8] https://yadi.sk/d/_2yaEW59gx3gG – Расположение выборок для обучения и тестирования.

Сведения об авторе

Копылов Иван Владимирович.

Аспирант.

Дискретная математика и математическая кибернетика.

Кафедра математического и программного обеспечения ЭВМ.

Череповецкий Государственный Университет.

Kopylov Ivan Vladimirovich.

Postgraduate student.

01.01.09 Discrete mathematics and mathematical cybernetics.

Department of Mathematical Insure and Software of Computing Machines.

Cherepovets State University.

Телефон: +79114448038

E-mail: ivv.kopylov@gmail.com

Подсчет людей, используя данные, собранные с помощью сенсора Microsoft Kinect*

Я.А. Валуйская

yanavaluyskaya@mail.ru

Факультет вычислительной математики и кибернетики

Московский государственный университет им.М.В.Ломоносова

Проблема подсчета людей привлекает внимание многих исследователей из-за ее высокой практической актуальности и применимости в системах видеонаблюдения. Идея основного метода, предложенная в [6], основана на естественном предположении, что голова находится ближе к датчику Kinect Microsoft, чем плечи, когда датчик направлен вертикально вниз. В этой статье была предложена модификация базового метода: сегментация глубины, которая позволяет снизить ограничения основного метода на входные данные. Экспериментальная оценка предлагаемого метода показывает эффективность на данных, полученных под углом в плотном потоке, когда возникает проблема заслонения.

Ключевые слова: подсчет людей, глубина карта, сенсор Kinect Microsoft, алгоритм заполнения водой, сегментация глубины

People counting using data gathered via Microsoft Kinect sensor*

Iana Valuiskaia

Faculty of Computational Mathematics and Cybernetics

Lomonosov Moscow State University, Moscow, Russia

People counting captures the attention of many researches, because of its high practical relevance and applicability in the video surveillance systems. The idea of a basic method, proposed in [6], is based on the natural assumption that a head is closer to the Microsoft Kinect sensor than shoulders when sensor is directed vertically downward. In this paper, the modification of a basic method was proposed: depth segmentation, which allows to ease restrictions of the basic method on the input data. Experimental evaluation of the proposed method shows its efficiency on data shot at an angle in heavy traffic, when occlusion problem is raised.

Keywords: People counting, depth map, Microsoft Kinect sensor, Water Filling algorithm, depth segmentation

1. Introduction

People counting plays an important role in video surveillance systems. This area is in great commercial interest because of the high practical relevance and applicability of the results. Areas of application are very diverse and extensive. These systems are used on the entries and exits of the buildings, in the hallways and rooms, where a large crowd of people may be critical or dangerous.

Areas of application:

- Attendance monitoring
- Evaluation of tourists flow
- Market researches
- Security systems

Information about the number of people in conjunction with other information (for example, sales) allows to evaluate the effectiveness of promotions in stores. In addition, this information can be used to control the number of open cash desks and to plan staff breaks.

In security systems, which are used in public places (like airports, train stations, department stores, stadiums) people counting also is of a great importance. People counting can prevent dangerous situation or

help if an emergency has occurred. For example, people counting can be helpful in finding groups of people of abnormal size (this requires the collection of statistics on attendance and group behavior for quite a long period of time). In addition, in the case of evacuation knowledge about the number of people entered and left from the building is vital for the efficient rescue operation.

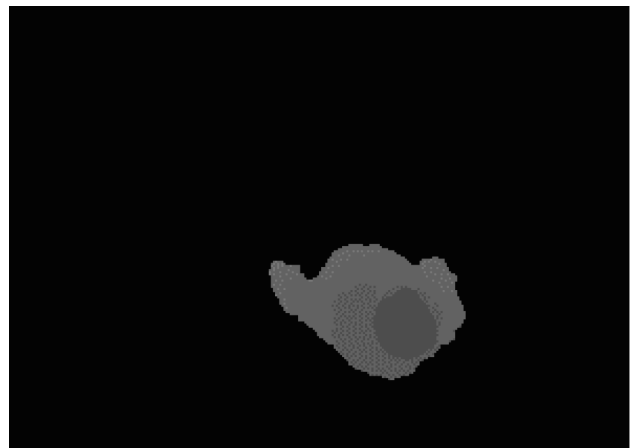


Figure 1: Data received from vertical Microsoft Kinect sensor

Работа выполнена и опубликована при финансовой поддержке РФФИ, гранты 14-01-00849, 15-07-20347

Different types of input data are used in people counting problem: data from temperature sensors, from infrared lasers, from rgb-cameras. Systems, based on video data, are the easiest to install and maintain. They create less problems to stuff and visitors, as opposed to, for example, mechanical turnstiles.

This paper describes a method that uses data, obtained from Microsoft Kinect sensor. Sensor is placed above people heads and directed either vertically downwards, as shown in Figure 1, or at angle to the vertical, Figure 2. When data is shot at an angle, occlusion problem is raised in heavy traffic.

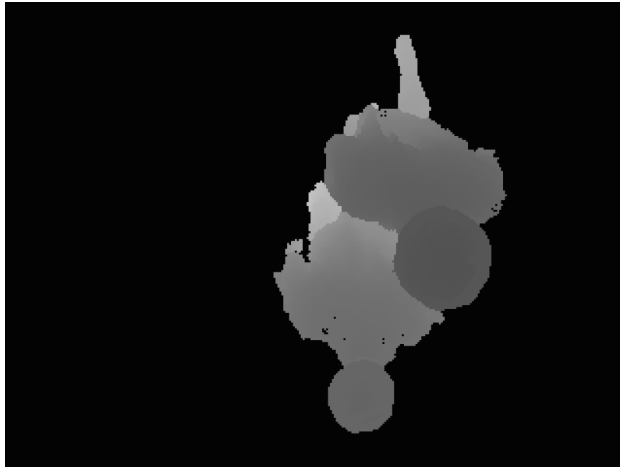


Figure 2: Data shot at an angle

Example of such system is shown in Figure 3. White horizontal lines mark region of interest. Depending on the direction of people movement, the number of people entered the ROI (IN) and the number of people left the ROI (OUT) is counted. People who haven't yet left the ROI are considered in counter STAY.

2. Review of existing methods

The main approaches for solving people counting problem are:

- Counting by people detection
- Counting by regression

In the first approach people detection problem is solved, for example, detection of peoples' heads. Then detected people are tracked by calculating movement trajectory. This approach is used in [1,4,6]. In the paper [1] authors used block-wise background subtraction and then k-means clustering is used to enable the segmentation of single persons in the scene. For tracking, they used "greedy algorithm". In the paper [4] authors extracted blobs and found peaks value for each one. Then Extended Kalman Filter is used. In the paper [6] head detection problem is reduced to finding suitable local minimums: Water Filling Algorithm is used. This approach demonstrates high accuracy when people flow is not too crowded. RGB-data and depth information are used as input.

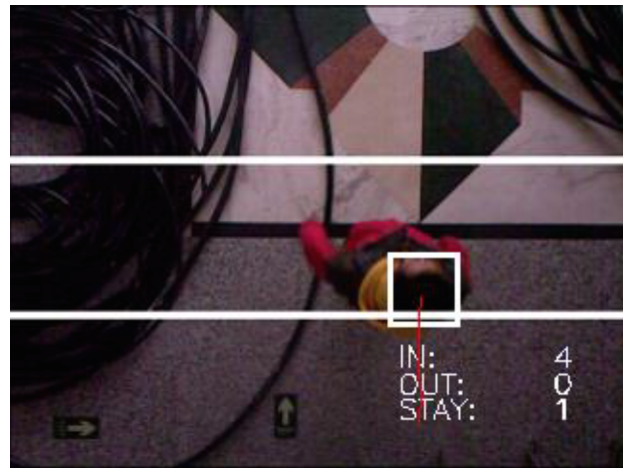


Figure 3: The result of people counting system

In the second approach some features of an input frame are extracted (for example, area, perimeter, HOG, etc). Sometimes, an input frame is segmented into several parts, for example, flows of people who go into different directions. In the paper [2] a set of simple holistic features is extracted from each segmented region, and the correspondence between features and the number of people per segment is learned with Gaussian Process regression. In the paper [5] the number of people is estimated in a set of overlapping sliding windows, using a regression function that maps from local features to a count. Then integer programming method is used.

This approach shows high results in crowded environment when people tracking is not possible. The major drawback of this approach is a necessity in a big amount of training data.

In addition, in paper [7] authors proposed a hybrid method. They use counting by detection if density of a flow is low. Otherwise, they analyze optical flow: segmentation by DBSCAN [3] algorithm is used. Accuracy of proposed method is 55% when evaluated on data with high density of a flow.

3. Proposed method

In this paper the method proposed in [6] is reviewed and modified. This basic method showed the best results among all the reviewed methods of first approach. In addition, it is capable of handling with tight groups of people (more than 3 people in a group). However, a major disadvantage of the basic method is a strict limitation on the input data: the sensor should be directed vertically downwards. If data is shot at an angle, occlusion problem appears. In this section, a modification of the basic method is proposed. It will ease the restrictions on the input data.

3.1 Basic method

The brief scheme of the basic method is shown in Figure 4.

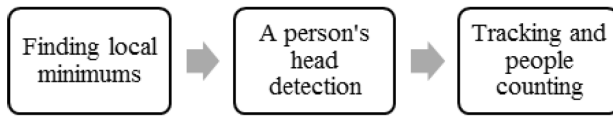


Figure 4: The brief scheme of the basic method

The input to the algorithm is a sequence of depth maps, obtained by Microsoft Kinect sensor. Depth map is an image, which contains information about the distance between objects and sensor's position. Vertical Microsoft Kinect sensor solves the occlusion problem naturally, that's way authors of the basic method use the fact that a person's head is a closest object to the sensor compared to some of it neighborhood. Thus, the problem of detecting peoples' heads is reduced to finding suitable local minimums of depth map.

3.1.1 Finding local minimums of the depth map

In order to find suitable local minimums of the depth map in [6] it was proposed to use Water Filling algorithm. Function

$$f(x, y) = \begin{cases} I_j(x, y), & (x, y) \in \text{Fore ground} \\ 0, & (x, y) \in \text{Back ground} \end{cases}$$

Additional function $g(x, y)$ "measures" function $f(x, y)$. It can be used to infer $f(x, y)$. The definition is below.

Definition: $g(x, y)$ is a measure function of $f(x, y)$ if and only if, $\exists \varepsilon > 0, \forall (x_1, y_1), (x_2, y_2)$, such as

$$\begin{aligned} & \| (x_1 - x_2)^2 + (y_1 - y_2)^2 \| < \varepsilon \quad \text{and} \\ & \text{if } f(x_1, y_1) \leq f(x_2, y_2) \quad \text{then} \\ & f(x_1, y_1) + g(x_1, y_1) \leq f(x_2, y_2) + g(x_2, y_2) \\ & \quad g(x_1, y_1) \geq g(x_2, y_2) \\ & \quad g(x_1, y_1) \geq 0, g(x_2, y_2) \geq 0 \end{aligned}$$

Finding proper measure function is about simulation rain process. Function $f(x, y)$ can be visualized as a land with hills and hollows. When it is raining, a raindrop in a hill will flow directly to the hollow under force of gravity. Measure function $g(x, y)$ reflects the number of water drops in a point with coordinates (x, y) .



Figure 5: 2D situation

Example is shown in Figure 5: green color is used for depth function and blue color for measure function.

Results of Water Filling algorithm is shown in Figure 6.

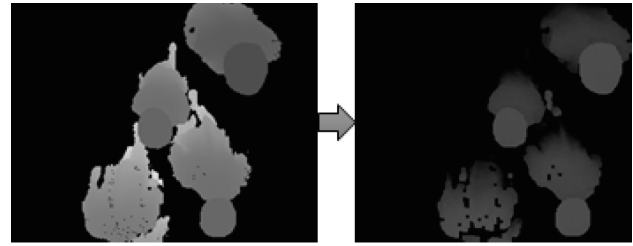


Figure 6: The result of Water Filling algorithm

3.1.2 Further analysis and tracking

At this stage, the resulting measure function $g(x, y)$ is analyzed to find regions that correspond to people's heads. This stage consists of several steps:

1. Thresholding of function $g(x, y)$
2. Region analysis

First step allows to get rid of noise regions, as well as to separate people who walk very close to each other, so only "shoulders-head" regions will remain.

On second step separated regions

Region $_i, i = 0, \dots, m$, in function $g(x, y)$ are calculated. It is known that the region "shoulders" is located below the region "head" of each person, so

$$g(x_h, y_h) > g(x_{sh}, y_{sh}), (x_h, y_h) \in \text{Head}_i, (x_{sh}, y_{sh}) \in \text{Shoulders}_i$$

Secondly, function $g(x, y)$ modified as follows:

$$g(x, y) = \begin{cases} g(x, y), & \text{wheng}(x, y) \geq M_i - \delta, (x, y) \in \text{Region}_i \\ 0, & \text{wheng}(x, y) \leq M_i - \delta, (x, y) \in \text{Region}_i \end{cases}$$

where $M_i = \max(g(x, y)), (x, y) \in \text{Region}_i, \delta > 0$ - parameter. A size of remaining regions is analyzed: too small regions (total area < 400 pixels) are removed.

After head detection it is possible to track people. Nearest Neighbor Tracking Algorithm is used for tracking people.

3.2 Modification of the basic method

If the sensor is directed at an angle to the vertical, then the first layer of people will be closer to the sensor than second. In case of occlusion problem, Water Filling Algorithm will not detect peoples' heads.

To solve this problem, in the preprocessing stage the input frame is splitted into segments according to depth. As a result, we receive groups of people located at the same distance from the sensor.

The depth segmentation consists of two steps:

1. Edge detection based on Canny edge detector
2. Identifying regions of people

First step allows to avoid situations when all foreground pixels will be allocated in one segment because of the occlusion problem.

Regions of people are identified by taking into account the edges obtained in step 1. The edges are not included in the resulting regions and used as delimiters.

Then, Water Filling Algorithm is applied not to the whole image, but separately to each segment found at the preprocessing stage. The result of depth segmentation is shown in Figure 7. Each segment is colored in its own color.



Figure 7: Depth segmentation result

4. Results

Implementation of the basic method and its modification were evaluated and compared with each other. Two datasets – Vertical and Angle – were used.

Vertical dataset consists of 3384 frames 320×240 . It was received from authors of the basic method [6]. This dataset was taken by Microsoft Kinect sensor, which was directed vertically downwards.

Angle dataset consists of 1102 frames 320×240 . It was taken from sensor, which was directed at an angle to the vertical. Both datasets were taken in natural environment. They consist of separated people and dense groups. In addition, people of different heights and sizes present in datasets.

Following metrics were used for evaluation and comparison of methods:

- Accuracy = $TP / (TP + FP)$, where TP – true-positive decisions, FP – false-positive decisions
- Recall = $TP / (TP + FN)$, where FN – false-negative decisions
- F-score = $2 * (Accuracy * Recall) / (Accuracy + Recall)$

Results are shown in Table 1. As the authors of basic method did not evaluate their algorithm on angle dataset, the comparison on angle dataset was carried out between the implementation of the basic method and modified algorithm.

Method	Vertical dataset			Angle dataset		
	Accuracy	Recall	F-score	Accuracy	Recall	F-score
Results of authors [6]	0.992	0.991	0.992	-	-	-
Basic method without modification	0.986	0.929	0.957	0.974	0.682	0.802
Modified method	0.986	0.929	0.957	0.985	0.779	0.87

Table 1: Experimental Evaluation

5. Conclusion

In this paper people counting method based on Water Filling Algorithm was reviewed. The analysis shows that its main disadvantage is a strict limitation on the input data: the efficiency of the basic algorithm decreases when a direction of Kinect sensor changes from vertical to angle. Modification of the basic method was proposed in order to ease this restriction.

Experimental evaluation of the modified algorithm showed its efficiency and improvement of the results compared with the implementation of the basic method.

Bibliography

- [1] Antic B. et al. K-means based segmentation for real-time zenithal people counting //Image Processing (ICIP), 2009 16th IEEE International Conference on. – IEEE, 2009. – pp.2565-2568.
- [2] Chan A. B., Liang Z. S. J., Vasconcelos N. Privacy preserving crowd monitoring: Counting people without people models or tracking //Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. – IEEE, 2008. – pp.1-7.
- [3] Ester M. et al. A density-based algorithm for discovering clusters in large spatial databases with noise //Kdd. – 1996. – V.96. – No. 34. – pp.226-231.
- [4] Hernandez D., Castrillon M., Lorenzo J. People counting with re-identification using depth cameras. – 2011.
- [5] Ma Z., Chan A. B. Crossing the line: Crowd counting by integer programming with local features //Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. – IEEE, 2013. – pp.2539-2546.
- [6] Zhang X. et al. Water filling: Unsupervised people counting via vertical kinect sensor //Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on. – IEEE, 2012. – pp.215-220.
- [7] Harebov P., Salimzibarov R. Zenithal people counting (In Russian) //The 22nd International Conference on Computer Graphics and Vision – 2012. – pp.158-162

Преобразования XYZ в спектр для свойств поверхности*

И.В. Валиев, А.Г. Волобой, Е.Ю. Денисов, С.В. Ершов, С.Г. Поздняков

piv@gin.keldysh.ru | voloboy@gin.keldysh.ru | eed@gin.keldysh.ru | measure@spp.keldysh.ru | ysp@gin.keldysh.ru

Институт прикладной математики им. М.В.Келдыша РАН, Москва

Описана процедура вычисления спектрограммы коэффициента отражения или пропускания, соответствующей заданной цветности в пространстве XYZ CIE 1931. Вычисления проводятся с помощью минимизации целевой функции, задающей отклонение заданных величин XYZ и XYZ для искомого спектрограммы.

Ключевые слова: XYZ Cie 1931, спектрограмма отражения.

Transformation of XYZ triplet into spectrum for surface properties*

I. V. Valiev, A. G. Voloboy, E. Yu. Denisov, S. V. Ershov, S. G. Pozdnyakov

The Keldysh Institute of Applied Mathematics Russian Academy of Science

The procedure of calculation of reflection or transmission spectrogram for given chromaticity XYZ CIE 1931 is presented. The calculations are carried out using the minimization of penalty function which is squared difference between required and current values of XYZ.

Keywords: XYZ Cie 1931, reflection spectrogram.

1. Введение

Типичной задачей компьютерной графики является моделирование распространения видимого света и построение изображения на этой основе. В общем случае полностью корректным может быть только спектральное моделирование. Зачастую полные спектрограммы распространяющегося света и (или) спектрограммы оптических атрибутов неизвестны и задаются в “усеченном” виде: либо в виде компонент RGB, либо в виде величин XYZ цветового пространства CIE XYZ. В связи с этим, часто возникает задача нахождения спектрограммы, соответствующей заданным величинам XYZ [1–4].

Преобразование спектрограммы в XYZ хорошо известно. Везде далее мы будем использовать только дискретное преобразование:

$$\begin{aligned} X &= \sum_{m=1}^M w_m^X s(\lambda_m) \equiv \sum_{m=1}^M w_m^X s_m \\ Y &= \sum_{m=1}^M w_m^Y s(\lambda_m) \equiv \sum_{m=1}^M w_m^Y s_m \\ Z &= \sum_{m=1}^M w_m^Z s(\lambda_m) \equiv \sum_{m=1}^M w_m^Z s_m \end{aligned} \quad (1)$$

Здесь w_m^X, w_m^Y, w_m^Z – известные коэффициенты преобразования CIE 1931 [1], λ_m – длина волны, $s_m \equiv s(\lambda_m)$ – спектрограмма. Ниже мы ограничимся стандартным преобразованием, использующим $M = 81$ длину волны: от 380nm до 780nm с шагом 5nm. Как правило, представления (1) вполне достаточно для целей компьютерной графики. При необходимости (1) можно применить и для непрерывных распределений, используя соответствующую интерполяцию.

Работа поддержана грантами РФФИ № 13-01-00454 и 15-01-01147, а также фирмой Integra Inc. (Япония). Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347.

В общем случае обратное преобразование — вычисление спектрограммы, соответствующей заданным величинам XYZ, может быть неединственным. Естественно, что вычисленная спектрограмма должна удовлетворять базовым физическим принципам. В случае зеркального или идеально диффузного (распределение Ламберта) отражения или пропускания света спектрограмма должна удовлетворять закону сохранения энергии (случаи переизлучения света с изменением длины волны здесь не рассматриваются):

$$0 \leq s_m \leq 1 \quad (2)$$

Следует особо отметить, что настоящий подход применим только к единственной спектрограмме, то есть или к спектрограмме одного вида отражения или к спектрограмме одного вида пропускания. В случае комбинации нескольких видов отражения и (или) пропускания соответствующие спектрограммы должны находиться совместно, чтобы результаты не нарушали закон сохранения энергии. Случаи диффузного отражения или пропускания, описываемые произвольными BDF, являются еще более сложными и в данной работе также не рассматриваются.

2. Предварительный анализ

В дальнейшем предполагается, что заданная тройка величин XYZ соответствует реальному цвету, то есть величины

$$x = \frac{X}{X+Y+Z}, y = \frac{Y}{X+Y+Z} \quad (3)$$

расположены внутри или на границе криволинейного “треугольника” цветностей CIE 1931 [1]. В противном случае считается, что тройка XYZ тем или

иным образом заменяется на “реальную”. Кроме этого, будем считать “реальной” тройку $X = Y = Z = 0$, соответствующую “нулевому” спектру, т.е. реальному цвету бесконечно малой яркости. Рассмотрим функцию $\Delta(s_m)$ для заданной тройки XYZ

$$\Delta = \left(\sum_{m=1}^M w_m^X s_m - X \right)^2 + \left(\sum_{m=1}^M w_m^Y s_m - Y \right)^2 + \left(\sum_{m=1}^M w_m^Z s_m - Z \right)^2 \quad (4)$$

Нетрудно видеть, что (4) достигает своего абсолютного минимума, если выполнены уравнения (1).

Для коэффициентов преобразования w_m^X, w_m^Y, w_m^Z и спектрограммы s_m удобно ввести квантово-механические обозначения “бра-кет” Дирака:

$s_m \rightarrow |s\rangle$, $w_m^X \rightarrow |w^X\rangle$, $w_m^Y \rightarrow |w^Y\rangle \rightarrow |w\rangle$, $w_m^Z \rightarrow |w^Z\rangle$ и определив для них скалярное произведение как: $\langle w^X | s \rangle = \sum_{m=1}^M w_m^X s_m$ и аналогично в других случаях.

Уравнение для искомой спектрограммы можно также сформулировать в следующем виде:

$$\hat{A}|s\rangle = |f\rangle \quad (5)$$

Здесь $\hat{A} = |w^X\rangle\langle w^X| + |w^Y\rangle\langle w^Y| + |w^Z\rangle\langle w^Z|$ – “проекторный оператор”, $|f\rangle = X \cdot |w^X\rangle + Y \cdot |w^Y\rangle + Z \cdot |w^Z\rangle$. Выражения вида $|w^X\rangle\langle w^X|$ следует понимать как прямое произведение.

С формальной точки зрения \hat{A} есть просто симметричная матрица размерности $M \times M$. Ее нормированные собственные вектора образуют ортонормированный базис соответствующего M -мерного пространства.

Для дальнейшего полезно проанализировать собственные значения Λ и собственные векторы оператора \hat{A} .

Уравнение для собственных значений есть:

$$\begin{aligned} \hat{A}|\varphi\rangle &= \Lambda \cdot |\varphi\rangle \\ \langle w^X \varphi | \cdot |w^X\rangle + \langle w^Y \varphi | \cdot |w^Y\rangle + \langle w^Z \varphi | \cdot |w^Z\rangle &= \Lambda \cdot |\varphi\rangle \end{aligned} \quad (6)$$

Из (6) следует, что при $\Lambda \neq 0$ собственные векторы \hat{A} представляются в виде

$$|\varphi\rangle = \alpha_X \cdot |w^X\rangle + \alpha_Y \cdot |w^Y\rangle + \alpha_Z \cdot |w^Z\rangle \quad (7)$$

а тройка коэффициентов $\alpha_X, \alpha_Y, \alpha_Z$ удовлетворяет следующему уравнению:

$$\begin{aligned} (W^{XX}\alpha_X + W^{XY}\alpha_Y + W^{XZ}\alpha_Z) \cdot |w^X\rangle \\ + (W^{YX}\alpha_X + W^{YY}\alpha_Y + W^{YZ}\alpha_Z) \cdot |w^Y\rangle \\ + (W^{ZX}\alpha_X + W^{ZY}\alpha_Y + W^{ZZ}\alpha_Z) \cdot |w^Z\rangle \\ = \Lambda\alpha_X|w^X\rangle + \Lambda\alpha_Y|w^Y\rangle + \Lambda\alpha_Z|w^Z\rangle \end{aligned} \quad (8)$$

Здесь $W^{XX} \equiv \langle w^X | w^X \rangle$, $W^{XY} \equiv \langle w^X | w^Y \rangle$, $W^{XZ} \equiv \langle w^X | w^Z \rangle \dots$

“Векторы” $|w^X\rangle, |w^Y\rangle, |w^Z\rangle$ линейно независимы, уравнение (8) преобразуется в задачу на собственные значения и собственные векторы для матрицы размерности 3×3 :

$$\begin{pmatrix} W^{XX} & W^{XY} & W^{XZ} \\ W^{YX} & W^{YY} & W^{YZ} \\ W^{ZX} & W^{ZY} & W^{ZZ} \end{pmatrix} \cdot \begin{pmatrix} \alpha_X \\ \alpha_Y \\ \alpha_Z \end{pmatrix} = \Lambda \begin{pmatrix} \alpha_X \\ \alpha_Y \\ \alpha_Z \end{pmatrix} \quad (9)$$

Матрица в уравнении (9) симметричная и невырожденная, она имеет три ненулевых собственные значения $\Lambda_1, \Lambda_2, \Lambda_3$ (они также являются тремя ненулевыми собственными значениями (6)) и соответствующие им тройки координат собственных векторов $\alpha_X^k, \alpha_Y^k, \alpha_Z^k, k = 1, 2, 3$. Собственные векторы (6), соответствующие $\Lambda_1, \Lambda_2, \Lambda_3$ находятся с помощью (7):

$$|\varphi_k\rangle = \alpha_X^k \cdot |w^X\rangle + \alpha_Y^k \cdot |w^Y\rangle + \alpha_Z^k \cdot |w^Z\rangle \quad (10)$$

Все остальные собственные значения оператора \hat{A} равны нулю. Далее считается, что полный ортонормированный базис $|\varphi_k\rangle, k = 1, 2, \dots, M$ оператора \hat{A} вычислен. Векторы $|s\rangle$ и $|f\rangle$ могут быть разложены по этому базису. При этом разложение $|f\rangle$ содержит только три слагаемых:

$$|s\rangle = \sum_{k=1}^M \alpha_k |\varphi_k\rangle, |f\rangle = b_1 |\varphi_1\rangle + b_2 |\varphi_2\rangle + b_3 |\varphi_3\rangle \quad (11)$$

После несложных преобразований с помощью (5) получаем:

$$|s\rangle = X \sum_{k=1}^3 \alpha_X^k |\varphi_k\rangle + Y \sum_{k=1}^3 \alpha_Y^k |\varphi_k\rangle + Z \sum_{k=1}^3 \alpha_Z^k |\varphi_k\rangle + \sum_{k=4}^M \alpha_k |\varphi_k\rangle \quad (12)$$

Вычисление всех величин, входящих в (12), кроме коэффициентов $\alpha_k, k \geq 4$, тривиально и быстро выполняется численно. Первые три слагаемые (12) дают спектрограмму, соответствующую заданной тройке XYZ , однако при этом условие (2) может не выполняться. Суть проблемы состоит в вычислении $\alpha_k, k \geq 4$, за счет которых должны быть выполнены ограничения (2).

3. Область допустимых величин XYZ

Прежде чем описать способ вычисления коэффициентов $\alpha_k, k \geq 4$, необходимо проанализировать область допустимых величин XYZ , точнее ее границу, при ограничениях (2), налагаемых на искомую спектрограмму.

Заметим, что допустимая область является выпуклой. Пусть две спектрограммы $|s^1\rangle$ и $|s^2\rangle$ соответствуют допустимым тройкам XYZ_1 и XYZ_2 и удовлетворяют условиям (2). В этом случае все семейство спектрограмм $t \cdot |s^1\rangle + (1-t) \cdot |s^2\rangle, 0 \leq t \leq 1$ также удовлетворяет (2), а соответствующее семейство

троек $t \cdot XYZ_1 + (1 - t) \cdot XYZ_2, 0 \leq t \leq 1$ является допустимым.

Анализировать границу допустимой области удобно используя “естественное” трехмерное Декартово пространство, оси которого соответствуют величинам, образующим тройки XYZ. Сразу заметим, что вся допустимая область представляет собой наклоненную “призму”. Нижнее и верхнее основания призмы есть одинаковые плоские выпуклые многоугольники, параллельные плоскости $Z = 0$, а ее боковая поверхность состоит из фасеток-параллелограммов, см. Рис. 1.

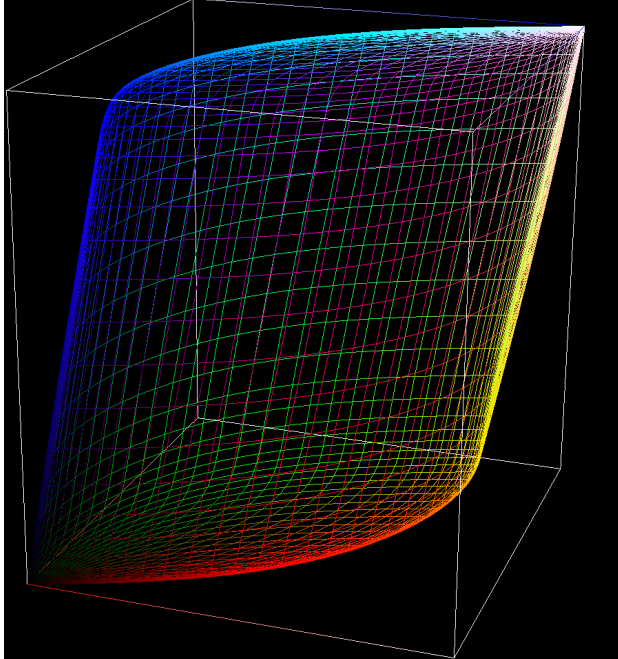


Рис. 1: Граница области допустимых XYZ.

В трехмерном пространстве компоненте спектрограммы s_m соответствует трехмерный вектор $\mathbf{w}_\lambda \equiv \mathbf{w}_m = \{w_m^X, w_m^Y, w_m^Z\}$. Векторы \mathbf{w}_m для пары $\lambda = 730nm$ и $\lambda = 750nm$ параллельны друг другу.

Также параллельны друг другу векторы для тройки $\lambda = 755nm$, $\lambda = 765nm$ и $\lambda = 775nm$. При дальнейшем анализе следует модифицировать множество векторов \mathbf{w}_m следующим образом:

$$\begin{aligned} \mathbf{w}_\lambda^* &= \mathbf{w}_\lambda, \lambda \neq 730, 750, 755, 765, 775nm \\ \mathbf{w}_{730}^* &= \mathbf{w}_{730} + \mathbf{w}_{750} \\ \mathbf{w}_{755}^* &= \mathbf{w}_{755} + \mathbf{w}_{765} + \mathbf{w}_{775} \\ \mathbf{w}_{750}^*, \mathbf{w}_{765}^*, \mathbf{w}_{775}^* &\text{ отсутствуют} \end{aligned}$$

Модифицированный набор содержит 78 векторов \mathbf{w}_m^* , в нем отсутствуют векторы для следующих длин волн: 750, 765 и 775nm. Коэффициенты спектрограмм s_m^* для модифицированного множества векторов удовлетворяют соотношению, аналогичному (2):

$$0 \leq s_m^* \leq 1 \quad (2')$$

Элементарный анализ показывает, что области допустимых значений XYZ для исходного \mathbf{w}_m и модифицированного \mathbf{w}_m^* наборов векторов при выполнении условий (2) и (2') полностью совпадают. С помощью непосредственных вычислений можно убедиться, что среди векторов модифицированного набора \mathbf{w}_m^* нет ни одной пары параллельных векторов.

Векторы \mathbf{w}_λ при $\lambda \geq 650nm$ имеют нулевую Z компоненту.

Поэтому допустимые тройки $XYZ(Z = 0)$, соответствующие спектру, содержащим только длины волн $\lambda \geq 650nm$, образуют “основание” всей допустимой области.

Преобразование (1) линейно, а область допустимых троек $XYZ(Z = 0)$ является выпуклой, поэтому границей основания будет ломаная линия, состоящая из отрезков прямых линий, образуемых векторами \mathbf{w}_m^* для $\lambda \geq 650nm$.

Нетрудно видеть, что последовательность векторов \mathbf{w}_m^* , образующих границу основания, диктуется углами между векторами и осью абсцисс. Ближняя к оси абсцисс часть границы состоит из векторов в порядке возрастания этого угла, дальняя — в порядке убывания. Практически очевидно, что верхнее основание допустимой области, соответствующее

$$Z_{max} = \sum_{k=1}^{81} w_k^Z = \sum_{k=1}^{78} w_k^{*Z}$$

может быть получено простым параллельным переносом нижнего основания на вектор, равный $\sum_m \mathbf{w}_m, \lambda_m \leq 645nm$.

Чтобы проанализировать структуру боковой поверхности допустимой области, выберем два вектора \mathbf{w}_m^* и \mathbf{w}_k^* . При этом хотя бы у одного из них должна быть ненулевая Z компонента. Пусть \mathbf{n} есть нормаль к плоскости, образованной \mathbf{w}_m^* и \mathbf{w}_k^* . Для определенности выберем направление нормали, для которого $n_Z > 0$. Возможность такого определения нормали для всех пар \mathbf{w}_m^* и \mathbf{w}_k^* легко проверить с помощью непосредственных вычислений. Рассмотрим два семейства спектрограмм $|s^{*1}\rangle$ и $|s^{*2}\rangle$, удовлетворяющих (2'):

$$\begin{aligned} s_j^{*1} &= \begin{cases} 1, (\mathbf{n} \cdot \mathbf{w}_j^*) > 0, j \neq m, k \\ 0, (\mathbf{n} \cdot \mathbf{w}_j^*) < 0, j \neq m, k \end{cases} & s_j^{*2} &= 1 - s_j^{*1}, j \neq m, k \\ & & (13) & \\ 0 \leq s_m^{*1}, s_k^{*1} &\leq 1 & 0 \leq s_m^{*2}, s_k^{*2} &\leq 1 \end{aligned}$$

Компоненты m и k обеих спектрограмм есть произвольные числа на отрезке от 0 до 1.

Спектрограммы (13) в пространстве XYZ соответствуют двум параллельным параллелограммам. Простой анализ показывает, что все допустимые тройки XYZ находятся в плоскостях, проходящих

через эти два параллелограмма, или лежат между этими плоскостями. Таким образом, параллелограммы, соответствующие спектрограммам (13) являются частью боковой поверхности допустимой области.

Можно показать, что вся боковая поверхность состоит из 5454 параллелограммов, соответствующих 2727 возможных пар \mathbf{w}_m^* и \mathbf{w}_k^* , у которых хотя бы один из векторов имеет ненулевую Z компоненту.

4. Вычисление спектрограммы

Вначале проверяем, лежит ли “входная” тройка XYZ внутри допустимой области. Если нет, то берём ближайшую к ней точку границы; её спектрограмма уже известна.

Если же XYZ лежит внутри области, начинаем вычисление спектрограммы $|s\rangle$. Алгоритм вычисления использует метод наискорейшего спуска. Стартував в некоторой точке внутри M -мерного гиперкуба, мы движемся против градиента Δ (4). Движение продолжается до тех пор, пока либо будет найдено удовлетворительное решение, либо траектория не пересечет какую-либо гиперплоскость. Если возникает пересечение, то траектория меняется: далее движение происходит внутри этой гиперплоскости (которая также есть гиперкуб меньшей размерности).

Как правило, такой итерационный процесс позволяет вычислить решение. Однако иногда сходимости нет. В этом случае процесс просто повторяется из другой стартовой точки. Наши численные эксперименты показали, что для большинства комбинаций XYZ 3–5 стартов более чем достаточно. А 100 попыток всегда дают необходимый результат.

При вычислениях удобно использовать сумму первых трех слагаемых (12), которая формально дает необходимый минимум (4), но соотношения (2) могут не выполняться:

$$|s^*\rangle = X \cdot \sum_{k=1}^3 \alpha_X^k \cdot |\varphi_k\rangle + Y \cdot \sum_{k=1}^3 \alpha_Y^k \cdot |\varphi_k\rangle + Z \cdot \sum_{k=1}^3 \alpha_Z^k \cdot |\varphi_k\rangle$$

Движение вдоль траектории происходит против направления градиента или его проекции на соответствующее гиперпространство,

$$\frac{\partial(|s\rangle - |s^*\rangle)}{\partial t} = -\hat{\mathbf{A}}(|s\rangle - |s^*\rangle) \quad (14)$$

Разложим (14) по базису $|\varphi\rangle$, $k = 1, 2, \dots, M$

$$|s\rangle - |s^*\rangle = \sum_m s_m(t) \cdot |\varphi_m\rangle, \quad \frac{\partial c_m}{\partial t} = \Lambda_m c_m \quad (15)$$

В случае гиперпространства с меньшей размерностью (15) приобретает вид:

$$\frac{\partial(|s\rangle - |s^*\rangle)}{\partial t} = -\hat{\mathbf{P}} \cdot \hat{\mathbf{A}}(|s\rangle - |s^*\rangle) \quad (16)$$

где $\hat{\mathbf{P}}$ – соответствующий проекционный оператор. Вектор $\hat{\mathbf{A}}(|s\rangle - |s^*\rangle)$ принадлежит подпространству, порожденному тройкой $|\varphi_1\rangle, |\varphi_2\rangle, |\varphi_3\rangle$, поэтому движение происходит в подпространстве, порожденном проекциями этой тройки: $\hat{\mathbf{P}} \cdot |\varphi_1\rangle, \hat{\mathbf{P}} \cdot |\varphi_2\rangle, \hat{\mathbf{P}} \cdot |\varphi_3\rangle$. Следовательно,

$$|s(t)\rangle = |s(0)\rangle + \sum_{k=1}^3 c_m(t) \cdot \hat{\mathbf{P}} \cdot |\varphi_m\rangle$$

Отсюда получаем:

$$\begin{aligned} \frac{\partial c_m}{\partial t} &= \Lambda_m [\langle s(t)|\varphi_m\rangle - \langle s^*|\varphi_m\rangle] = \\ &= -\Lambda_m [\langle s(0)|\varphi_m\rangle - \langle s^*|\varphi_m\rangle] - \\ &- \Lambda_m \sum_{k=1}^3 c_n(t) \langle \varphi_n | \hat{\mathbf{P}} | \varphi_m \rangle \end{aligned} \quad (17)$$

Уравнения (17) есть система трех обыкновенных линейных дифференциальных уравнений:

$$\frac{\partial \mathbf{c}}{\partial t} = \mathbf{f} + \hat{\mathbf{B}} \cdot \mathbf{c} \quad (18)$$

где трехмерный вектор \mathbf{f} и 3×3 матрица $\hat{\mathbf{B}}$ есть:

$$f_m \equiv -\Lambda_m [|s(0)\rangle - |s^*\rangle], B_{mn} = \langle \varphi_m | \hat{\mathbf{P}} | \varphi_n \rangle.$$

Пусть $\mathbf{c}^* = -\hat{\mathbf{B}}^{-1} \cdot \mathbf{f}$, $\beta_1, \beta_2, \beta_3$ есть собственные значения матрицы $\hat{\mathbf{B}}$, а соответствующие нормированные собственные вектора есть $\mathbf{q}^1, \mathbf{q}^2, \mathbf{q}^3$. Если вектор \mathbf{c}^* не дает искомую спектрограмму с необходимой точностью, тогда спуск продолжается:

$$\frac{\partial[\mathbf{c}(t) - \mathbf{c}^*]}{\partial t} = \hat{\mathbf{B}} \cdot [\mathbf{c}(t) - \mathbf{c}^*]$$

где $\mathbf{c}(t) - \mathbf{c}^* = p_1(t) \cdot \mathbf{q}^1 + p_2(t) \cdot \mathbf{q}^2 + p_3(t) \cdot \mathbf{q}^3$. Далее

$$\sum_{m=1}^3 \frac{\partial p_m}{\partial t} \mathbf{q}^m = \sum_{m=1}^3 p_m \hat{\mathbf{B}} \cdot \mathbf{q}^m = \sum_{m=1}^3 p_m \beta_m \mathbf{q}^m, \quad \frac{\partial p_m}{\partial t} = p_m \beta_m$$

В результате получаем:

$$p_m(t) = p_m(0) \exp(\beta_m t), \quad \mathbf{c}(t) = \mathbf{c}^* + \sum_{m=1}^3 p_m(0) \exp(\beta_m t) \mathbf{q}^m$$

Окончательно

$$\begin{aligned} |s(t)\rangle &= |s(0)\rangle + \sum_{m=1}^3 c_m(t) \cdot \hat{\mathbf{P}} \cdot |\varphi_m\rangle = \\ &= |s(0)\rangle + \sum_{m=1}^3 c_m^* \cdot \hat{\mathbf{P}} \cdot |\varphi_m\rangle + \\ &+ \sum_{m,n=1}^3 p_n(0) \cdot \exp(\beta_m t) q_m^n \cdot \hat{\mathbf{P}} | \varphi_m \rangle \end{aligned} \quad (19)$$

Далее численно определяется момент t пересечения траектории (19) с поверхностью гиперкуба, т.е. когда одна из компонент $|s(t)\rangle$ становится равной 0 или 1. Следующая итерация стартует из точки пересечения в пространстве меньшей размерности.

Процесс продолжается до тех пор пока, либо решение не будет найдено, либо решение уравнения

$\mathbf{f} + \hat{\mathbf{V}} \cdot \mathbf{c} = 0$ перестанет существовать, т.е. когда размерность подпространства станет менее 3.

Если решение не получено, то повторяем процедуру с другой начальной точкой, и т.д.

5. Заключение

Описанная процедура вычисления спектрограммы отражения является устойчивой, вычисления с помощью этой процедуры быстры и позволяют достичь высокой точности. Вместе с тем следует отметить один ее недостаток: спектрограммы для близких троек XYZ могут значительно различаться. Это вызвано как неоднозначностью решения, так и случайным выбором стартовой точки. Как правило, этот недостаток не играет большой роли, но в некоторых случаях его нельзя игнорировать. В дальнейшем предполагается изменить алгоритм,

чтобы устранить этот недостаток для большей части допустимой области.

Литература

- [1] http://en.wikipedia.org/wiki/CIE_1931_color_space.
- [2] Жданов Д.Д., Потемин И.С. Построение спектрального представления из RGB данных в задачах спектрального моделирования // Труды 20-ой международной конференции по компьютерной графике и зрению ГРАФИКОН-2010, 20-24 сентября 2010, Санкт-Петербург, Россия, С.144-147
- [3] Brian Smits. An RGB to Spectrum Conversion for Reflectances / University of Utah, January 21, 2000
- [4] Glassner A.S. How to derive a spectrum from an RGB triplet / IEEE Computer Graphics and Applications 9, 4 (July 1989), pp.95-99

Значение особенностей зрительного восприятия для интерпретации визуальных моделей произвольных данных *

А.А. Захарова, А.В. Шкляр

zaa@tpu.ru

Томск, Россия, Институт кибернетики, Томский политехнический университет

Работа посвящена вопросам исследования особенностей зрительного восприятия человека в качестве ключевого фактора, определяющего результативность визуального анализа информации, имеющей различное происхождение, степень сложности или тип. Предложено использовать понимание процесса зрительного восприятия в качестве управляющего параметра в формализованной процедуре построения визуальных моделей данных. Оправданное применение инструментов визуализации, являющееся следствием этого подхода, обеспечивает усиление когнитивного потенциала визуального анализа.

Ключевые слова: визуальная модель, зрительное восприятие, интерпретация.

The Value of Features of Visual Perception to Interpret Visual Patterns of Arbitrary Data*

A.A. Zakharova, A.V. Shklyar

Tomsk, Russia, Institute of Cybernetics, Tomsk Polytechnic University

Work is devoted to studies of the human visual perception as a key factor determining the effectiveness of visual analysis of information of different origins, the degree of difficulty or the type. It is proposed to use the understanding of the process of visual perception as a control parameter in a formalized procedure for the construction of visual data models. Justify the use of visualization tools, which is a consequence of this approach, provides amplification of the cognitive potential of visual analysis.

Keywords: visual model, visual perception, interpretation.

1. Введение

Визуальный анализ является эффективным, удобным, распространенным и привычным методом исследования экспериментальных, расчетных и даже абстрактных данных. Современные технические средства визуализации позволяют получать изображения высокой степени сложности, соответствующей, при необходимости, фотореалистичному восприятию. Эта возможность создает условия для построения визуальных моделей данных, интерпретация которых опирается на широкий спектр когнитивных функций, свойственных зрительному восприятию человека.

В качестве опорного понятия будем называть методом визуализации алгоритм сопоставления между некоторыми исходными данными и зрительно воспринимаемым объектом. В этом подходе, указанный объект назовем визуальной моделью, а построение алгоритма сопоставления – задачей визуализации [5].

Разнообразие существующих и появляющихся технических возможностей, используемых для создания визуальных моделей, делает актуальными обобщение и формализацию процедуры построения таких моделей при практическом использовании методов визуального анализа. Решение этой задачи позволит использовать существующие технологии

визуализации в качестве полноценного инструмента научных исследований, соответствующего уровню сложности современных задач.

2. Восприятие и интерпретация образов

Зрительное восприятие является сложным, неоднозначным и слабоизученным процессом [3]. При создании визуальных моделей любых данных решается, как минимум, одна из следующих задач: передача информации наблюдателю или поиск нового смысла в данных посредством анализа визуальной метафоры. В первом варианте возникает проблема точности интерпретации зрительного образа, а в другом – стимулирования когнитивных механизмов, участвующих в создании новой информации об изучаемой системе.

Решение этих задач происходит постоянно, анализ зрительной информации является естественной и непрерывной функцией мозга человека [2]. Однако, индивидуальные особенности восприятия, сформированные как результат собственного развития, не позволяют эффективно использовать этот активный потенциал в общем случае. Таким образом, формулируется задача поиска или построения системы инвариантных образов, которые могли бы стать основанием для построения предсказуемо интерпретируемых визуальных моделей.

Для задач визуализации, целью которых является получение совершенно новой для наблюдателя информации при помощи активных когнитивных про-

Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347

цессов, интерпретация, жестко предопределенная заранее, является осложняющим обстоятельством. Понимание присутствия этих разнонаправленных факторов при взаимодействии с визуальными образами приводит к актуальной задаче создания формализованной многопараметрической процедуры разработки визуальных моделей данных.

3. Визуальные высказывания

Визуализация как научный метод опирается на идею возможности представления информации о любом объекте или системе в виде визуального образа [6]. Разработка принципов унификации такого представления необходима для обеспечения быстрого и управляемого обмена информацией между людьми. Участие визуализации в информационном обмене, распространенность и ее активное использование приводят к необходимости создания инструментов, обеспечивающих визуализацию признаками и возможностями языковой системы. Вместе с формированием словаря «визуального языка» уже ведутся работы, нацеленные на определение принципов, способных обосновать синтаксис, лексику и более сложные аспекты «визуального языка».

Ближайшим шагом в этом направлении предлагается исследование закономерностей построения «визуальных высказываний», т.е. последовательностей или совокупностей зрительных образов, интерпретация которых имеет большее информативное содержание, чем каждый из образов по отдельности. Восприятие и интерпретация таких «высказываний» является менее изученным процессом, чем для статичных образов. Одним из перспективных графических инструментов, соответствующим этой задаче, определена компьютерная анимация. В свою очередь, привлечение анимации к решению информационно-аналитических задач создает условия для ее собственного развития в новом направлении.

Зрительное восприятие обладает совершенно особыми механизмами, функционирование которых происходит по правилам, определение и исследование которых является важной задачей. Обозначение границ применимости или эффективной результативности каждого конкретного метода визуализации позволит снизить ресурсоемкость визуального анализа, уменьшая время предварительного обоснования допустимости выбранной модели [2].

Наряду с традиционным визуальным сопоставлением, распознаванием или выделением при переходе к изучению визуального языка следует учитывать более сложные виды взаимодействия с информативным образом. Ассоциативное мышление, эмоциональный отклик или эстетические предпочтения могут оказывать существенное влияние на

когнитивное взаимодействие с визуальным образом информации, особенно, если это насыщенный поток данных. Изучение этих явлений нуждается как в теоретических, так и в экспериментальных исследованиях [9].

Зрительное восприятие человека функционирует таким образом, что позволяет анализировать и запоминать широкий поток информации непрерывно, с множеством деталей и особенностей, включая изменения состояния образа во времени. Это обстоятельство позволяет изучать визуальный язык, как инструмент, оперирующий параллельными высказываниями с высокой степенью информативности и интерпретируемости. Следовательно, возможна разработка множества новых визуальных моделей, отличных от традиционных, и при этом не уступающие им с точки зрения информационной содержательности.

4. Экспериментальное исследование

Понимание того, каким образом у каждого конкретного наблюдателя происходит формирование понимания смысла наблюдаемого им образа, на современном этапе развития науки остается слабоизученным и неоднозначным. Исследование этой проблемы в контексте применения инструментов визуализации для анализа произвольной информации имеет первостепенное значение. Выявление общих закономерностей восприятия, определение приемов, позволяющих эффективно реализовать индивидуальный когнитивный потенциал, правила построения визуальных моделей, обладающих предсказуемой интерпретируемостью у широкого круга наблюдателей – эти и многие другие вопросы являются важнейшими для развития визуального анализа и компьютерной графики.

Одним из важнейших приемов, участвующих в создании визуальных моделей, является метафора представления – набор выразительных средств, определяющих зрительный образ и помогающий зрителю понять смысл визуализируемой информации при помощи сопоставления особенностей исходной информации и свойств самого образа [7]. В настоящее время, авторами разрабатывается серия экспериментов, направленных на выявление степени интерпретируемости и когнитивной значимости различных метафор визуального представления. Первая задача, сформулированная в этом направлении, посвящена оценке и сравнению традиционных приемов визуализации информационной основы различной степени сложности. В числе таких приемов находятся двух- и трехмерные образы плотности значений, структурные схемы, изометрические поверхности, линии тока и т.д. [4]

Для следующего этапа исследований разрабатываются тестовые программы по определению применимости современных технологических решений в

области компьютерной графики для получения новых визуальных метафор. Основным направлением этого этапа предложено привлечение приемов интерактивного взаимодействия исследователя с виртуальной средой для их объединения с изучением визуальных образов информации.

Отдельными задачами рассматриваются исследование особенностей взаимодействия наблюдателя с динамически изменяющимися данными, определение времени интерпретации, скорости анализа и запоминания для таких моделей. Изучение этих процессов проводится для обоснования разрабатываемых выразительных средств «визуального языка».

4.1 Тестовый стенд

Построение визуальных моделей, обладающих предсказуемой интерпретируемостью, требует фактических данных, на основании анализа которых можно будет определить преимущество одного способа визуализации перед другим. Возникает задача измерения параметров такого сложного процесса, как понимание зрителем смысла наблюдаемого им визуального образа. С этой целью был разработан тестовый стенд, позволяющий получать численные характеристики величин, связанных с интерпретацией визуальной модели.

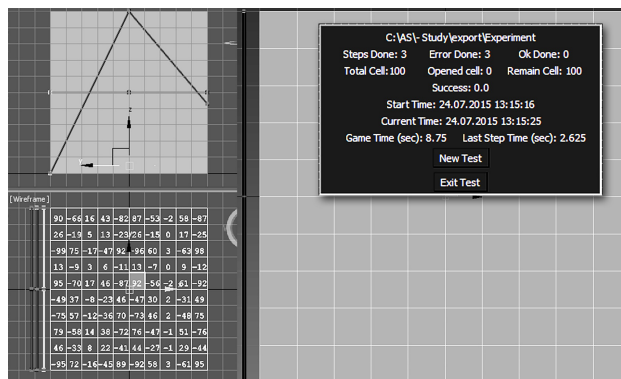


Рис. 1: Интерфейс тестового стенда

Тестовый стенд представляет собой интерактивную виртуальную среду, позволяющую наблюдателю взаимодействовать с исходной информацией в рамках ограничений, являющихся условиями проводимого эксперимента (рис. 1). Исходной (исследуемой) информацией является распределение в виртуальном пространстве стенда некоторых данных, например, значений скалярного или векторного поля. Для каждого проводимого эксперимента граничным (начальным) условием является предположение, что участнику эксперимента не известно ничего об этом распределении, кроме того факта, что существует определенная закономерность, которому оно подчинено. Человек, принимающий участие в отдельном эксперименте, является исследователем, задачей которого является выявление

возможных закономерностей в исходных данных на основании наблюдений за визуальной моделью.

Любой эксперимент, проводимый с помощью стенда, представляет собой тест, состоящий из последовательности дискретных шагов, обладающих фиксируемой результативностью. На каждом шаге наблюдатель высказывает предположение о значении исходной величины в выбранной им самой точке экспериментального пространства. Если сделанное предположение соответствует действительности, то исследователь получает этому подтверждение и значение считается определенным. Эта процедура повторяется до определения значений во всех точках или до достижения заданной степени безошибочности, определяемой как последовательность точных предположений. Одной из основных задач эксперимента является выявление момента, после которого наблюдатель сможет понять закономерность, которой подчинено исходное распределение.

Величинами, измеряемыми в ходе эксперимента, являются соотношение правильных предположений и их общего числа («интерпретируемость»), время достижения заданного уровня этого параметра, интервал между правильными предположениями («время принятия правильного решения»), изменение среднего времени на принятие решения как зависимость от объема уже известных данных и т.д. Информация о необходимых количественных характеристиках проходящего эксперимента постоянно присутствует в поле зрения наблюдателя. Кроме того, возникает возможность наблюдения за индивидуальными особенностями восприятия каждого участника, изменением скорости появления правильных или ошибочных решений («утомляемость»), и даже объяснения с помощью результатов наблюдений выбираемой стратегии исследования неизвестных данных [10].

Переход к изучению особенностей визуального восприятия происходит после дополнения описанной выше схемы стенда компонентами визуальных моделей. Замена или дополнение символического отображения значений исходной величины в дискретных точках пространства на соответствующий им зрительный образ позволяют сравнивать процессы анализа наблюдателем визуализированных данных. Указанные выше измеряемые характеристики интерактивного взаимодействия с пространством тестового стенда теперь могут рассматриваться как численные данные, описывающие свойства зрительного образа. Возникает возможность анализа и сравнения свойств визуальных моделей данных с точки зрения их эффективной интерпретируемости.

Таким образом, предлагается набор измеряемых параметров, описывающих свойства визуальных

моделей. Управление значениями этих параметров в заданном диапазоне делает визуальные модели контролируемым инструментом анализа данных. Это делает возможным введение классификационных признаков задач визуализации, опирающихся на измеряемые величины. Важной особенностью этих параметров является их зависимость как от постановки задачи визуализации, типа изучаемых данных, использованного способа визуализации, так и от особенностей восприятия наблюдателей. Таким образом, характеристики наблюдателя становятся неотъемлемой частью задачи визуализации.

4.2 Визуализация скалярного поля

При помощи описанного выше тестового стенда проведена серия экспериментов, целью которой являлись определение и сравнение свойств некоторых существующих визуальных моделей, примененных для анализа одного из типов данных. Разделение данных на типы (эмпирические, абстрактные, гипотетические и т.д.), с учетом предположения о возможности визуализации любого из них, имеет смысл с точки зрения выделения их особенностей. Специфика данных становится параметром задачи визуализации, поэтому представляет интерес исследование характеристик визуальных моделей для всех типов данных. Визуальные модели пространственного распределения скалярных величин широко и давно используются для их анализа. Во многом существующие приемы визуализации являются следствием технических возможностей того периода истории науки, к которому относится их появление. Современные возможности компьютерной графики приводят к разработке совершенно иных приемов. Их сравнительная эффективность также может быть изучена, чтобы определить преимущественную сферу применения [1].

В рассматриваемой задаче визуализации скалярного поля исходными данными, подвергаемыми визуальному анализу, является распределение числовых значений на плоскости. Распределение формируется как комбинация из нескольких элементарных функций, зависящих от пространственных координат, со случайными весовыми параметрами. В начале эксперимента исследователю ничего не известно о возможном виде распределения, кроме диапазона допустимых значений. Действия исследователя представляют собой пошаговое повторение алгоритма, состоящего из выбора ячейки тестового поля, получения конечного набора возможных значений исследуемой скалярной величины и выбора варианта, который, по мнению исследователя, наиболее близок к правильному. В эксперименте диапазон допустимых значений разбивался на десять равных участков, создавая таким образом большой выбор вариантов, но не увеличивая

время эксперимента сверх порога утомляемости его участника.

В случае, когда сделанное предположение оказывалось правильным, т.е. ближайшим к истинному значению изучаемой величины в данной точке, текущий шаг засчитывался как «достоверный» в статистике эксперимента. Возможность перехода к новой ячейке без необходимости получения правильного значения на предыдущей не ограничивалась. Подобная организация эксперимента должна имитировать процесс зрительного восприятия, когда человек изучает незнакомый ему объект. Получаемые таким образом данные о предпочтительной последовательности наблюдения за отдельными частями образа также дают информацию, которую следует учитывать при разработке визуальных моделей. Целью описываемого экспериментального исследования является выявление возможных закономерностей, влияющих на зрительное восприятие в общем случае. Также исследуются когнитивные процессы, приводящие к возникновению новой информации в мозге человека при анализе визуального образа или их последовательности.

Для определения влияния именно визуального анализа исследование любой визуальной модели содержало три серии экспериментов. Задачей тестов из первой серии было определение значимости собственных способностей исследователя для анализа неполных (фрагментированных) данных без участия зрительного восприятия. Для создания такого ограничения исследователю, сделавшему правильное предположение о значении исходного распределения в выбранной точке, лишь сообщалось об этом, а ячейка, в которой было верно определено значение, помечалась цветом. Таким образом, в анализе данных были задействованы память участника эксперимента, его аналитические способности и, вероятно, интуиция.

Во второй серии тестов к поиску закономерностей в исходных данных подключался анализ визуальных данных. Ограничением этой серии было представление информации, правильно определенной в ходе эксперимента лишь в символьном виде. Для этого ячейка не помечалась цветом, а заменялась на цифровое значение, соответствующее исходному распределению. В третьей серии экспериментов оценивалась эффективность использования визуальных моделей в сравнении с результатами первых двух серий. Цифровое значение, в этой постановке, дополнялось или заменялось фрагментом визуальной модели исходных данных. Вариант тестов, в котором одновременно визуализированы и символьная и графическая модель, предполагает ознакомление участника эксперимента с метафорой представления только в ходе теста.

4.3 Анализ экспериментальных данных

Для экспериментов из первой серии, в которых в постоянном распоряжении наблюдателя находится лишь информация о координатах правильных предположений, типичный результат определения исследователем собственного «интуитивного потенциала» выглядит так (рис. 2). Кривая, расположенная в правой части, показывает изменение значения «интерпретируемости» - соотношения правильных решений и общего числа шагов, в ходе эксперимента.

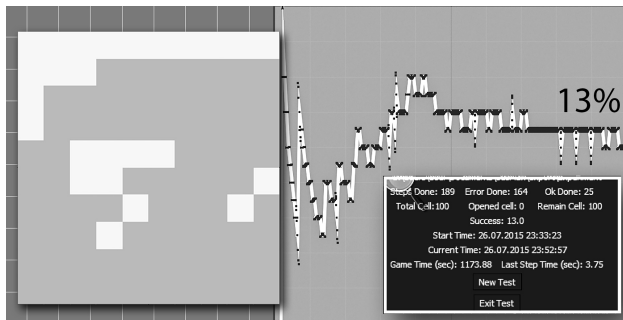


Рис. 2: Эксперимент первой серии (уровень интерпретируемости – 13%, информированность – 23%, время теста – 1170 сек.)

Эта ситуация рассматривается как «нулевой» уровень визуализации, т.е. принимается точка зрения, что визуальный анализ в ней не имеет никакого влияния на выдвигаемые предположения относительно исследуемого распределения. Для разных исходных данных и разных участников число правильных предположений колебалось в среднем от 12 до 25% общего числа принятых решений. Во всех остальных тестах, когда для наблюдения были доступны любые формы напоминания об определенных ранее значениях исходных данных, считалось, что результат определяется взаимодействием наблюдателя с визуальной моделью.

На рис.3 представлено состояние стенда при проведении эксперимента из второй серии. В этом случае правильно определенные данные постоянно доступны для наблюдения. Таким образом, считаем, что визуальной моделью являются именно цифровые обозначения, расположенные в соответствующих точках пространства. Задачей этой серии являлось изучение возможных изменений в процессе понимания закономерности в исследуемых данных. Причиной таких изменений могла быть только визуализация фрагментированных данных.

Проведенные измерения показали несколько отличий от результатов первой серии. Число правильно сделанных предположений с самого начала теста становится выше результатов первой серии. Спустя некоторое время этот параметр заметно возрастал. Это позволяло утверждать, что наблюдателем

определена закономерность, присутствующая в исходных данных. На этом этапе количество правильных предположений достигает 40-50%.

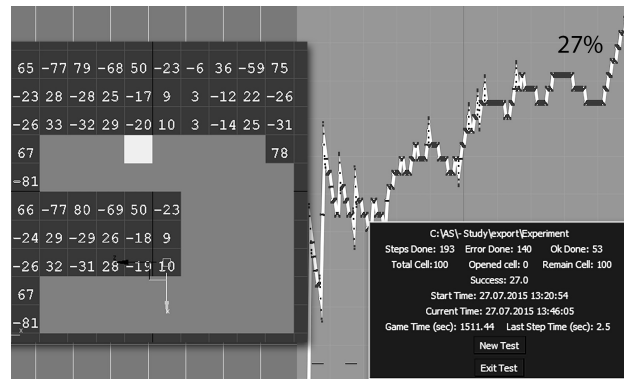


Рис. 3: Ход тестирования второй серии (уровень интерпретируемости – 27%, информированность – 62%, время теста – 1511 сек.)

При определении порогового значения уровня информированности (т.е. числа точно установленных значений из всех возможных), необходимого для определения вида изучаемой закономерности, было установлено, что это значение, в основном, оказывается в диапазоне 30-45% и зависит от индивидуальных качеств наблюдателя, а также от выбираемой им последовательности исследования тестового поля.

В одном из вариантов экспериментов третьей серии цифровые данные, появившиеся во второй серии, дополнялись картой цветového кодирования. Возможен также вариант, в котором цифровые данные убираются с тестового поля, однако, это приводит к необходимости введения в поле зрения шкалы цветového кодирования. В общем случае, наблюдателю предоставлялась возможность самостоятельно определения правил кодирования, которые и являются метафорой представления (рис.4).

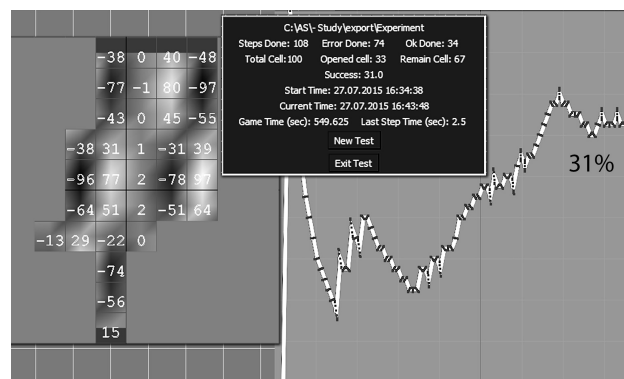


Рис. 4: Эксперимент с цветovým кодированием (уровень интерпретируемости – 31%, информированность – 33%, время теста – 549 сек.)

Для эксперимента их этой серии уровень правильных предположений на начальной стадии оказался сопоставим с показателями тестов из второй группы. Однако, время, необходимое для визуального поиска закономерностей, управляющих распределением, в большинстве случаев, оказывалось меньше на 15-20%. Отсутствие предварительной осведомленности о правилах использованного цветового кодирования замедляло принятия правильных решений лишь на начальном этапе (5-8% общего числа сделанных шагов).

Отдельный интерес при исследовании визуальных моделей данных вызывает изучение временных затрат, необходимых наблюдателю для принятия решения о смысле изучаемого зрительного образа (рис.5). Использование временного ресурса является важнейшим показателем эффективности визуальной модели. Как видно из полученных данных, время для принятия любого решения постепенно сокращается по мере прохождения теста.

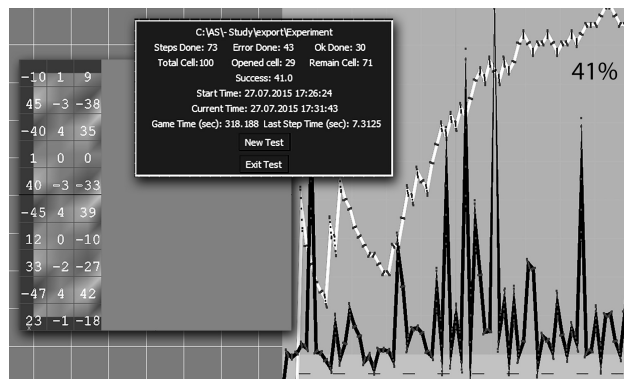


Рис. 5: Изменение времени принятия решения (уровень интерпретируемости – 41%, информированность – 29%, время теста – 318 сек.)

Как видно на рис.5 (темная линия), после некоторых ошибочных предположений возникает пауза, необходимая, видимо, для мысленного уточнения устанавливаемой закономерности. У многих участников эксперимента, длительная пауза перед принятием нового решения появлялась после сделанных «правильных предположений». Можно предположить, что это время затрачивается для выбора необходимого направления в исследовании.

Характеристики визуальных моделей, влияющие на временные интервалы, определяющие достижение наблюдателем необходимой степени «понимания», являются важными параметрами решения задачи визуализации. Понимание объективных ограничений создаваемого вида визуальных моделей, а также разработка способов, изменяющих его когнитивные свойства в необходимых пределах, способны сделать визуальный анализ инструмен-

том, применение которого будет уместным и результативным.

4.4 Описание визуальных моделей

Предлагаемая процедура тестирования позволяет измерять значения некоторых параметров визуальных моделей данных. Эта возможность позволяет говорить о системе унифицированных характеристик таких моделей. Для создания подобной системы необходимо провести обобщающее исследование методов визуального анализа, оценивая их эффективность применительно к типизированным исходным данным, либо к систематизированным целям визуального исследования.

В настоящий момент, авторами предлагается несколько характеристик, значимость которых считается высокой. В дальнейшем, этот набор будет дополнен величинами, являющимися признаками типов моделей и позволяющими разрабатывать их классификацию.

К характеристикам, изучение которых возможно при помощи описанного тестового стенда, относятся следующие:

- Скорость интерпретации - величина, обратная времени, необходимому наблюдателю для понимания смысла образа исследуемых данных с точностью, определяемой условиями задачи визуализации.
- Самодостаточность - готовность модели к интерпретации без предварительного или дополнительного ознакомления наблюдателя с метафорой представления.
- Совместимость - возможность одновременной интерпретации состояний нескольких величин в рамках одной визуальной модели.
- Степень инвариантности - уровень независимости результатов интерпретации от особенностей восприятия произвольного наблюдателя).

Успешность принятия решений при взаимодействии с визуальной моделью может быть обеспечена в том случае, если разработчиком учтены и реализованы способы взаимодействия с восприятием исследователя. Особые трудности возникают при выборе способа визуализации, если модель должна содержать в пространстве своего представления одновременно образы данных разного типа, изменяющиеся или недостоверные. Введение системы параметров, отражающих свойства визуальной модели как когнитивного инструмента, должно создать условия для их правильного применения при решении любых задач, связанных с обработкой и анализом как экспериментальных, так и расчетно-теоретических данных.

Практическими задачами, решение которых может быть упрощено или ускорено при обоснованном применении визуальных моделей, является анализ

и интерпретация точечных измерений, полученных в геологических исследованиях, экспериментальных физических измерениях величин, имеющих пространственное распределение (поля температур, давлений). Кроме того, при сопоставлении реальных данных и результатов интерпретации визуальных моделей появляется возможность выбора или уточнения метафоры представления, соответствующей специфике конкретной задачи. Во многих случаях, значимым фактором, определяющим результативность применения визуальной модели, становятся особенности восприятия, сформировавшиеся как характерная черта группы людей, участвующих в решении специфичной задачи анализа.

4.5 Последовательность восприятия

Проведенные эксперименты позволили получить представление о том, как протекает изучение распределенных в пространстве визуализированных данных у различных участников исследований. Целью этих наблюдений являлось получение первичных данных о динамических характеристиках процесса изучения незнакомого визуального образа. Понимание факторов, оказывающих влияние на этот процесс, должно стать необходимой составляющей разработки новых типов визуальных моделей.

Для значительной части экспериментов характерной особенностью являлось последовательное переключение внимания наблюдателя (рис. 6). Это свидетельствует о привлечении сведений в уже изученных точках к формированию предположений о значениях исследуемой величины в новых точках. Кроме того, последовательность шагов, включая выбор начальной точки, во многом аналогична чтению.

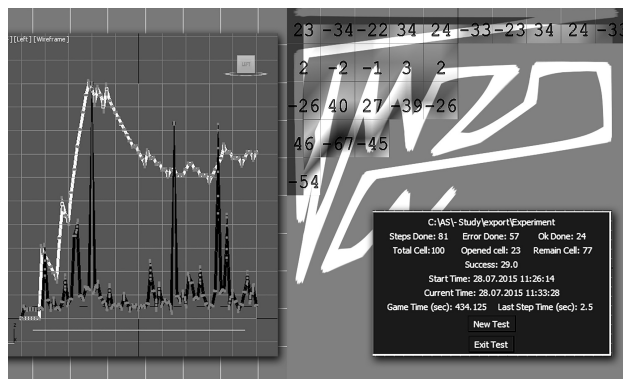


Рис. 6: Последовательность шагов, аналогичная чтению (уровень интерпретируемости – 29%, информированность – 23%, время теста – 434 сек.)

Однако, около трети участников предпочитали последовательность шагов, начинающуюся во внутренней точке тестового поля, а затем расширяю-

щуюся к его краям в направлениях, которые легче и быстрее поддавались анализу (рис. 7). Подобная последовательность, скорее, соответствует изучению именно визуальных образов, в т.ч. переключению внимания на особенности изображений, заинтересовавшие наблюдателя. Следует отметить, что интерпретируемость моделей у исследователей, придерживающихся этой стратегии, была стабильно выше на 5-10%.

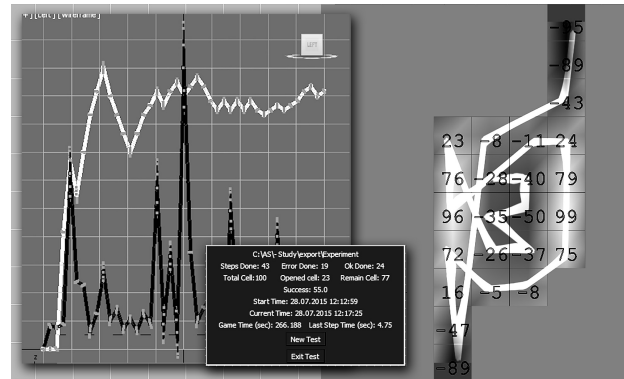


Рис. 7: Образная последовательность восприятия (уровень интерпретируемости – 55%, информированность – 23%, время теста – 266 сек.)

Самостоятельный выбор стратегии изучения модели данных каждым участником эксперимента способен приводить к различным выводам в тех случаях, когда речь идет о визуальном анализе сложных многокомпонентных моделей данных [8]. Это обстоятельство также необходимо учитывать при разработке информативно-насыщенной графики.

5. Формализация визуального анализа

Результатом изучения особенностей визуального анализа метафор представления информации должна стать классификация задач визуализации. Выделение признаков исходной исследовательской задачи, позволяющих быстро переходить к визуальному анализу обеспечит снижение временных затрат и позволит прогнозировать его результативность. Критериями такой классификации могут стать как особенности исходных данных, так и предпочитаемые способы их изучения.

Формализация процесса создания визуальных моделей для данных, имеющих широкое разнообразие по уровню сложности и происхождению, становится условием применимости визуального анализа для расширяющегося круга задач. Эта задача может рассматриваться как общенаучное исследование, т.к. находится на стыке множества наук и направлений, обеспечивая взаимодействие исследователей с большими объемами данных. Переход к использованию визуального языка, лексически единицами которого можно рассматривать зрительные образы, станет возможным только вместе

с достижением значимых результатов в определении принципов их интерпретируемости.

Разработка новых визуальных моделей является ресурсоемким процессом. Его эффективность зависит от правильного определения особенностей исходных данных, а также корректного понимания смысла задачи визуализации и точного представления о восприятии предлагаемого образа наблюдателем. Построение формального описания визуальных моделей данных в общем случае и его применение для перехода к решению конкретной задачи визуализации являются определяющими для обеспечения значимости визуальных методов в научных и прикладных исследованиях.

6. Заключение

Исследование особенностей восприятия человеком зрительной информации находится на ранней стадии исследования. Успешная реализация запланированных экспериментальных исследований в этом направлении позволит перейти к обоснованной разработке новых типов визуализации, использующих для своего функционирования последние достижения в области компьютерной графики. Обоснована необходимость разработки формальной процедуры постановки и решения задач визуализации. Это позволит использовать визуальный анализ для более широкого круга задач. Показана возможность перехода к управляемому использованию методов визуального анализа. Введена система количественных характеристик для визуальных моделей, обосновывающая их применимость. Предложена разработка классификации визуальных моделей, опирающаяся на измеряемые параметры. Объяснена необходимость учета и использования индивидуальных качеств исследователя в качестве параметра задачи визуализации. Значительная роль в эффективном использовании визуального анализа данных, по результатам проводимых исследований, отводится интерактивному взаимодействию с данными, а также применению психофизических методов управления.

Литература

- [1] Bolotova Yu.A., Druki A.A. и Spitsyn V.G License plate recognition with hierarchical temporal memory model [Конференция] // 9th International Forum on Strategic Technology (IFOST -2014). – Chittagong, October 21-23, 2014.: Chittagong: CUET, 2014. – pp.121-124, 2014.
- [2] Mazza R. Introduction to Information Visualization [Книга]. – London: Springer-Verlag Limited, 2009. – 149p.
- [3] Ware Colin. Information Visualization, Third Edition (3rd Edition) Perception for Design (Interactive Technologies) [Book]. – [s.l.]: Morgan Kaufmann, 2012. – 536p.
- [4] Бондарев В.А., Галактионов А.Е. Анализ развития концепций и методов визуального представления данных в задачах вычислительной физики [Статья] // Препринты ИПМ им. М.В.Келдыша. – 2009. №53. – 28с.
URL: <http://library.keldysh.ru/preprint.asp?id=2009-53>.
- [5] Захарова А.А., Шкляр А.В. Информативные признаки задач визуализации [Статья] // Научная визуализация. – 2015. – 2: Т. VII. – С.73-80.
- [6] Зиновьев А.А. Основы логической теории научных знаний [Книга]. – М.: Издательство ЛКИ, 2010. – Издание 2-е, испр. и доп. – 264с.
- [7] Никитина М.А. О когнитивном потенциале метафоры в научном тексте [Журнал] // Вестник ИГЛУ. – 2010. – №3.
- [8] Пирс Ч.С. Начала прагматизма. Том 2. Логические основания теории знаков [Книга]. – СПб.: Лаборатория метафизических исследований философского факультета СПбГУ; Алетейя, 2000. – 352с.
- [9] Соломоник Абрам. Философия знаковых систем и язык [Книга]. – Минск: МЕТ, 2002. – 404с.
- [10] Яковив И.Б. Взаимосвязь информации и знаков [Статья] // Information Technology and Security. – Киев: ИССЗИ НТУУ "КПИ 2012. – 1 (1). – С.31-41.

Авторы

Захарова Алена Александровна, Институт кибернетики, Томский политехнический университет, Томск, Россия,
zaa@tpru.ru

Шкляр Алексей Викторович, Институт кибернетики, Томский политехнический университет, Томск, Россия,
shklyarav@mail.ru

Алгоритм для определения положения пользователей в мировых координатах и его применение для задач слежения*

А.Д. Грингауз, Е.В. Шальнов, А.С. Конушин

grin3s@mail.ru | shalnov.eugen@gmail.com | ktosh@graphics.cs.msu.ru

Факультет вычислительной математики и кибернетики

Московский государственный университет им.М.В.Ломоносова

В статье представлен многопользовательский алгоритм слежения (трекинга), который работает в мировой системе координат. Этот алгоритм является модификацией подхода [1] и основан на минимизации энергии в течение временного скользящего окна. Он оценивает путь пользователя и его высоту в мировых координатах от известного места и положение его головы в координатах пространства изображения.

Ключевые слова: алгоритм, трекинг, система слежения, мировые координаты, координаты изображения, 3D слежение, MCMC DA, минимизация энергии, оценка высоты.

An algorithm for estimating locations of people in world coordinates and its application for tracking*

Alexander Gringauz, Eugene Shalnov, Anton Konushin

Lomonosov Moscow State University, Department of Computational Mathematics and Cybernetics

The paper presents a multi-person tracking algorithm that operates in world coordinates. This algorithm is a modification of the approach from [1] and is based on energy minimization within a temporal sliding window. It estimates a person's track and his height in world coordinates from known location of his head in image coordinates.

Keywords: multi-person tracking algorithm, world coordinates, image coordinates, 3D tracking, MCMC DA, energy minimization, height estimation.

Introduction

Multi-target tracking is an important computer vision task. It implies constructing trajectories for all people in a given video fragment. The trajectory contains a unique identifier for every person and his position in all frames of the video. This task is important for many applications, for example: video surveillance, improving pedestrian safety, marketing research. Despite a significant progress in recent years, humans are still far ahead of existing automatic algorithms in terms of solving this task.

All methods of object tracking can be divided into two big groups: visual tracking methods and tracking by detection methods. Algorithms from the first group can be used for all kinds of objects and don't require an object detector. An object's position on the first frame must be set manually. An algorithm then finds objects on the following frames that are similar to the object on the first frame. The "Flock of features" algorithm [2] is an example of such methods.

Unlike the visual tracking methods the tracking by detection methods don't require an initialization of the object's position on the first frame. They use an object detector (for example, a person detector) to find objects and then associate detections belonging to a single object into a track. Greedy methods and

bayesian models can be used to form tracks. The inference in bayesian models can be performed using MCMC [1] or graph cuts [3].

In case of a static camera with a known calibration matrix, we can estimate a person's 3D position in world coordinates. In [4, 5] the authors used multiple cameras. It allowed them to estimate 3D positions and height of all people more accurately. In [4] a 3D appearance model was introduced that allowed the algorithm to re-identify people more accurately, thus improving tracking performance. In [3] the authors showed that using 3D positions improved tracking performance. In [6] a particle-based tracker was used to construct trajectories on the ground plane in world coordinates.

The position and height of people is very important information that can be obtained from the scene. It can be used to determine the positions of people on a location map, which is useful for video surveillance systems.

We propose an approach to evaluate the location and height of the person in the world coordinates from the head location of the same person in different frames of the video. Then we embed this method into the person tracking algorithm [1]. Unlike other algorithms we do not use a full body detector to evaluate a person's location on the ground plane.

Работа выполнена и опубликована при финансовой поддержке РФФИ, гранты 14-01-00849, 15-07-20347

Proposed Method

Brief description The proposed method is a modification of the approach from [1].

We choose the key frames from the whole set of frames $\{I_t\}$ with the interval $step$. In our experiments $step = 5$. Then a detector (see. section 44) is applied to every key frame to find heads. In the next step the algorithm constructs a tracklet (see. section 44) for each head detection. The tracklet is a combination of a head's location and its motion information in a small video segment. All tracklets from the current frame are added into the sliding window. Tracklets that went out of the sliding window are removed from it. After that the set of trajectories T is updated using new tracklets. They are added to existing tracks or form new ones. Then final results are formed for the frames that are between the key frame in the middle of the sliding window and the following key frame. A track contains a head's position on the image, legs' position and height of a person on every frame he was detected on. $T = \{T_i\}$, $T_i = \{(x_t^{(i)}, y_t^{(i)}, w_t^{(i)}, h_t^{(i)}, X_t^{(i)}, Y_t^{(i)}, Z_t^{(i)}, H_t^{(i)})\}$, $i = \overline{1, J}$, where $x_t^{(i)}, y_t^{(i)}, w_t^{(i)}, h_t^{(i)}$ - head's bounding box parameters on the frame t , $X_t^{(i)}, Y_t^{(i)}, Z_t^{(i)}$ - legs' position in world coordinates, $H_t^{(i)}$ - height.

In the following sections a more detailed description of the algorithm is provided.

Finding people

The proposed algorithm is aimed for people tracking. Therefore we use a HOG based head detector to detect people on each frame. It performs better than a full body detector in case of occlusions [7].

Building tracklets

The algorithm constructs a tracklet for each found head in the key frame. A tracklet is an object that contains information about the detection's location and its *velocity estimates* in a small video segment. We use the Flock of Features visual tracking algorithm [2] to construct it. It uses only one detection and tracks it forward and backward in time for several frames. Information about the detection includes a time stamp, parameters of the bounding box, its size and detector confidence. For more details see [1].

Estimation of a person's 3D position In this section we describe the modified part of the base algorithm. All other parts of the proposed method are almost the same as the corresponding parts of the base algorithm.

Our goal is to embed three-dimensional coordinates into the algorithm. In order to achieve that we propose an algorithm to solve the task of estimating a single person's position and his height knowing his head's coordinates in the image.

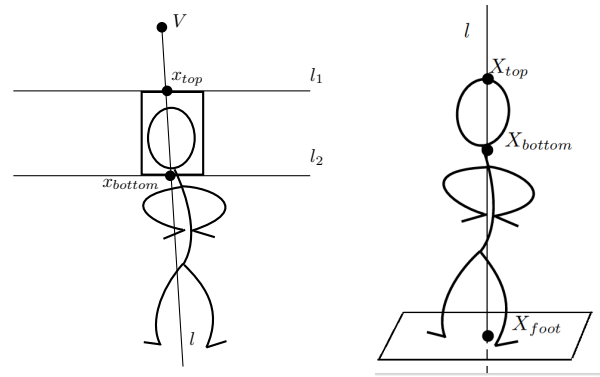


Figure 1: Visualization of the introduced variables.

Let $\{x_i, y_i, s_i\}, i = \overline{1, N}$ be the head's position and its size on several frames, N - number of frames the current person was detected on. P - the camera calibration matrix. A_{gp} - a perpendicular to the ground plane. R - the ratio of the person's height and his head's size (one of the algorithm's parameters, for all people it is the same). We use R equal to 6.5 in our experiments. Formally, our goal is to estimate the three-dimensional position of the person on A_{gp} : $\{X_i, Y_i, Z_i\}, i = \overline{1, N}$ and his height H .

Let's consider two approaches to solve this task: independently on each frame, jointly on all frames.

Position estimation on a single frame Lets introduce notations:

- $X_{foot} = [X, Y, Z, W]^T$ are the homogeneous coordinates of the person's feet in the world coordinates.
- X_{top} are the homogeneous coordinates of the top point of the person's head in the world coordinates.
- x_{top} are the homogeneous coordinates of the top point of the person's head on the image.
- H is the person's height in world coordinates.
- X_{bottom} are the homogeneous coordinates of the bottom point of the person's head in the world coordinates.
- x_{bottom} are the homogeneous coordinates of the bottom point of the person's head on the image.
- l_1 is the horizontal line on the image that goes through x_{top} .
- l_2 is the horizontal line on the image that goes through x_{bottom} .
- l is the projection of the vertical line going through X_{foot} onto the image.

These notations are shown on the figure 1.

The lines l_1 and l_2 can be computed by selecting two points on the head's bounding box. To compute the line l we must find the vanishing point of the vertical lines that are orthogonal to the ground plane. It can be found as shown in [8]:

$$V = PA_{gp} \quad (1)$$

Thus, l can be found using V and the center of the head's bounding box.

X_{foot} lies in A_{gp} :

$$A_{gp}X_{foot} = 0 \quad (2)$$

Normalization constraint on the last coordinate of X_{foot} :

$$W = 1 \quad (3)$$

The top point of the head can be found as:

$$X_{top} = X_{foot} + HA_{gp} \quad (4)$$

x_{top} is a projection of X_{top} onto the image:

$$x_{top} = PX_{top} \quad (5)$$

The bottom point of the head can be found as:

$$X_{bottom} = X_{foot} + \frac{R-1}{R}HA_{gp} \quad (6)$$

x_{bottom} is a projection of X_{bottom} onto the image:

$$x_{bottom} = PX_{bottom} \quad (7)$$

x_{top} lies on l_1 :

$$l_1x_{top} = 0 \quad (8)$$

x_{bottom} lies on l_2 :

$$l_2x_{bottom} = 0 \quad (9)$$

x_{top} lies on l :

$$lx_{top} = 0 \quad (10)$$

Let's solve the system (2) - (10) with respect to X, Y, Z, H . It can be transformed into the system of linear equations with the square non-singular matrix M and the right side b :

$$MX_{res} = b, \quad X_{res} = [X, Y, Z, H]^T \quad X_{res} = M^{-1}b \quad (11)$$

Estimating position on multiple frames If the person's positions on multiple frames are known then we can use the algorithm described in the previous section for every frame separately and average the values of height, computed for each frame.

We consider the person's height to be a constant for each person (if his doesn't jump, crawl or lie down). This assumption can be used to smooth errors that come from solving the system (11) for each frame separately. Despite the fact that the system (11) can be solved exactly, errors occur due to slightly inaccurate results produced by the detector.

This problem can be formulated as the system of linear equations that is a union of systems of the form

(11) for every frame with a common variable H , corresponding to the person's height.

$$M_{all}X_{all} = b_{all}, \quad X_{all} = [X_1, Y_1, Z_1, \dots, X_N, Y_N, Z_N, H]^T \quad (12)$$

The size of M_{all} is $4N \times (3N + 1)$, therefore the system (12) is over-determined and can only be solved approximately. One of possible approaches to solve it is to minimize the norm of the residual $\|M_{all}X - b_{all}\|$. In the current work we used the L-1 norm because it is more stable than the L-2 norm if there are outliers in data. We reduced this optimizations problem to the linear programming problem and solved it with the simplex method.

Our experiments show that in order to achieve the best results it is necessary to set different weights for the equations (2) - (10), that is to multiply them by a certain number. These weights show how strong the impact of every equation is on the residual while approximately solving (12). Those equations that must be satisfied very precisely should have larger weights compared to the rest. The equations (2) and (10) were multiplied by 1000, others were left without changes.

Building trajectories In this section an algorithm to combine tracklets into trajectories is described.

Let D be a set of all detection in the sliding window. H - a hypothesis that defines how D is divided into trajectories, that is: $H = \{T_1, \dots, T_J\}$, $T_i = \{d_n^j\}$, d_n^j - the n^{th} detection of the j^{th} trajectory. Now we define a generative model for detections and trajectories:

$$p(D, H) = p(D|H)p(H) \quad (13)$$

We need to find:

$$H^* = \operatorname{argmax}_H p(D, H) \quad (14)$$

This probabilistic model is almost the same is the model from [1]. The inference is performed using MCMC. The only difference is that the legs' position in world coordinates is used while computing the likelihood of the trajectory hypothesis. The legs' positions are computed using the algorithm from the section 44. Besides, it is necessary to translate the velocity estimates in tracklets from the image coordinates to the world coordinates. For more details see [1].

Experimental evaluation

The aim of the experimental evaluation is to compare the base algorithm [1] with its modification. The modified part consists in estimating people's positions and height in world coordinates and using this information for tracking (see sections 44 and 44). We want to find out how the usage of world coordinates affects tracking performance.

We use the TownCenter dataset for numerical comparison of the proposed algorithm with the basic one. It is a high resolution ($1920 \times 1080/25\text{fps}$) video, framed from a static camera. The calibration matrix for the camera and the ground truth are provided. The method is implemented in MATLAB using C++ MEX-functions. The time the algorithm needs to process 1000 frames using the head detector once in 5 frames is ~ 4 hours on a computer with Inter Core i7, 12 GB RAM.

To evaluate the quality of the algorithms standard metrics like precision and recall along with the CLEAR MOT [10] group of metrics were used: FP - number of false positives, FN - number of false negatives, ID - number of identity switches, MOTA - a total error that takes into account FP, FN and ID; MOTP shows how close the trajectory lies to the real person's position obtained from the ground truth.

The results are shown in the table 1. All metrics were computed in the image space.

Algorithm	Base	Modification
Precision	80.7	71.4
Recall	74.7	75.6
FP	2609	4407
FN	3696	3567
ID	22	43
MOTA	56.6	45.0
MOTP	47.2	47.2

Table 1: Comparison of the base and the modified algorithm.

Analyzing results

The experimental results show that the proposed method can successfully estimate a person's leg position and his height in world coordinates. However, the proposed method shows slightly lower accuracy than the base method. The main drawback of the proposed method of finding positions of people in world coordinates is the necessity to know the ratio between a person's height and his head's size. The head detector produces bounding boxes that may have different size. They may be shifted from the head's actual position. These reasons can lead to errors in estimating the leg position and height.

Conclusion

In this work we proposed a new method of estimating the position in world coordinates and height of a person using his head's position on the image. Based on this method we design a modified version of the people tracking algorithm that can find trajectories and persons' heights in world coordinates. Despite some shortcomings, this approach allows us to use 3D



Figure 2: An example of tracking results.

positions of people for tracking. The further development of the algorithm may include: code optimization, adding new factors to the probabilistic model (appearance, foreground [12]), expanding the method in order to process scenes filmed from multiple cameras [11] (in this case we don't need to know the ratio between the head size and the full body size).

Bibliography

- [1] Shalnov E., Konushin A. Improvement of MCMC-based video tracking algorithm. Pattern recognition and image analysis (PRIA-11-2013), 2013, p.727-730.
- [2] Kolsch M., Turk M. Fast 2d hand tracking with flocks of features and multi-cue integration //Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on. – IEEE, 2004. – p. 158-158.
- [3] Andriyenko A., Schindler K., Roth S. Discrete-continuous optimization for multi-target tracking //Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. – IEEE, 2012. – p. 1926-1933.
- [4] Baltieri D. et al. Multi-view people surveillance using 3D information //Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on. – IEEE, 2011. – p. 1817-1824.
- [5] Berclaz J. et al. Evaluation of probabilistic occupancy map people detection for surveillance systems //Proceedings of the IEEE International Workshop on Performance Evaluation of Tracking and Surveillance. – 2009. – num. LIDIAP-CONF-2009-064.
- [6] Batanov P., Kononov V., Konushin A. People tracking in surveillance systems for sport games using multiple cameras. Graphicon, 2013, pp. 333-336 (in Russian).
- [7] Konushin A., Filippov I., Konushin V., Kononov V. Counting people in a video sequence based on head detection. Programmnye produkty i sistemy, no. 1, 2015, pp. 121-126 (in Russian).
- [8] Hartley R., Zisserman A. Multiple view geometry in computer vision. – Cambridge university press, 2003.
- [9] Milan A., Roth S., Schindler K. Continuous energy minimization for multitarget tracking //Pattern Analysis and Machine Intelligence, IEEE Transactions on. – 2014. – Vol. 36. – num. 1. – p. 58-72.
- [10] Keni B., Rainer S. Evaluating multiple object tracking performance: the CLEAR MOT metrics

- //EURASIP Journal on Image and Video Processing. – 2008.
- [11] Kononov V., Konushin V., Konushin A. People Tracking Algorithm for Human Height Mounted Cameras. DAGM, 2011, pp. 163-72.
- [12] Chetverikov N., Konushin A. Finding objects in a video stream using graph cuts. Graphicon, 2012, pp. 262-265 (in Russian).
- [13] Gringauz A., Shalnov E., Konushin A. Modification of the Multi-target Tracking Algorithm Based on Energy Minimization. GraphiCon-2014, 2014, pp.139-142.

Разработка и развитие системы интерактивного визуального анализа многомерных данных*

О.П. Масленников[†], И.Е. Мильман[†], А.Э. Сафиуллин[†], А.Е. Бондарев[‡],
Ш.У. Низаметдинов[†], В.В. Пилыгин[†]

maslolpavl@gmail.com | igalush@gmail.com | amir147@rambler.ru | bond@keldysh.ru
sh_nizam@mail.ru | VVPilyugin@mephi.ru

[†]НИЯУ МИФИ, Москва, Россия;

[‡]Институт прикладной математики им. М.В.Келдыша РАН, Москва, Россия

Работа посвящена развитию интерактивной системы для решения задач анализа многомерных данных интерактивным визуальным методом. Визуальная аналитика предоставляет удобные средства для решения задач анализа многомерных данных и в данной работе показано развитие инструментария для решения задачи кластеризации и дискриминантного анализа. Разрабатываемая система позволяет отобразить многомерное облако данных и проводить его анализ в пространствах меньшей размерности (2D и 3D), выдвигать и проверять различные гипотезы об исходных данных, с возможностью последующего построения предположений для проведения некоторых счетных методов, с помощью геометрических построений в интерактивном режиме.

Ключевые слова: визуальная аналитика, анализ многомерных данных, интерактивный интерфейс

Design and development of system for interactive visual analyzing of multidimensional data*

O.P. Maslennikov[†], I.E. Milman[†], A.E. Safulin[†], A.E. Bondarev[‡],
Sh.U. Nizametdinov[†], V.V. Pilyugin[†]

[†]National Research Nuclear University MEPhI, Moscow, Russian Federation;

[‡]Keldysh Institute for Applied Mathematics RAS, Moscow, Russian Federation

The article presents a development of interactive system intended for visual analyzing of multidimensional data. The examples and illustrations are enclosed. Considered problems can be referred to visual analytics. By means of described interactive system user can work directly with data volume in question projections to 2D and 3D subspaces. Also the user is able to verify hypotheses about types of data inside the volume by interactive geometrical constructions.

Keywords: visual analytics, multidimensional data, interactive system

1. Введение

Современные задачи обработки и анализа огромных разнородных объемов информации требуют интенсивного развития методов, принципов и программных средств, позволяющих осуществить их решение. В роли средства решения выступает сравнительно молодая междисциплинарная ветвь исследований – визуальная аналитика. Методы визуальной аналитики интенсивно внедряются во все значимые прикладные аспекты человеческой деятельности. В практической плоскости визуальную аналитику можно рассматривать как решение задач анализа данных с использованием способствующего интерактивного визуального интерфейса, т.е. визуальная аналитика призвана организовать человеко-машинный интерфейс, усиливающий человеческие аналитические способности [1]. Основные методы, подходы и алгоритмы визуальной аналитики описаны в работах [2–6]. В этих же работах приведен ряд примеров современного при-

менения визуальной аналитики в различных сферах человеческой деятельности, а также приведены описания ряда программных продуктов, построенных на основе визуальной аналитики [1].

Внимательное изучение литературы, посвященной описанию конкретных приложений в области визуальной аналитики, позволяет утверждать, что в реальности интерактивным системам работы с многомерными данными зачастую придается меньшее значение по сравнению с системами постпроцессинга результатов применения методов Data Analysis [1].

Данная работа представляет расширение разрабатываемой интерактивной системы визуального анализа многомерных данных. В рамках данной системы рассматриваются классические задачи анализа многомерных данных, такие как: выявление кластеров в многомерном облаке данных, построение системы решающих правил для процедур классификации объектов, реализация отображения многомерного объема данных в двумерных проекциях на все возможные пары координат. Разрабатываемая система позволяет пользователю:

Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347

- непосредственно работать с отображениями данных в пространствах меньшей размерности – двумерных и трехмерных;
- выдвигать гипотезы о наличии кластеров и классов в облаке данных и проверять их непосредственно с помощью интерактивного геометрического моделирования;
- выделять аномальные объекты в зависимости от расстояния между объектами многомерного пространства;
- принимать решения о возможности построения решающих правил для задач классификации новых объектов;
- проводить непосредственный поиск кластеров по множеству двумерных проекций и визуальный анализ значимости координатных направлений с точки зрения вклада в дисперсию.

Следует также отметить, что разрабатываемая интерактивная система, дает в перспективе возможность при дальнейшем применении математических методов анализа многомерных данных использовать полученные геометрические построения и гипотезы в качестве начальных приближений для более точных вычислений. При разработке интерактивной системы использовались материалы работ [7-10].

2. Интерактивный визуальный анализ

В рамках разрабатываемой интерактивной системы визуального анализа на сегодняшний день реализовано решение следующих задач.

Решение задачи кластерного анализа 3D проекционным методом

Решение данной задачи обеспечивает пользователю возможность интерактивной работы с проекциями исходного многомерного пространства в трехмерных пространствах, образованных из исходных координат по выбору пользователя. Пользователю предоставляется возможность интерактивного построения кластеров и выделение удаленных точек (аномальных объектов).

Идея метода заключается в том, что при проекции задается параметр d , отвечающий наибольшему внутрикластерному расстоянию. Если расстояние в исходном пространстве между двумя точками меньше чем d , то между данными точками строится отрезок. Точки во время проекции представляются сферами, а отрезки цилиндрами (Рис.1) [1].

Оптическая модель сцены предполагает присвоение цилиндрам цвета, отвечающему расстоянию между точками. Чем ближе расстояние к d , тем синее цилиндр.

Пользователю предлагается проводить анализ разбиения на подмножества в зависимости от параметра d .

Предусматривается два метода перехода – последовательный просмотр при задаваемых пользователем d или задание двух значений параметра и построение видеоряда.

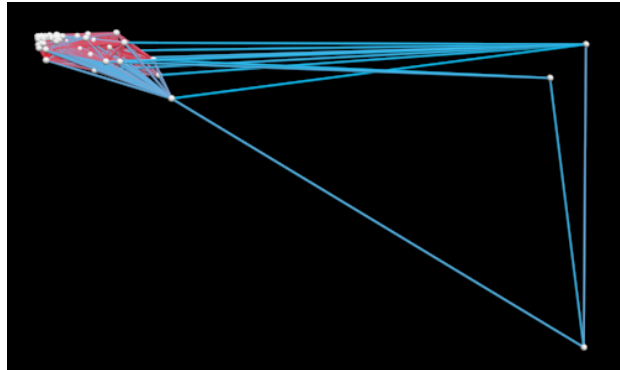


Рис. 1: Отображение множества точек

Возможными подмножествами являются:

Кластер – подмножество, при заданном множестве точек, попарное расстояние между которыми не превышает заданное d , а расстояние между точками кластера и остальными точками не меньше заданного d .

Удаленная (одиночная) точка – точка, удаленная от всех остальных точек исходного множества на расстояние, большее заданного d . В качестве примера можно привести зеленую точку на рис. 2.

Сгусток – подмножество точек, большая часть расстояний между которыми не превышает заданное d .

Квазиудаленная (Квазиодиночная) точка – точка, не являющаяся удаленной, но и не входящая в сгусток или кластер при заданном разбиении. В качестве примера можно привести желтую точку на рис. 2.

Определение точки к тому или иному подмножеству выполняется аналитиком в интерактивном режиме в зависимости от пространственной сцены, полученной при заданном d .

При разбиении на подмножества, пользователям предлагается использовать различные цветовые обозначения для распределения точек по подмножествам.

После получения удаленных точек, выполняется этап микроанализа. На данном этапе выявляется, какие координаты точек вносят наибольший вклад в расстояние между точками, а какие координаты влияют слабо.

Для этого предлагается строить графическую проекцию на плоскость (X_i, Y_i) . При этом используются те же цветовые обозначения, введенные на предыдущем шаге.

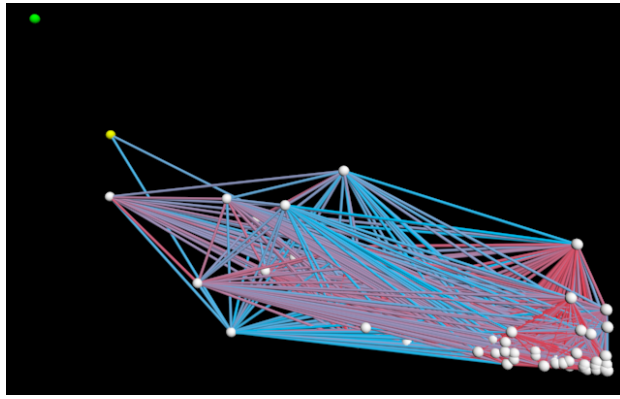


Рис. 2: Примеры подмножеств точек

На рис. 3 представлена плоская проекция результатов выполнения макроанализа на одну из координат. На данной проекции видно, что анализируемая координата вносит значительный вклад в удаленность зеленой точки, однако вносит малый в удаленность фиолетовой и розовой точек.

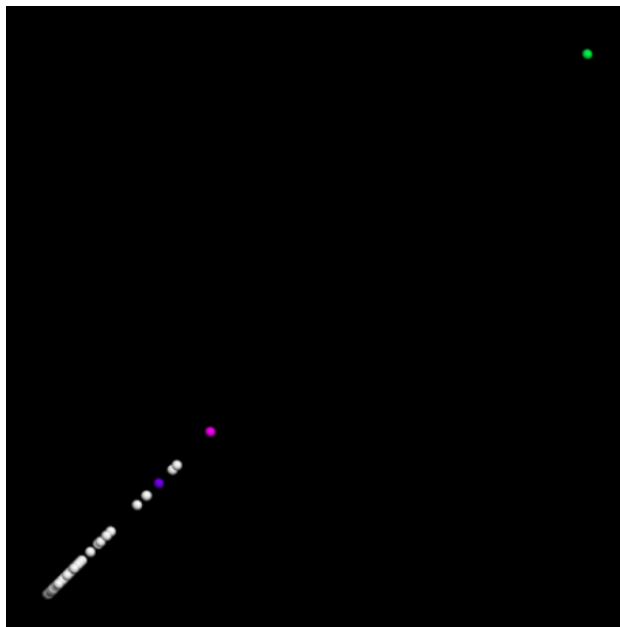


Рис. 3: Плоская проекция исходного подмножества

Решение задачи дискриминантного анализа 2D и 3D проекционным методом

Второй задачей, реализованной на сегодняшний день в разрабатываемой системе интерактивного визуального анализа многомерного облака данных, является классическая задача дискриминантного анализа. Задача решается с помощью проекционного метода, реализованного для 2D и 3D случаев. Основным предположением дискриминантного анализа является то, что существуют две или более группы, которые по некоторым переменным отличаются от других групп, причем такие перемен-

ные могут быть измерены по интервальной шкале либо по шкале отношений. Дискриминантный анализ помогает выявлять различия между группами и дает возможность классифицировать объекты по принципу максимального сходства [10].

В качестве алгоритма решения задачи было предложено поочередно применять следующие методы:

1. Метод построения разделяющей гиперплоскости с построением последовательных проекционных изображений.

Гиперплоскость называется разделяющей для 2х групп точек, если при подстановке координат точки в аналитическое описание этой гиперплоскости все точки одной группы будут иметь один знак, а все точки другой группы – противоположный. Если в проекции плоскость является разделяющей, то при переходе к пространству с размерностью на единицу больше, данная плоскость будет являться так же разделяющей. Таким образом, предлагается последовательный обход через возможные 2х и 3х мерные пространства и поиск в каждом из таких пространств разделяющей прямой или плоскости.

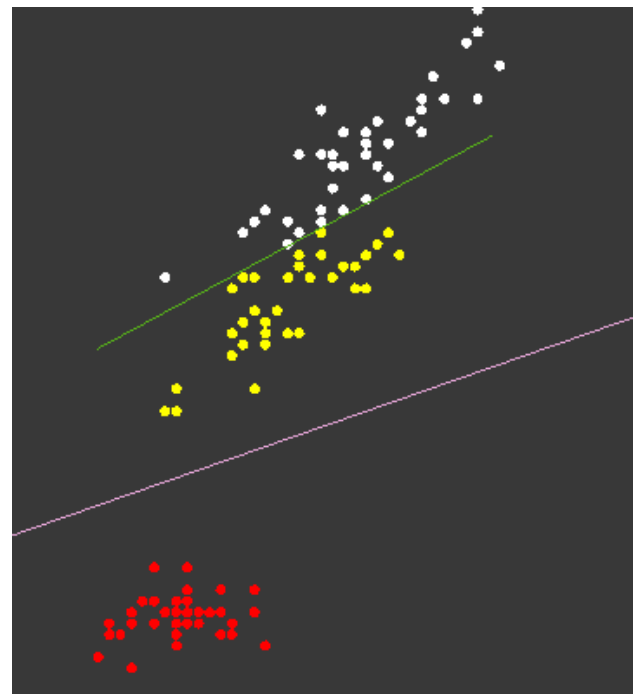


Рис. 4: Построение решающего правила в двумерном режиме работы

2. Переход в пространство главных компонент и применение метода построения разделяющей гиперплоскости в них.

В пространство главных компонент приводит линейное преобразование исходного пространства. При линейном преобразовании разделяющая плоскость продолжает являться разделяющей. Обратное преобразование также является линейным. Та-

ким образом, при построении решающего правила в пространстве главных компонент пользователь получает решающее правило и в исходном пространстве также. Переход в пространство главных компонент позволяет получить дополнительное пространство для поиска решения задачи и увеличение числа значимых коэффициентов в аналитическом описании гиперплоскости.

3. Метод построения решающего правила в виде теоретико-множественных операций.

В качестве решающего правила предложено строить систему гиперплоскостей. На двумерной проекции система гиперплоскостей представляет собой многоугольник, построенный пользователем. С помощью такой сложной фигуры можно разделить группы точек, которые не удалось разделить с помощью применения предыдущих методов. В качестве решающего правила будет выступать теоретико-множественное отношение в базисе «и, или, не», в качестве элементов множества выступают ребра многоугольника.

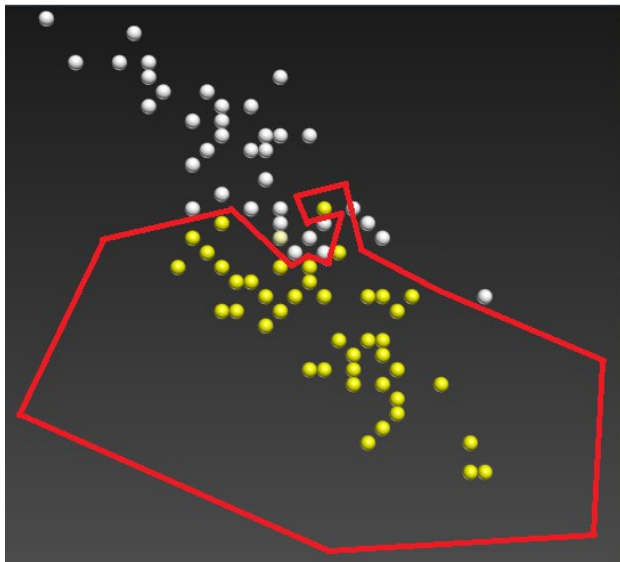


Рис. 5: Окаймляющий многоугольник и отношение, задающее его

После получения решающего правила решается задача о верификации новых объектов.

Решение задачи выделения кластеров 2D проекционным методом

В основе разрабатываемого метода разбиения исходного множества на кластеры лежит использование гипотезы компактности. Полагаясь на данную гипотезу, можно сделать суждение о том, что точки, близкие в каждом двумерном подпространстве, близки в n -мерном пространстве, а точки далекие в исходном пространстве, далеки в одном из двумерных подпространств.

Алгоритм решения задачи [11]:

1. Точки многомерного множества данных проецируются во все двумерные арифметические пространства; в итоге получается «матрица проекций».
2. На одной из проекций выделяются характерные образования – кандидаты на скопление.
3. Анализируются остальные проекции и в случае обнаружения точек, лежащих далеко от скопления, они исключаются.
4. Оставшиеся выделенные точки помечаются как кластер и исключаются из рассмотрения. Если сгустков не осталось (то есть остались лишь одиночные (изолированные) точки, либо все точки помечены как кластер) то задача решена, иначе следует вернуться к пункту 2.

Полученные визуальным методом кластеры при необходимости можно использовать как начальное приближение для модифицированного метода k -средних. Количество кластеров, подающихся на вход модифицированного алгоритма k -средних, равняется числу выделенных кластеров плюс число оставшихся одиночных точек. Для наглядности представления результатов кластеризации, помимо представления результатов в табличном виде (объект - номер кластера), было решено реализовать метод двумерного представления объектов кластеризации, называемый профильной диаграммой (диаграмма с параллельными координатами).

После проведения кластеризации, объекты, объединенные в одну группу, помечаются соответствующим номеру кластера цветом. Эта возможность позволяет оценивать адекватность полученного разбиения и помогает в его интерпретации. Представив таким образом полученные результаты, пользователь может определить, какие объекты являются спорными, т.е. могут принадлежать как одному, так и другому кластеру, какие группы вообще не имеют места быть, в силу невозможности их интерпретации с точки зрения здравого смысла.

Так же в качестве одного из методов визуализации иерархической кластеризации применяются дендрограммы. В случаях, когда число признаков велико (больше 40) диаграмму рассеяния становится сложно визуально анализировать. С помощью метода главных компонент программа позволяет снизить размерность признакового пространства. На рис. 1. представлены многомерные данные – изображены лиц двух национальностей. После сокращения размерности представляется возможность наблюдать диаграммы рассеяний – проекции изображений в признаковое подпространство. Как видно по первым двум компонентам отчетливо видна разделимость двух групп.

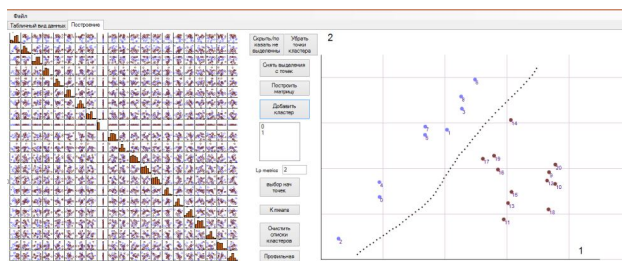


Рис. 6: Диаграммы рассеяния и увеличенная проекция значений на двумерное подпространство из первых двух главных компонент

3. Заключение

В данном докладе представлен ряд реализованных задач, относящихся к развивающейся в настоящее время интерактивной системе визуального анализа многомерных данных. Основная цель данной разработки – предоставление пользователю возможности интерактивной работы с двумерными и трехмерными проекциями исходного многомерного объема данных для получения первичной информации о структуре изучаемого объема и взаимном расположении точек в изучаемом объеме.

Литература

- [1] Масленников О.П., Мильман И.Е., Сафиуллин А.Э., Бондарев А.Е., Низаметдинов Ш.У., Пилюгин В.В. Интерактивный визуальный анализ многомерных данных / ГрафиКон'2014: 24-я Международная конференция по компьютерной графике и зрению: Ростов-на-Дону, Академия архитектуры и искусств ЮФУ Труды конференции. – С.51–54.
- [2] Thomas J., Cook K. Cook, Illuminating the Path: Research and Development Agenda for Visual Analytics. IEEE-Press, 2005.
- [3] Keim D. A, Mansmann F, Schneidewind J, Thomas J, Ziegler H: Visual analytics: Scope and challenges, Visual Data Mining: 2008, S. 82.
- [4] Keim D., Andrienko G., Fekete J.-D., Gorg C., Kohlhammer J., and Melancon G. “Visual Analytics: Definition, Process, and Challenges”, A. Kerren et al. (Eds.): Information Visualization, LNCS 4950, pp. 154–175, 2008. Springer-Verlag Berlin Heidelberg 2008.
- [5] Kielman, J. and Thomas, J. (Guest Eds.) (2009). Special Issue: Foundations and Frontiers of Visual Analytics, Information Visualization, Volume 8, Number 4, Winter 2009, p. 239-314.
- [6] Keim D., Kohlhammer J., Ellis G. and Mansmann F. (Eds.), Mastering the Information Age – Solving Problems with Visual Analytics, Eurographics Association, 2010.
- [7] Пилюгин В.В., Маликова Е.Е., Пасько А.А., Аджиев В.Д. Научная визуализация как метод анализа научных данных / Научная визуализация. Т.4, № 4, с.8-25, 2012, URL: <http://sv-journal.org/2012-4/06.php?lang=ru>
- [8] Бондарев А.Е., Галактионов В.А. Анализ многомерных данных в задачах многопараметрической оптимизации с применением методов визуализации / Научная визуализация. Т.4, № 2, с.1-13, 2012, URL: <http://sv-journal.org/2012-2/01.php?lang=ru>
- [9] Основы научной визуализации [сайт]. URL: <http://sv-journal.org/unl> (дата обращения: 10.05.2015)
- [10] Низаметдинов Ш.У. Анализ данных. М.: МИФИ, 2006
- [11] Масленников О. П., Мильман И.Е., Сафиуллин А.Э., Бондарев А.Е., Низаметдинов Ш.У., Пилюгин В.В. Разработка системы интерактивного визуального анализа многомерных данных//Научная визуализация, МИФИ, 2014, 4, С.30-49. URL: <http://sv-journal.org/2014-4/04.php?lang=ru>

Оптическая система захвата движения для анализа и визуализации трехмерных процессов*

В.А. Князь

knyaz@gosniias.ru

ФГУП Государственный научно-исследовательский институт авиационных систем, Москва, Россия

Качество решения технических задач зависит от вида, достоверности и степени полноты данных, используемых для анализа исследуемых процессов. Важным классом задач является анализ и визуализация сложных, быстротекущих перемещений и деформаций объектов и их частей, требующий специальных средств регистрации движения. В работе представлены результаты разработки и применения оптической системы захвата движения, построенной на фотограмметрических методах обработки изображений, обеспечивающих высокую точность измерений. Приведен обзор современных систем захвата движения, представлено техническое описание разработанной системы, основных алгоритмов, обеспечивающих решение комплекса задач захвата трехмерного движения. Обсуждаются результаты применения для решения различных классов задач, связанных с обработкой, анализом и визуализацией трехмерного движения.

Ключевые слова: захват движения, обработка изображений, 3D анализ данных, 3D визуализация, калибровка, автоматизация.

Optical Motion Capture System for 3D Process Analysis and Visualization *

V.A. Knyaz

State Research Institute of Aviation Systems (GosNIIAS), Moscow, Russia

Paper presents the review of motion capture systems based on different physical principles and the results of development, evaluation and application for analysis and visualization of vision-based 3D motion capture system. The description of the hardware and original software of developed system is given along with its general technical characteristics. The results of the system application for various tasks solution such as estimation of slow motion, robot dynamical model identification, UAV navigation accuracy estimation are presented.

Keywords: motion capture, image processing, 3D data analysis, 3D visualization, calibration, automation.

Введение

Качество решения технических задач зависит от вида, достоверности и степени полноты данных, используемых для анализа исследуемых процессов. Важным классом задач является анализ и визуализация сложных, быстротекущих перемещений и деформаций объектов и их частей, требующий специальных средств регистрации движения. В работе представлены результаты разработки и применения оптической системы захвата движения, построенной на фотограмметрических методах обработки изображений, обеспечивающих высокую точность измерений.

Приведен обзор современных систем захвата движения, представлено техническое описание разработанной системы, основных алгоритмов, обеспечивающих решение комплекса задач захвата трехмерного движения. Обсуждаются результаты применения для решения различных классов задач, связанных с обработкой, анализом и визуализацией трехмерного движения.

Работа выполнена при поддержке Гранта Президента для Ведущих научных Школ России (НШ-3477.2014.8) и поддержке РФФИ, грант 5-08-99580. Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347.

Типы систем захвата движения. В настоящее время для захвата движения применяется несколько типов систем, построенных на различных физических принципах. Среди них - механические, акустические, магнитные и оптические системы.

Механические системы используют потенциометры, расположенные в заданных точках на теле актера и обеспечивающие регистрацию пространственных положений. К преимуществам таких систем можно отнести интерфейс, аналогичный системе покадровой съёмки, широко используемой в киноиндустрии; независимость от магнитных полей или отражений и быструю настройку. С другой стороны, их существенным недостатком являются ограничения, накладываемые проводами, соединяющих датчики с системой регистрации и ограниченность применения для сложных или быстротекущих процессов.

В акустической системе набор акустических приемников принимает звуки из звуковых передатчиков, расположенных на объекте (актере). Сигналы от излучателей принимаются приемниками, а дальность до передатчика рассчитывается по промежутку времени между переданным и принятым сигналом. Для определения положения в простран-

стве каждого передатчика выполняется триангуляция рассчитанных расстояний между излучателем и каждым из приемников.

Акустические системы захвата движения имеют ряд недостатков, таких как: ограничение на свободу перемещений, обусловленной проводами, прикрепленными к актеру, ограниченное число передатчиков, которые могут быть использованы, и восприимчивость к отраженным сигналам или внешнему шуму.

Будучи относительно недорогими, магнитные системы обладают достаточно высокой точностью и быстродействием (около 100 кадров в секунду) для захвата относительно несложного движения объекта. Они используют набор магнитов в качестве маркеров заданных точек и набор приемников для измерения положения и ориентации маркеров по отношению к антенне [1]. К недостаткам магнитных систем можно отнести ограничения, вызванные кабелями и возможной интерференцией в магнитном поле, вызванной различными металлическими конструкциями.

Типичные оптические системы захвата движения используют методы обработки изображений для обнаружения заданных точек объекта. Как правило, они включают набор видеокамер для записи последовательностей изображений актера/объекта, на котором размещены специальные метки. Затем видеопоследовательности обрабатываются для обнаружения заданных точек, идентификации и прослеживания движения. Точность расчёта 3D-координат точек определяется процедурой калибровки системы и зависит от задач приложения. Эти системы являются наиболее дорогими на рынке вследствие применения передовых технологий, а именно: камер высокого разрешения и высокой частоты съемки и сложного запатентованного программного обеспечения. Стоимость таких систем достигает сотен тысяч долларов.

Несомненными преимуществами оптических систем является возможность захвата движения на очень высокой скорости, отсутствие ограничений для актера при передвижении и большой потенциал для автоматизации процесса обработки.

Области применения. Первоначальный импульс созданию систем захвата движения был дан индустрией развлечений, в которой появилась необходимость точного захвата быстрых движений актёра. И по сей день киноиндустрия и индустрия видеоигр являются основными потребителями данных от систем захвата движения. Сфера применения систем захвата движения расширяется очень динамично. К основным областям применения относятся медицина, спорт, различные отрасли промышленности, научные исследования [2].

Системы захвата движения в медицинских целях используются для точного анализа движений человека, которые не могут быть зарегистрированы с помощью других средств. Подробное изучение движения человека, его численный анализ и визуализация позволяют понять существо проблем и предложить способы их устранения.

На высокоточные системы захвата движения существует большой спрос в спорте высоких достижений, вследствие того, что получаемая с их помощью информация о движениях спортсмена во время соревнований регистрируется с очень высокой скоростью. Это информация является основой для улучшения спортивной техники и достижения более высоких результатов.

Захват движения часто является единственным инструментом в научных исследованиях конкретных задач, где информация о 3D-движении объекта не может быть получена с помощью других средств. Типичными примерами являются анализ очень быстрого динамического процесса, анализ вибраций, расчет и анализ 3D траектории объекта и другие.

Система захвата движения

Целью разработки являлось создание недорогой системы захвата движения, обеспечивающей высокую скорость регистрации данных и высокую точность расчета трехмерных координат заданных точек, позволяющих решать задачи реконструкции трехмерного движения, в первую очередь, технических систем для анализа их технических характеристик.

Разработанная система захвата 3D-движения, получившая название "Mosca" (Motion capture scalable system), основана на фотограмметрических методах 3D-измерений и обеспечивает возможность работы при различных размерах рабочей области и высокий уровень автоматизации обработки захваченных данных. В зависимости от области применения система может быть легко модифицирована для рабочих пространств от 100 мм до 10 м.

Структура и технические характеристики.

Разработанная система захвата движения (рис. 1) использует от 2 до 4 камер технического зрения для получения видеопоследовательностей движения объекта. Все камеры работают в синхронном режиме со скоростью кадров до 100 кадров в секунду под управлением персонального компьютера (ПК), обеспечивая возможность точного расчета 3D-координаты заданных точек. Система может быть расширена за счет дополнительных камер при включении дополнительного персонального компьютера в систему.



Рис. 1: Система захвата движения Mosca

Для обеспечения высокой точности 3D измерений на предварительном этапе работы системы выполняется процедура калибровки и внешнего ориентирования по заданному тестовому полю, обеспечивающая точность измерений на уровне 0.01% размера рабочего пространства. Процедура калибровки автоматизирована за счет применения оригинальных кодированных меток (Князь, 1998) для детектирования, идентификации и измерения координат опорных точек тестового поля на изображениях. Основные технические характеристики системы приведены в таблице 1.

Таблица 1: Технические характеристики системы.

Разрешение камеры	656 × 491 пикс
Скорость захвата	до 100 кадр/сек
Кол-во отслеживаемых точек	до 200
Рабочее пространство	Настраиваемое: от 0.1 × 0.1 × 0.1 м до 10 × 10 × 10 м
Точность координат 3D-точек	0.0% РП

Возможность изменения масштаба изображения обеспечивается быстрой и высокоавтоматизированной процедурой калибровки и внешнего ориентирования.

Программно-алгоритмическое обеспечение. Система захвата движения Mosca работает под управлением оригинального программного обеспечения, реализующего законченный цикл получения и первичной обработки данных, а именно:

- калибровка и внешнее ориентирование системы в автоматизированном режиме;
- синхронный захват видеопоследовательностей с 4 камер системы;
- детектирование и отслеживание меток на видеопоследовательности;
- расчет трехмерных траекторий заданных точек;
- построение скелетной модели объекта;
- визуализация захваченного трехмерного движения объекта.

Программный интерфейс системы с обнаруженными и идентифицированными метками и 3D-моделью скелета, построенной с использованием захваченных данных показан на рис. 2.



Рис. 2: Интерфейс программного обеспечения

Для системы захвата движения, основанной на обработке изображений, надежность и удобство для пользователя являются ключевыми факторами, определяющие качество системы. Поэтому при разработке существенное внимание было уделено возможности автоматизации процессов детектирования, идентификации, отслеживания меток, определяющих движение объекта. Ряд оригинальных алгоритмов [2] обнаружения меток и сопоставления одинаковых меток на снимках с разных камер системы, отслеживания и разрешения коллизий меток в 2D и 3D пространстве был разработан и реализован в программном обеспечении системы.

Приложения

Разработанная фотограмметрическая система захвата движения использовалась для решения ряда задач, требующих точных и снятых с высокой частотой данных о движении объекта. Среди них биометрические и биомеханические приложения, оценка равномерности движения динамического стенда на малых скоростях, идентификация динамической модели четырехколесного робота, расчёт точности

оценки собственного положения беспилотных летательных аппаратов, контроль виртуальных объектов и другие.

Оценка равномерности движения стенда. Важным инструментом полунатурного моделирования являются многостепенные динамические стенды (рис. 3), позволяющие имитировать динамику движения летательного аппарата в замкнутом контуре управления. Одним из режимов функционирования стенда является работа на малых угловых скоростях (порядка Для оценки качества выполнения данного режима требуется определить точность и равномерность выдерживания заданной угловой скорости.



Рис. 3: Динамический многостепенный стенд

Традиционные средства измерений угловой скорости, использующиеся сегодня при аттестации динамических стендов, позволяют проводить измерения с темпом порядка 10 Гц, что недостаточно для надежной оценки равномерности угловой скорости на малых скоростях.

Вместе с тем, применение оптической системы захвата движения обеспечивает решение данной задачи с требуемой точностью и временным разрешением. На рис. 4 приведены результаты обработки траекторий 12 меток, расположенных на вращающемся кольце стенда.

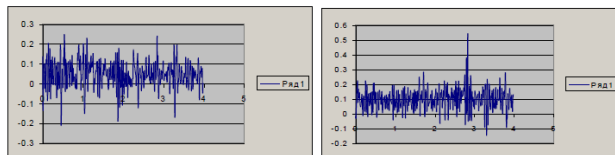


Рис. 4: Графики изменения угловой скорости стенда от времени ($\omega_{зад} = 0.05(^{\circ}/) - 0.1(^{\circ}/).$)

Идентификация динамической модели четырёхколёсного робота. Задачи идентификации

систем требует знания реакции системы на заданное входное воздействие. В случае колесного робота необходимо зарегистрировать выходные параметры (скорость v_x и угловое положение θ) робота, синхронно с входными командами управления. Для получения необходимой информации могут быть использованы акселерометры, но недостаточная точность измерений и необходимость синхронизации требуют применения дополнительного оборудования.

Применение системы захвата движения позволяет эффективно решить данную задачу, при включении в программное обеспечение специальный программный блок синхронизации входных команд робота и захвата кадров.

Для регистрации движений робота Hercules (рис. 5) в ходе эксперимента на его верхней панели был расположен набор круглых меток, используемых для расчёта выходных параметров v_x и θ .

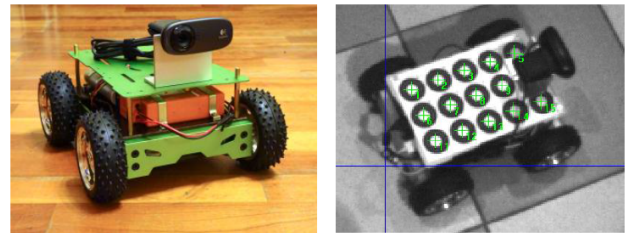


Рис. 5: Четырёхколёсный робот Hercules.

Чтобы оценить полную динамику модели робота Hercules, генерировалась последовательность команд в форме белого шума. Последовательность входных команд, продольная скорость v_x и скорость вращения θ , полученные с помощью системы захвата, показаны на рис. 6.

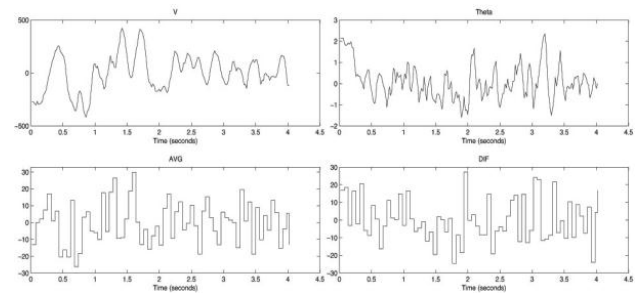


Рис. 6: Выходные значения v_x , и входные команды (белый шум).

Эти данные позволили идентифицировать модель системы и обеспечить требуемое качество управления роботом.

Оценка точности БЛА. Для оценки качества работы алгоритмов по определению собственного положения системы машинного зрения необходимо

сравнение результатов собственной оценки с данными независимых измерений. В случае беспилотного летательного аппарата (БЛА) наиболее рациональным решением является использование оптической системы захвата движения.

Исследования проводились с использованием БЛА Parrot AR.Drone2.0, имеющим в составе оборудования две видеокамеры: фронтальную и вертикальную. Изображения, поступающие с фронтальной камеры, использовались в полете для оценки собственного положения БЛА.

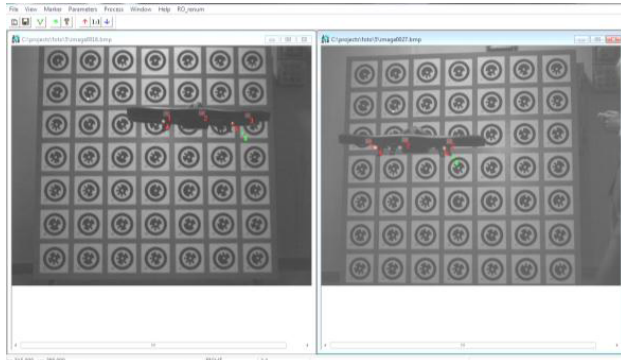


Рис. 7: Изображения БЛА, полученные системой захвата.

Одновременно полет БЛА с нанесенными на него метками расчета трехмерной траектории регистрировался системой Mosca (рис. 7). Оба набора данных (собственная оценка положения и внешние траекторные измерения) были зарегистрированы в общей системе координат, определяемой калибровочным полем. Средние ошибки в положении и угловой ориентации БЛА приведены в таблице 2.

Таблица 2: Средние ошибки БЛА.

α	ω	\varkappa
0.0232	-0.08392	0.03281
X, mm	Y, mm	Z, mm
-5.2618	8.0955	27.2022596

Выводы

Разработана оптическая система захвата движения Mosca, обеспечивающая высокую точность измерений и высокий уровень автоматизации процессов обработки данных на основе оригинальных алгоритмов для обнаружения, идентификации и отслеживания точек объекта.

Результаты применения системы Mosca в различных областях применения, таких как биомеханика, идентификация динамической модели, оценка точности ориентирования беспилотных летательных аппаратов подтверждают высокую точность и высокую надежность разработанной фотограмметрической системы для захвата 3D-движения.

Литература

- [1] *Yabukami S., Kikuchi H., Yamaguchi M., Arai K.I., Takahashi K., Itagaki A., Wako N.* Motion capture system of magnetic markers using three-axial magnetic field sensor. // *Magnetics*, 2000. – Vol.36, №.5, – pp.3646–3648.
- [2] *Moeslund T.B., Hilton A., Kruger V.* A survey of advances in vision-based human motion capture and analysis. // *Computer Vision and Image Understanding*, 2006. – Vol.104, pp.90–126.
- [3] *Князь В.А.* Scalable photogrammetric motion capture system “Mosca”: development and application // *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XL-5/W6, 43-49, doi:10.5194/isprsarchives-XL-5-W6-43-2015, 2015.

Синхронизация распределённого воспроизведения мультимедийных данных на полиэкранах*

Б.С. Долговесов, М.А. Городилов, И.Д. Храмыцов

bsd@iae.nsk.su | gorodilovm@gmail.com | khramtsov9203@ya.ru

Институт автоматизации и электротехники СО РАН, Новосибирск

Статья посвящена решению некоторых вопросов отображения распределённых мультимедийных данных на больших экранах. В частности, рассмотрены проблемы синхронизации процесса параллельной обработки и вывода фрагментов видеозображения на соответствующие экранные модули полиэкранной системы отображения. Предложен алгоритм синхронизации и его программная реализация с использованием графических ускорителей, обеспечивающий визуальную непрерывность динамических сцен при отображении на полиэкранах.

Ключевые слова: полиэкранный экран, синхронизация, мультимедийные данные, распределённый рендеринг.

Synchronization of distributed data multimediyh play split screen*

B.S. Dolgovesov, M.A. Gorodilov, I.D. Khramtsov

Institute of Automation and Electrometry of the Siberian Branch of the RAS, Novosibirsk, Russia

When developing visualization systems based on tiled display walls it's crucial to provide seamless video playback across all individual tiles of the display wall. This paper proposes a GPU based software approach to solving the problem of inter-tile synchronization. The proposed approach doesn't require usage of specialized hardware, allows to effectively utilize frame buffering mechanism and thus is more resistant to irregular frame input rates than some other existing approaches to synchronization.

Keywords: tiled display walls, video synchronization, distributed rendering.

Введение

В настоящее время актуальными являются решения для высококачественного отображения данных на экранах большого размера в многофункциональных мультимедийных системах для различных приложений. В одних приложениях это обуславливается необходимостью одновременной демонстрации информации большому числу пользователей (видеоконференции, ситуационные центры, спортивные мероприятия), в других – высокими требованиями к разрешению изображения (системы визуализации научных данных, интерактивные обучающие системы). Распространённым решением для этих целей является использование полиэкранов – набора из нескольких экранных модулей (плазменных панелей, мониторов или проекторов), составляющих один большой экран.

Для вывода изображений наряду с применением специализированных устройств, используются видеопроцессоры на основе персональных компьютеров (ПК). Из-за высокого разрешения, необходимого для генерации выходного изображения полиэкрана, ресурсов одного ПК в большинстве случаев оказывается недостаточно. Самым распространённым решением этой проблемы является использование распределённого рендеринга – подхода, когда множество объединённых в сеть ПК, образующих вычислительный кластер, параллельно обрабаты-

вают отдельные фрагменты общего изображения с последующим выводом их на соответствующие модули полиэкрана. Однако без дополнительной синхронизации различные ПК кластера будут отображать кадры в разное время из-за непостоянных задержек в передаче сетевых сообщений, а также различий в нагрузке на процессор и заполненности буфера видеокарты. Кроме того, различия по частоте и фазе развёртки экранных

модулей также несут трудно контролируемые временные погрешности. Это приводит к тому, что видеокарты, являющиеся частями общего изображения, будут отображаться на соответствующих экранных модулях в разное время. Такие различия во временах отображения кадров приводят к нарушению визуальной непрерывности изображения и отрицательно сказываются на субъективном восприятии видео зрителем. Минимальное значение разницы во времени кадров, приводящее к такому негативному эффекту, зависит от динамичности воспроизводимого видео и конфигурации полиэкрана и составляет от 15 до 30 мсек [1].

Предлагаемый в данной статье метод синхронизации видеопотоков в вычислительном кластере позволяет обеспечить визуальную непрерывность изображения на полиэкранной системе без использования дорогостоящих специализированных аппаратных средств.

Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347

Синхронизация видеоданных для полиэкранного отображения

Обзор существующих подходов. Существующие подходы синхронизации видеопотоков для полиэкранных систем можно разделить на программные и аппаратные. Аппаратные решения используют специализированные внешние устройства для синхронизации развёртки экранных модулей или обновления данных в буфере графических акселераторов. Программные решения реализуют тот или иной протокол синхронизации передачи данных между компьютерами кластера и времени их обработки. Это позволяет снизить различия во времени отображения кадров.

В качестве примеров аппаратных решений можно рассмотреть устройства синхронизации *ATI FirePro S400* и *Nvidia Quadro Sync* со схожими характеристиками, объединяющие каждый до четырёх графических ускорителей на одном ПК с возможностью синхронного вывода данных, в том числе и на полиэкранный. Данные модули совместимы лишь с весьма ограниченным набором моделей специализированных и дорогостоящих видеокарт.

Пример программного решения – алгоритм синхронизации для полиэкранного отображения на базе масштабируемой адаптивной графической среды *SAGE* (*Scalable Adaptive Graphics Environment*) [2]. Алгоритм использует два варианта синхронизации: с помощью сетевого барьера и с помощью синхронизации времени работы компьютеров. В первом варианте все компьютеры, входящие в состав кластера, осуществляют ожидание на сетевом барьере перед выводом данных на соответствующие экранные модули.

Когда все части изображения готовы и потоки данных на всех компьютерах доходят до барьера, происходит одновременный вывод изображений на все экранные модули полиэкрана. При таком подходе, в случае возникновения задержки на одном из компьютеров кластера, происходит задержка в работе всей системы. Во втором варианте время на всех компьютерах кластера синхронизируется с помощью протокола *NTP* (*Network Time Protocol*). Один из компьютеров является главным – на каждом новом кадре он рассылает всем остальным компьютерам время, в которое они должны вывести свою часть изображения на экранный модуль. Это время вычисляется путём прибавления к текущему времени главного компьютера некоторой эмпирически задаваемой константы (*Presentation Time offset, PTO*), неудачный выбор которой может отрицательно повлиять на работу алгоритма. Малое значение *PTO* приведёт к тому, что некоторые компьютеры могут не успеть отобразить свои данные на экранном модуле к назначенному времени, а

слишком большое приведёт к излишнему их простоям. В приведённых в статье [2] результатах тестирования максимальная рассинхронизация времени кадров находится в пределах 1-2 мсек. Однако, оба варианта этого алгоритма применимы только в том случае, если используют предположение, что новые кадры поступают на компьютеры равномерно и без потерь и не пропускаются кадры при отображении. Подобные методы применяются и в других межплатформенных программных средствах для построения распределённых систем визуализации, например, *Chromium* и *Equalizer*. Данные программные продукты предоставляют сетевые барьеры, как основной примитив синхронизации кадров при полиэкранном отображении и, следовательно, имеют те же основные недостатки, что и в алгоритме, представленном в работе [2].

Предложенный алгоритм синхронизации. Вданной работе не рассматривается распределение видеоданных по компьютерам кластера. Если проигрывается видеофайл, считается, что каждый компьютер получает копию видео (либо только интересующий его фрагмент видео) перед началом воспроизведения. Для потокового видео (например, трансляции с видеокamеры) считается, что кадры разбиваются на фрагменты или передаются целиком в реальном времени.

Основная идея предлагаемого в данной статье алгоритма синхронизации заключается в том, что компьютеры кластера осуществляют буферизацию входящих кадров, затем на основе текущей заполненности буфера графического процессора определяется расчётное время, в которое очередной кадр будет отображен на экране. Важно отметить, что кадровая частота выходного видеопотока при этом совпадает с частотой вертикальной развёртки экранного модуля (режим вертикальной синхронизации). Рассмотрим подробно цикл работы алгоритма для одного компьютера кластера и одного нового кадра.

Все новые кадры предварительно загружаются во входной буфер (*FIFO*-очередь). У каждого кадра из этого буфера есть временная метка T_i , определяющая, когда кадр должен быть отображён. Но, поскольку практически во всех современных графических приложениях используется двойная или тройная буферизация при рендеринге изображения, то даже при отправке графическому процессору команды «Отобразить» для нового кадра, он не отобразится на экране в этот момент. Сначала он поступает в один из вторичных буферов (*back buffer*) и лишь после того, как все предыдущие кадры будут отображены, его буфер станет первичным (*front buffer*), и новый кадр будет отображён на экране. Таким образом, кадр, отправленный на отображение в момент времени t на самом деле бу-

дет отображён в момент времени

$$t' = t + \Delta T \cdot k + dt$$

где ΔT – длительность одного кадра (в данном случае совпадающая с периодом развёртки экранного модуля), k – количество кадров в буфере графического процессора, а dt – фазовая составляющая, определяющаяся разницей между временем поступления запроса на отображение и началом следующего цикла развёртки монитора. Поскольку заполненность буфера графического процессора может зависеть от текущей нагрузки и значительно варьироваться в разные моменты времени на разных компьютерах, без дополнительной коррекции время отображения кадров также будет различаться более чем на ΔT . Это составляет около 16 мсек, для типичного монитора с частотой обновления 60 Гц.

Таким образом, предлагается, используя данные о времени отображения предыдущих кадров и пропорционально-интегральный регулятор (ПИ-регулятор), определять момент времени, в который новый кадр будет в действительности отображён на экране. Этот момент времени (оценочное время) вычисляется, как:

$$T'_i = T'_{i-1} + \Delta T + dt_k$$

где T'_{i-1} – (время отображения предыдущего кадра),

оценка того же момента для предыдущего кадра (для первого кадра – текущее системное время узла),

ΔT – период вертикальной развёртки монитора, а dt_k – корректировочный коэффициент. В процессе функционирования алгоритма хранится информация о последних N_{hist} кадрах: оценочное (T'_i) и реальное (T_i) времена отображения. С использованием этой информации значение коэффициента dt_k вычисляется как средняя разница между реальным и оценочным временем, умноженная на настроечный коэффициент K :

$$dt_k = K \frac{1}{N_{hist}} = \sum_{i=1}^{N_{hist}} (T_i - T'_i)$$

До тех пор, пока хотя бы один из вторичных буферов графического процессора свободен для записи, компьютер кластера принимает решение о том, какой кадр отправить на отображение следующим. Компьютер сравнивает текущее время t с оценочным временем T'_k первого кадра во входной очереди.

Если $t < T'_k$ или входная очередь пуста, то будет повторно выбран последний отображённый кадр, иначе – первый кадр будет удалён из очереди и отправлен на отображение.

Программная реализация алгоритма синхронизации. Для проверки работы алгоритма был разработан прототип системы синхронизации видеопотоков для отображения на полиэкране. За основу был взят мультимедийный фреймворк *Microsoft Media Foundation*. Поскольку его архитектура основана на слабо связанных компонентах, формирующих граф потоковой обработки медиаданных. Это позволяет разместить логику синхронизации процесса отображения в одном независимом модуле. В качестве такого модуля был создан компонент-приёмник (sink) на основе *DirectX11 API*.

В предлагаемой архитектуре среди всех распределённых компьютеров кластера один выполняет роль *master*-компьютера, который управляет воспроизведением и является источником синхронизированного времени для остальных *slave*-компьютеров (ведомых). Временная синхронизация *начала, паузы* и *остановки* воспроизведения медиаданных между распределёнными компьютерами кластера осуществляется путём рассылки *master*-компьютером управляющих сообщений всем *slave*-компьютерам. Синхронизация времени реализована программно согласно протоколу *PTP (Precision Time Protocol)*, что позволило достичь точности синхронизации часов на распределённых компьютерах кластера менее 1 мсек. Информация о времени отображения видеокладов, требуемая для реализации алгоритма, определяется средствами *DXGI 1.2*, включающими интерфейс *IDXGISwapChain* для доступа к очереди буферов графического процессора и метод сбора статистики кадра — *GetFrameStatistics*. Исходные данные о временах отображения кадров поступают вместе с кадрами от вышележащих компонентов графа воспроизведения через интерфейс *IMFSample*.

Тестирование алгоритма синхронизации

Разработанный прототип системы синхронизации апробирован на системе из двух компьютеров, соединённых через локальную сеть *Ethernet* и представляющих вычислительный кластер. Один из них исполнял роль *master*-компьютера, другой – *slave*-компьютера. Тестовый видеофайл был заранее разбит на фрагменты и размещён на соответствующих компьютерах.

За основную характеристику, определяющую качество синхронизации изображения, была выбрана разница между временами воспроизведения компьютерами видеокладов, являющихся частями общего кадра изображения полиэкрана. Чем ближе эта величина к нулю, тем лучше синхронизованы изображения на отдельных мониторах.

Для проверки влияния неравномерности поступления входных кадров на работу системы синхрони-

зации была введена периодическая задержка в модуль, осуществляющий приём кадров, т.е. каждые N кадров работа процесса приостанавливалась на T мсек. Таким образом была смоделирована ситуация, когда один из компьютеров кластера работает с переменной задержкой относительно других, либо когда кадры изображения поступают на компьютеры неравномерно. Размер входной очереди кадров и количество вторичных буферов графического акселератора были выбраны равными четырём.

Тестовый видеофайл с частотой 30 кадров/сек. был последовательно воспроизведён в трёх вариантах: без буферизации, с буферизацией без какой-либо коррекции времени кадров и с буферизацией с коррекцией времени кадров согласно алгоритму, описанному в секции 2.2.

В первом варианте буферизация кадров при выводе на экран не использовалась. Каждый компьютер, находясь в режиме ожидания, с приходом момента времени для очередного кадра отправляет его на отображение. Как показывают результаты измерений, возникающая при задержках разница во временах кадров совпадает с величиной задержки T (рис. 1).

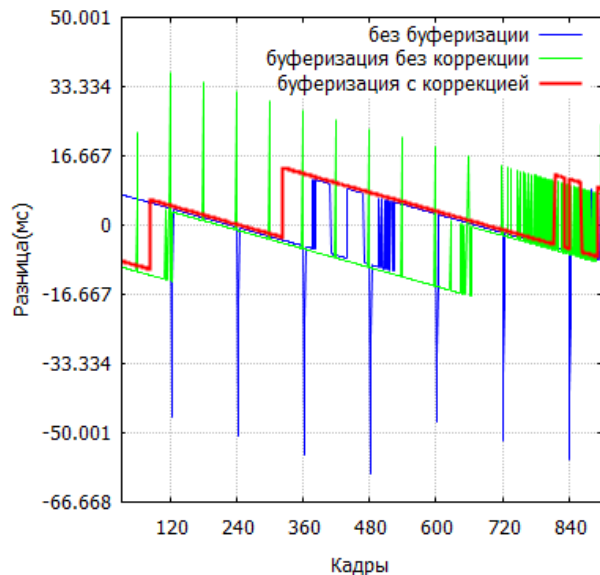


Рис. 1: Разница во времени кадров.

Во втором варианте при выводе кадров была использована буферизация с 4 вторичными буферами. Работа компьютеров осуществлялась по алгоритму, описанному в секции 2.2, за исключением того, что заполненность буфера не учитывалась при принятии решения в какой момент времени отправлять очередной кадр на отображение. В данном случае использование буферизации позволяет снизить разницу во временах воспроизведения, воз-

никающую из-за задержек поступления кадров, до 30 мсек.

В третьем варианте, как и в предыдущем, используется буферизация кадров с теми же параметрами. В данном варианте разница находится в пределах ± 16 мсек (при той же величине задержки, что и в варианте 2), что для монитора с частотой развёртки 60 Гц составляет менее одного кадра.

Заключение

Предложенный алгоритм синхронизации распределенного процесса подготовки видеоданных для полиэкранных средств отображения позволяет обеспечить визуальную непрерывность выходного изображения без использования специализированных аппаратных средств. В отличие от аналогичных программных решений он также позволяет сохранить преимущества двойной/тройной буферизации при распределённом рендеринге и устойчив к неравномерности частоты поступления кадров, благодаря буферизации кадров на входе и на выходе. Размер буфера является настраиваемой величиной – чем он больше, тем выше устойчивость системы к задержкам в поступлении кадров, но и тем больше задержка между их выводом на полиэкранный экран. Алгоритм не накладывает ограничений на число компьютеров в кластере, поскольку вычислительные затраты на синхронизацию на одном компьютере составляют незначительную часть (менее 1%) от общего объёма вычислений и не увеличиваются с ростом числа компьютеров.

Литература

- [1] *Deshpande S., Daly S.* Quality of Experience for Large Ultra-High-Resolution Tiled Displays with Synchronization Mismatch // *EURASIP Journal on Image and Video Processing*, 2011. <http://jivp.eurasipjournals.com/content/2011/1/647591>.
- [2] *Nam S., Deshpande S., Vishwanath V., Jeong B., Renambot L., Leigh J.* Multi-Application Inter-Tile Synchronization on Ultra-High-Resolution Display // *Wall*: <http://www.mcs.anl.gov/papers/P1678.pdf>

Об авторах

Долговесов Б. С. – к.т.н., заведующий лабораторией синтезирующих систем визуализации института автоматки и электротметрии СО РАН.
e-mail: bsd@iae.nsk.su

Городилов М. А. – м.н.с. лаборатории синтезирующих систем визуализации института автоматки и электротметрии СО РАН.
e-mail: gorodilovm@gmail.com

Храмцов И. Д. – инженер-программист лаборатории синтезирующих систем визуализации института автоматки и электротметрии СО РАН.
e-mail: khramtsov9203@ya.ru

Разработка супервизорной системы удаленного управления коммутацией медиапоточков*

М.А. Городилов, Б.С. Долговесов, А.Х. Радостев

bsd@iae.nsk.su | gorodilovm@gmail.com | dargerog@mail.ru

Институт автоматизации и электрометрии СО РАН, Новосибирск, Россия

В статье рассматривается способ пересылки по сети превью произвольного количества медиапоточков с возможностью прямого контроля над ними в реальном времени и с постоянной нагрузкой на сеть.

Ключевые слова: *медиапоток, удаленное управление, мультимедийные системы, коммутация.*

Development of a supervisory system for remote control switching media streams*

M.A. Gorodilov, B.S. Dolgovesov, A.J. Radostev

Institute of Automation and Electrometry of the Siberian Branch of the RAS, Novosibirsk, Russia

Describing a method of sending arbitrary amount of video previews over network providing ability to control their resolution in real time with limited network load.

Keywords: *media stream, remote control, systems, video switching.*

Введение

Развитие научно-технологической базы в сфере мультимедийных данных позволяет постепенно переходить от решения частных задач к более универсальным подходам. В частности, к новым технологическим решениям для организации многоканального ввода и демонстрации аудиовизуальных данных с помощью специализированных систем управления, ввода, мониторинга входных данных, коммутации и вещания. В подобных распределенных системах отображения [1] возможность управления вычислительными узлами не всегда возможна локально. Рассматривается ситуация, когда необходим удаленный контроль над коммутатором медиапоточков.

Предлагается способ, позволяющий гарантировать постоянство количества используемого трафика при передаче по сети произвольного количества превью видеопотоков с возможностью управления разрешением отдельных превью.

Традиционные решения

Для оперативного управления системой отображения удаленный оператор должен иметь возможность просмотра данных от всех входных источников и результата их микширования. Многие аппаратные решения предоставляют подобную возможность.

Одним из подходов является передача одного видеопотока с возможностью переключения. Выбирается один просматриваемый видеопоток, который интересует удаленного оператора, и передается по сети [2].

Достоинством метода можно считать хорошее качество передаваемого изображения. С другой стороны, переключение на другой источник данных занимает время. Также, недостатком является отсутствие возможности одновременного просмотра предварительных данных нескольких источников. Просмотр видеоданных одновременно возможен, когда исходные данные всех источников передаются с низким разрешением [3]. Среди недостатков: нагрузка на сеть зависит от количества источников, потенциально делая её непостоянной; отсутствует синхронизация первичных данных между собой; нет возможности увеличить разрешение отдельного видеопотока; удаленному клиенту нужно декодировать несколько видеопотоков в реальном времени, что накладывает ограничение на производительность устройства удаленного оператора.

Предлагаемое решение

Так как вычислительные ресурсы компьютера оператора ограничены, то невозможно одновременно обрабатывать видеоданные со всех входных источников в исходном разрешении для отображения на удаленном устройстве управления. Для оптимального решения данной задачи предлагается формировать на отдельном вычислительном узле видеопоток, кадры которого содержат композицию видеоокон управляемых потоков в уменьшенном масштабе, и передавать его в сжатом виде по локальной сети. На компьютере оператора этот поток разжимается и разделяется на отдельные видеопотоки, в соответствии с информацией о расположении исходного видео в кадре. При этом поток данных о расположении превью должен быть синхронизирован с видеопотоком от сервера. Требуемое разрешение для превью, отображаемого на пользовательском интерфейсе, устанавливается оператором

Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347

и передается вычислительному узлу, отвечающему за упаковку видеоокон.

Процесс передачи одного кадра композиции превью видеопотоков включает следующие операции:

- прием от компьютера оператора информации о размере превью и их позиции на экране оператора;
- генерация композиции превью и дополнительной информации о расположении и размерах видеоокон;
- кодирование кадра с композицией видеоокон;
- отправка сжатых данных по сети посредством сетевого протокола на компьютер оператора и их декодирование;
- интерпретация дополнительной информации компьютером оператора.

Детали реализации

Для реализации метода разработан алгоритм, формирующий графические окна в соответствии с параметрами, определяющими размер и позицию превью на экране интерфейса. Окна помещаются в видеоконтейнер по очереди в порядке убывания размера. При сравнении размеров окон и видеоконтейнеров используется функция сортировки, учитывающая соотношение сторон исходного видеоконтейнера. При этом должно выполняться условие — окно, не помещающееся в видеоконтейнер, будет уменьшено до необходимого размера, сохраняя соотношение сторон. В результате работы алгоритма формируется набор окон, содержащих координаты и размеры кадров превью для отображения. Эти данные передаются в видеоконтейнер в качестве дополнительной информации.

Результаты тестирования

Для тестирования алгоритма формирования потока кадров фиксированного размера, содержащего мозаику кадров изображений всех источников, использовались: библиотека FFMpeg, формат видеоконтейнера MP4, видеокодек H264, протоколом передачи данных RTMP. Для передачи дополнительной информации о мозаике кадров использовался отдельный поток данных. Для реализации удаленного вызова процедур использовался протокол JSON-RPC.

Свободное место в видеоконтейнере. Так как при упаковке графических окон сохраняется соотношение сторон, использование всего доступного свободного места в видеоконтейнере бывает невозможным.

Была проведена серия статистических тестов количества оставшегося свободного места в видеоконтейнере. В контейнер размером 1920×1080 упаковывались прямоугольники с соотношением сторон

4:3 либо 16:9. Было выявлено, что алгоритм обеспечивает использование не менее 75% места в контейнере. На рис. 1 приведен график одного из таких тестов.

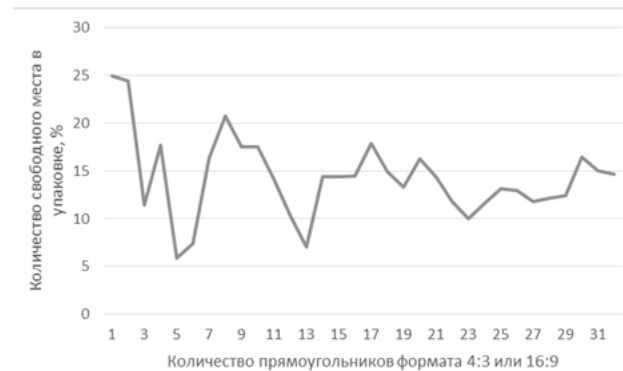


Рис. 1: Пример графика зависимости количества свободного места в видео контейнере от количества упакованных окон.

Графические артефакты при упаковке. Проведена серия тестов графических искажений при упаковке графических окон. Для этого была использована величина пикового соотношения сигнала к шуму (PSNR) [2]. Пример подобного теста представлен на рис. 2. Можно отметить, что среднее значение величины PSNR находится в районе 50 Db.



Рис. 2: График пикового соотношения сигнала к шуму для одного видеопотока, упакованного предложенным методом.

В результате тестирования выявлено, что на границах стыковок кадров при динамических изменениях присутствуют артефакты в виде небольших искажений квадратной формы (не более 5Db). Это обусловлено тем, что кодек H264 кодирует информацию в изображении небольшими блоками (например, 16×16 пикселей). В таком случае, если сделать размеры кадров изображения кратными 16 пикселям, можно добиться исчезновения данного искажения.

Следует отметить, что разница в значении PSNR, вызванная этим искажением, незначительна и составляет порядка 1 Db.

Заключение

Для удаленного управления отображением видеоданных предложен эффективный метод организации удобного пользовательского интерфейса. Для отображения превью интерфейса, соответствующих входным источникам данных, формируется ограниченный поток данных для передачи по сети, не зависящий от количества источников. Метод обеспечивает в реальном времени предварительный просмотр входных источников и оперативный вывод требуемых видеоданных для отображения на полиэкранных средствах. Кроме того, данный подход не накладывает существенных ограничений на производительность компьютера оператора и не требует дополнительных аппаратных средств, что позволяет использовать планшетные компьютеры.

Литература

- [1] Морозов Б.Б., Долговесов Б.С., Мазурок Б.С., Городилов М.А. Построение распределенной мультимедийной виртуальной среды с многоканальной визуализацией медиаданных на графических акселераторах // Программирование. – 2014. №4. – С.55–63.
- [2] Gibson J. Concurrent autonomous video-lecturing system applicability study // Theseus, Helsinki Metropolia University of Applied Sciences. – 2013.
- [3] Binotto A.P.D. A Real-time Collaborative Teleultrasonography // Potentials, IEEE (Volume 31, Issue 6). – 2012.
- [4] Wikipedia: Peak signal to noise ratio // https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio.

Об авторах

Долговесов Борис Степанович – кандидат технических наук, заведующий лабораторией синтезирующих систем визуализации ИАиЭ СО РАН.
Его адрес: bsd@iae.nsk.su

Городилов Михаил Андреевич – научный сотрудник лаборатории синтезирующих систем визуализации, инженер-программист.
Его адрес: gorodilovm@gmail.com

Радостев Александр Хосе – магистрант НГУ.
Его адрес: dargerog@mail.ru

Система обработки и отображения визуальной информации для тренажерных комплексов*

Б.С. Долговесов, М.А. Городилов, Ф.В. Фатьянов

bsd@iae.nsk.su | gorodilovm@gmail.com | Fedor@graphica.com

Институт Автоматики и Электростроения СО РАН, Новосибирск, Россия

В данной работе рассмотрен вариант построения системы отображения, в которой при большом количестве входных мультимедийных данных используется программная реализация обработки, декодирования и визуализации медиаданных с использованием современных графических акселераторов.

Ключевые слова: *система отображения, микшер-коммутатор, интерфейс управления.*

The system of processing and display of visual information for simulators*

V.S. Dolgovesov, M.A. Gorodilov, F.V. Fatyanov

Institute of Automation and Electrometry SB RAS, Novosibirsk, Russia

Describing a method of constructing visualization systems that uses software implementation based on modern graphical processor units resources for processing, decoding and visualization in case of significant amount of input multimedia data.

Keywords: *: visualization system, mixer-switcher, control interface.*

Введение

Быстрый рост потребностей как по количеству одновременно отображаемых источников мультимедийных данных, так и по разнообразию конфигураций входных источников данных обуславливает переход от частных решений к универсальным подходам в создании систем обработки и отображения визуальной информации, предназначенных для мониторинга. Актуальной является задача не только создания новых систем, но и модернизации уже существующих. В подобных системах используются видеокamеры стандартной четкости, сигнал от которых передается аналоговым способом по коаксиальному кабелю. В таких случаях требуется сохранить часть имеющегося оборудования, и добавить видеокamеры высокой четкости и другие источники видео. Кроме того, необходимо обеспечить возможность дальнейшего расширения и модернизации системы, путем добавления новых видео источников и замены старых. Требуется так же решать задачу по формированию и выводу на экраны «мозаик» - комбинированного изображения источников видеосигналов произвольного размера и расположения. Одно из решений - использование матричного коммутатора [1] вместе с внешними графическими процессорами. Поскольку финальное изображение зависит сразу от нескольких модулей, требуется создавать дополнительную систему интерактивного управления, которая должна управлять сразу всеми разнородными устройствами. Крайне сложно решить проблему синхронности смены варианта мозаики и коммутации сигналов, потому что время исполнения команды не детерминировано. Кро-

ме того, потребуется использовать дорогостоящие матричные коммутаторы, позволяющие заменять модули ввода, что необходимо для выполнения требования по гибкости системы.

В работе [2] предложен подход к построению подобных систем на базе современных графических акселераторов, позволяющих разработчику программно реализовать различные функции обработки графических данных, существенно упрощающих традиционные аппаратные решения, используемые при визуализации мультимедийных данных. В данной работе рассмотрен пример системы, построенной по предложенным принципам.

Структура системы

Рассмотрим данный подход в применении к решению задачи модернизации телевизионной аппаратуры комплекса тренажеров Российского сегмента Международной космической станции.

В данной системе используются восемь аналоговых камер стандартного разрешения и система компьютерной генерации изображений, моделирующая визуальную обстановку в приборах и средствах наблюдения тренажера. При этом изображения с системы компьютерной генерации отображаются на отдельном мониторе и изменение коммутации во время работы комплекса невозможно. Требуется дополнить эту систему тремя видеокamерами высокой четкости. Необходимо также заменить несколько отдельных мониторов для вывода одним большим или несколькими объединенными в видеостену, куда будет выводиться мозаика из выбранных источников. Управление настройками мозаики должно осуществляться удаленно с помощью отдельной рабочей станции или мобильного планшетного устройства.

Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347.

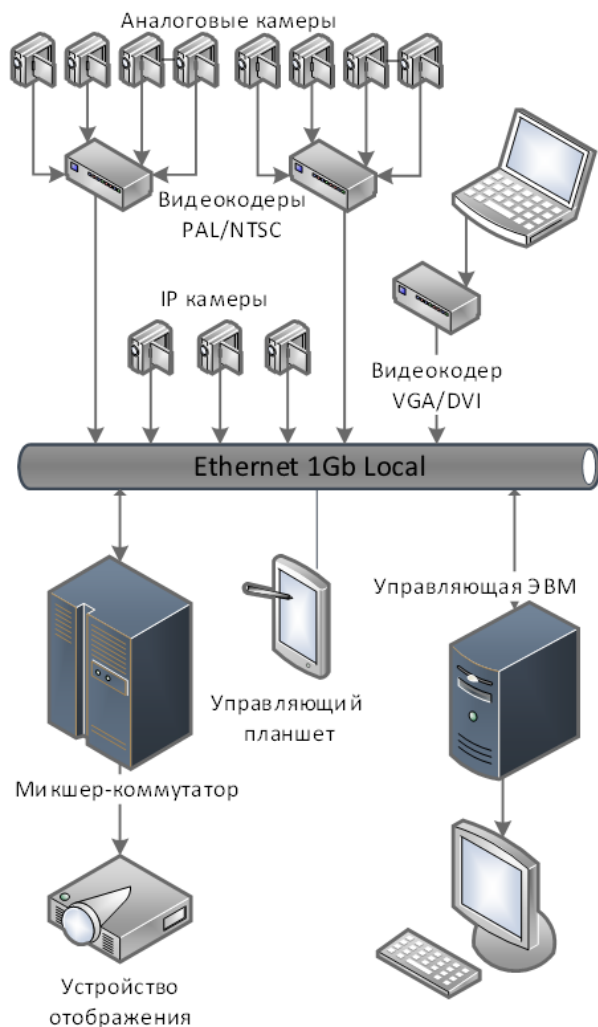


Рис. 1: Структура системы.

В целях минимизации стоимости системы и уменьшения количества точек коммутации, что требуется для повышения надежности системы, в качестве дополнительных камер используются IP-камеры. Для ввода мультимедиа данных с других источников используются модули кодирования, поддерживающие соответствующий формат сигналов. Основным элементом системы является программный «микшер-коммутатор». На рисунке 1 представлены модули системы и схема коммутации между ними. Современные графические акселераторы позволяют осуществлять обработку сразу нескольких видео потоков чересстрочного формата в реальном времени, что дает возможность масштабировать такие видео без искажений. Использование стандартных аппаратных средств позволяет снизить стоимость системы и повысить её ремонтпригодность за счет широкого распространения аппаратных компонентов и их взаимозаменяемости. Программное решение позволяет организовать удаленное управление с возможностью одновременно

го предпросмотра видео источников и результата микширования, которое может осуществляться с управляющей ЭВМ или планшета. При увеличении числа входных источников система может быть дополнена несколькими микшерами-коммутаторами. Каждый из которых может выводить изображение на один из мониторов, образующих видеостену. Даже в таком случае интерфейс управления будет оставаться прежним сохраняя простоту управления и настройки комплекса.

Реализация программной части

Для реализации системы используется среда разработки «Qt Quick», в состав которой входит декларативный язык QML. QML используется для описания связей между модулями системы и основной логики приложений. Несмотря на то, что в состав Qt Quick входит достаточно большой набор модулей, не все подзадачи можно реализовать, используя базовые элементы встроенных библиотек. Поэтому были реализованы модули расширения на языке C++.

Воспроизведение и вывод медиа данных осуществляется средствами мультимедийного фреймворка Microsoft Media Foundation. Поэтому были реализованы модули расширения для построения конвейера обработки мультимедийных данных из модулей Media Foundation и доступа до некоторых объектов конвейера из среды исполнения QML. Передача видео данных из этого конвейера осуществляется с помощью специального компонента-приёмника, который осуществляет предварительную обработку видео потока и выдает результат в виде ресурса в памяти графического акселератора. Ещё один модуль позволяет отображать этот ресурс в виде графического элемента в QML сцене. Также реализован модуль, позволяющий записывать изображение QML сцены в память графического акселератора, которая будет считываться в компоненте источнике данных внутри конвейера Media Foundation.

Поскольку реализация пользовательского интерфейса находится в отдельном приложении, которое исполняется на удаленном компьютере, был реализован модуль для связи нескольких приложений, использующий среду исполнения QML скрипта. Любое приложение может содержать как несколько управляемых объектов, так и несколько управляющих. При этом с одним управляемым объектом может быть одновременно связано несколько управляющих. Управляемый объект ссылается на один из других объектов внутри своей среды исполнения. Используя мета описание этого объекта можно создать удаленную копию, которая будет перенаправлять вызовы к исходному объекту. Такой подход позволяет работать с удаленными объектами практически так же, как и с локальными.

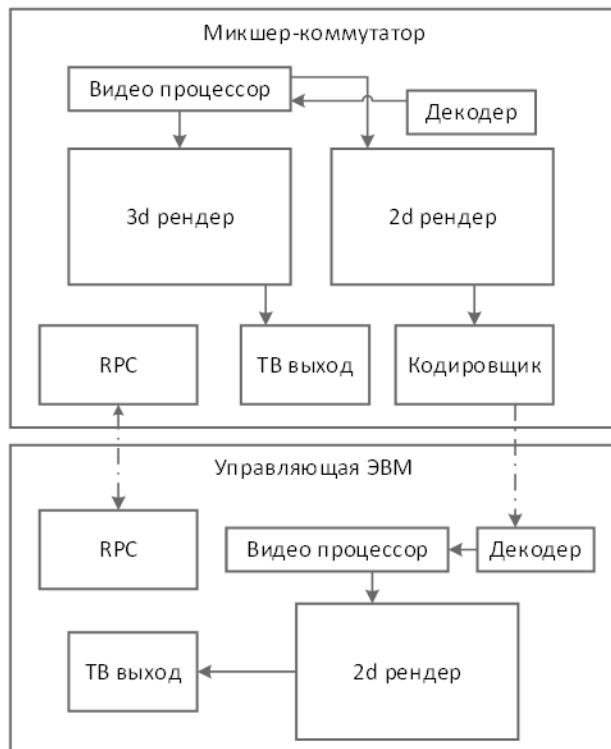


Рис. 2: Схема взаимодействия между программными модулями системы.

При этом не требуется создания дополнительного мета описания объектов. Одним из ограничений является запрет на использования функций с возвращаемым значением. Вместо этого предлагается использовать дополнительные свойства объектов для передачи результата выполнения. На рисунке 2 представлена схема связей между основными программными модулями системы.

Кроме того, был реализован модуль для визуализации 3x мерной сцены и ряд других вспомогательных модулей таких как модуль журналирования и сериализации.

Заключение

В данной работе рассмотрен альтернативный подход, предусматривающий использование компрессированных цифровых данных для передачи в универсальных сетевых средах в комбинации с локальной коммутацией неkomпрессированных сигналов непосредственно на тренажерном месте.

Основы этого подхода изложены в работе [2]. Используемая распределенная среда обеспечивает гибкое конфигурирование и интерактивное управление процессом отображения в любом доступном месте среды. Предлагаемое решение позволяет снизить коммутационные нагрузки по сравнению со схемой «каждый с каждым» типовой для систем с матричным коммутатором. Одним из преимуществ предлагаемого подхода является возможность поэтапного наращивания и совершенствования как системы в целом так и её частей.

Данное решение обеспечит необходимую масштабируемость и дальнейшую модернизацию системы. Помимо этого, использование программного микшера-коммутатора облегчит решение подзадачи по формированию комбинированных изображений с входных источников, так как для этого не требуется использования дополнительных модулей, в отличие от варианта с матричным коммутатором.

Литература

- [1] Ерохин А.В., Фрадкин Б.Г., Левин И.И., Рыжик О.А. Матричный Коммутатор // Патент на изобретение 2103729. Официальный бюллетень Роспатента «Изобретения. Полезные модели», 27.01.1998.
- [2] Морозов Б.Б., Долговесов Б.С., Мазурок Б.С., Городилов М.А. Построение распределенной мультимедийной виртуальной среды с многоканальной визуализацией медиаданных на графических акселераторах // Программирование, 2014, номер 4. С.55-63.

Об авторах

Долговесов Б.С., к.т.н., заведующий лабораторией Синтезирующих систем визуализации в Институте автоматизации и электротехники СО РАН.

E-mail: bsd@iae.nsk.su

Городилов М.А., м.н.с. лаборатории Синтезирующих систем визуализации в Институте автоматизации и электротехники СО РАН.

E-mail: gorodilovm@gmail.com

Фатьянов Ф.В., инженер-программист лаборатории Синтезирующих систем визуализации в Институте автоматизации и электротехники СО РАН.

E-mail: Fedor@graphica.com

Автоматическое создание динамических вставок для спецэффектов видео*

К. Ватаначоте[†], В. Кумара[†], Т.К. Ши[†], А. Клименко[‡]

kanoksak.wattanachote@gmail.com | chinthakawk@gmail.com | timothykshih@gmail.com
andy.klimenko@gmail.com

[†]Факультет информатики и информационных технологий,

Центральный национальный университет, Чжунли, Тайвань

[‡]Институт физико-технической информатики, Протвино, Россия

Изменения в визуализации динамических текстур, приводящие к новому виду наблюдаемого движения и интенсивности определяется как динамическое преобразование текстур. Например, сцену водопада в видеоряде можно проявить в качестве текстуры огня (пожара) или наоборот. Эта статья предлагает новый способ для динамического преобразования текстур между двумя динамическими текстурами в видеоклипах. Целью работы было нахождение методов отображения динамических текстур с различной динамикой и визуализацией в видеоряде. Патч (фрагмент) перехода есть концепция с предположением о том, что две динамические текстуры могут иметь патчи (фрагменты), которые идентичны по внешнему виду, но отличаются по динамике. Патч (фрагмент) перехода состоит из двух модулей, модуль поиска патча (фрагмента) во-первых. Иррегулярной формы патч (фрагмент) используется чтобы найти конгруэнтные участки на исходной и целевой динамической текстуре. Второй модуль отвечает за соответствие патча (фрагмента) перехода результату алгоритма сопоставления шаблонов. Здесь мы предлагаем новый алгоритм сопоставления шаблонов, который допускает соответствия только в Текстуре Интереса (ТОИ), и два метода сегментации в реальном времени для поиска ТОИ. Экспериментальные результаты показывают, что каждый патч (фрагмент перехода) может осуществлять алгоритм поиска с эффективно повышенной точностью соответствия.

Ключевые слова: динамическая текстура, трансформация динамической текстуры, шаблон соответствия, редактирование видео.

Automatic Creation of Dynamic Patches in Special Effect Videos*

Kanoksak Wattanachote[†], W.G.C.W. Kumara[†], Timothy K. Shih[†], Andrei Klimenko[‡]

[†]Dept. of Computer Science and Information Engineering,

National Central University, Zhongli, Taiwan (R.O.C.);

[‡]Institute of Computing for Physics and Technology, Protvino, Russia

Change in dynamic texture visualization to provide a new look both in dynamic appearance and intensity connectivity is given as the definition for dynamic texture transformation. For instance, a waterfall scene in a video can be appeared as a fire texture or vice versa. This paper proposes a novel method for dynamic texture transformation between two dynamic textures in video clips. The goal was to find out techniques to display dynamic texture with varying dynamics and appearances in a video. Patch transfer is a concept with the assumption of two dynamic textures can have a patch that is identical in appearance but different in dynamics. Patch transfer is composed of two modules, the patch searching module being the first. An irregular shaped patch is designed to find out the congruent regions on source and target dynamic textures. The second module is responsible for patch transfer by adapting to the output of the template matching algorithm. Here, we propose a new algorithm for template matching which allows the matching to be occurred only in the Texture of Interest (TOI), and two methods for real-time segmentation to find the TOI. The experimental results show that each patch can realize the matching algorithm with effectively enhanced matching accuracy.

Keywords: dynamic texture, dynamic texture transformation, template matching, video editing.

I. Introduction

Dynamic texture is described as a sequence of images in moving scenes that continuously exhibits movement of intensities in time, such as smoke, fire, water flow, etc. Changes in the intensities and their connectivity derived from other textures were studied to create a new visualization of dynamic textures.

Video editing has become a very important field with the wide adaptation of social media in the Internet and the developments of the film industry allowing users to manipulate their personally captured or other readily available video files in different ways. One aim of this finding was to widen the capabilities of video editing with increasing user experience and satisfaction. With the advances in the field, video editing is now being used in different new dimensions making it further interesting and creative going beyond the traditional

Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347

uses like fixing or restoring the damaged video scene. Nowadays, users are interested in creating their own video both in the production and the post-production process. Specially, in film industry, video editing can drastically reduce the risk of engaging in very dangerous outdoor acts by the actors themselves like with fire, smoke, water flood, wind storms, etc.

Dynamic texture segmentation methods described in Related Work section are accurate and robust, but not fast enough for real-time segmentation for our purpose. Hence, dynamic texture segmentation methods by color filtering and connected component filtering approach are proposed in this study. This paper studied the appearance pattern of a connectivity set and its motion on dynamic texture, based on a simple model of appearance matching with high accuracy measured by correlation coefficient statistics.

Section II discusses the related research works for this paper. Section III describes the algorithms for patch searching, while dynamic texture segmentation is devoted in section IV. In the section V patch transfer and corresponding algorithms are discussed. The experimental results of our study are demonstrated in section VI. Finally conclusions and future work are discussed in sections VII.

II. Related work

Aforementioned, at present video editing is an attractive research topic both in academics and industry leading to several purposes. For instance, video completion is one which has a powerful capability in fixing and restoring damaged videos to make the completed video natural and visually pleasant for human eyes. In recent years, several methods have been proposed for video editing.

In [1], Tsai *et al* proposed a video editing using motion inpainting which allows users to change the dynamic texture in video background. Dynamic texture such fire, smoke, water etc., could be edited by using patches in irregular shaped blocks to reproduce a realistic dynamic background [1]. In this vein, dynamic texture synthesis is a method to generate the dynamic background in the video. The synthesized background can be used to fill the background hole as proposed by Lin and Cheng [2]. In addition, editable dynamic textures [5] proposed by Doretto and Soatto serve a different purpose of video editing by synthesizing the dynamic texture to create the realistic sequences of images of dynamic scenes. Nonetheless, there is no existing method proposed for changing dynamic texture appearances.

Inspired by the works and experiments of Doretto *et al* [6], the changing of texture was proposed to study in two approaches. The first one is the changing of appearance, and the other one is the changing of dynamics. Consequently, dynamic texture transformation is

studied in this paper to find the appropriate methods for changing dynamic textures which can be different in texture appearance and dynamics, between two video clips.

Further, to increase the performance, accuracy of changing and subjective quality assessment by human eyes, dynamic texture segmentation is inescapably considered in this paper. The following researches demonstrate various aspects of dynamic texture segmentation. In [3], Fazekas *et al* proposed a method for dynamic texture segmentation by using a level set scheme which three possible alternatives of gradient constancy, color constancy and brightness conservation were examined which aim to overcome the segmentation both on weak and strong dynamic texture. In [8], Phillips III *et al* combined color and motion information computed from video sequences to locate fire texture. Wang *et al* [9] proposed fire frame segmentation tool based on the Hue, Saturation and Intensity (HSI) color space model. However, the segmentation method in our purpose does not deal only fire frame, but also includes other dynamic textures (e.g., waterfall and smoke). The visual perception of the variation in HSV has been implemented by O'Malley *et al* [10], and Mazzeo *et al* [11], in different purposes, and has been found that the advantages of using HSV for visual perception are faster and easier in filtering both static and dynamic texture in video. Hence, in this study, we adopt color filtering method that is generally implemented based on the HSV color space, for the patch searching.

We propose the patch matching algorithms by adapting template matching whose various matching metrics are implemented to measure the minimal different intensity of patches. The concept of traditional template matching as found in Ouyang *et al* [1] and Delalandre *et al* [13] indicate that most matching algorithms are generally used in classifying an object by comparing portions of images with another image, and generally use Sum of Squared Differences (SSD) as the function for distance measurement.

III. Patch Searching

Patch searching is processed on a frame in the source video. A difficulty is finding out an irregular shaped patch on the dynamic texture. To overcome this difficulty, we applied connected component filtering approach to determine the irregular shaped patch and integrated the brightness difference value parameters to allow the user to arrange the size of the patch.

A. Connected Component Model.

In video processing, we describe the video as a volume of sequence images, represented by Φ^3 . A patch is a region of connected pixels within the component, described by Ω^2 which can be found in an image or "frame" in the video denoted as Φ^2 , where $\Phi^2 \in \Phi^3$

and $\Omega^2 \in \Phi^2$. A pixel is described by I as a matrix consists of three components of intensity as in (1). Each of whose is the intensity of the independent red, green and blue components. The luminance is weighted sum of three corresponding linear intensity values as in (2), described as a grayscale value (I_y).

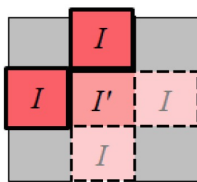
$$I = [I_r, I_g, I_b]^T \quad (1)$$

$$I_y = 0.299 \times I_r + 0.587 \times I_g + 0.114 \times I_b \quad (2)$$

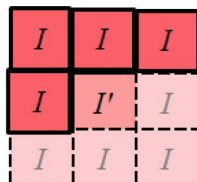
The components in our proposed method are considered in grayscale values range. Hence I and I'_p in the following description mean in grayscale values range calculated by (2). Let I_s represents a given pixel to find out a patch called “seed”, obviously $I_s \in \Omega^2$. I'_p represents a component pixel of Ω^2 , $I'_p \in \Omega^2, p = 0, 1, 2, \dots, k$. A given I_s is considered to be the first component of patch, $I_s = I'_0 \in \Omega^2$. Connected component is a concept of finding out a region of connected pixels starting from a given pixel I_s where all pixels within a component set have close values. Let L is a maximal lower brightness difference between the currently observed pixel I and the component I'_p . Let U is a maximal upper brightness difference between the currently observed pixel I and the component I'_p , where $\{L, U\} \in \mathbb{R}$, floating range, and I represents a pixel at (x, y) in Φ^2 . The pixel I is considered to belong to the component, $k + 1$, if its value on floating range meets the following conditions for any p of k components.

$$I'_p - L \leq I \leq I'_p + U \quad (3)$$

According to the connected component labeling algorithm [14], we prefer two connectivity models for finding the component, as shown in Fig. 1, namely 4-connectivity and 8-connectivity. The 4-connectivity represents the path that is carried out by checking the labels of pixels that are located in the top and left side of I' , whereas the 8-connectivity represents the path that is carried out by checking the label of pixels on the left, top-left, top and top-right side of I' .



(a) 4-connectivity.



(b) 8-connectivity.

Figure 1: The 4-connectivity and 8-connectivity models for two passes connected component labeling algorithm.

The algorithm provides two passes over the frame. The first pass is to record equivalences and assign temporary labels whereas the second pass is to replace

each temporary label by the label of its equivalence class [14].

B. Patch Searching Algorithm.

A patch is derived from a given pixel given by user or system. After a seed point was assigned on the texture, all similar value neighbors are considered to be the component.

According to the concept in Section III - A, patch searching algorithm can be processed as described in the following steps

Step 1: Determine the seed point I_s on the dynamic texture in a video frame.

$$I_s \in \Phi_i^2,$$

where i is a frame number in a video, $i \in \mathbb{N}_0$.

Step 2: Arrange the connected component size by adjusting L and U values, and define the connectivity model by choosing an appropriate connectivity, between 4-connectivity and 8-connectivity.

Step 3: Save a satisfied patch for the upcoming process. Since Ω^2 represents the patch, then lets say:

$$I_s \in \Omega^2 \text{ and } \Omega^2 \subset \Phi_i^2,$$

where $\{\Phi_0^2, \Phi_1^2, \Phi_2^2, \dots, \Phi_i^2, \dots, \Phi_n^2\} \in \Phi^3, n$ denotes the last frame in the video clip.

IV. Dynamic texture segmentation

It is difficult to specify an appropriate segmentation method for dynamic textures due to its discrepancy nature. We propose two real-time segmentation methods for dynamic texture in the following sections.

A. Segmentation by Color Filtering.

Color filtering uses concept of color tracking which is frequently used in computer vision and robotics related applications for fast object tracking and object chasing. Color tracking is done based on the Hue, Saturation and Value or Intensity (HSV) color space. HSV values describe a different color for each basis RGB space. The HSV with emphasis on the visual perception of the variation in Hue, Saturation and Intensity values of an image pixel has been implemented for segmentation by Sural *et al* [7]. The advantages of color tracking are faster results and easier for tracking both static and dynamic texture from the image or video frame.

Hence, this approach is applied for dynamic texture segmentation for creating TOI in our study which aims for real-time segmentation. We modeled the algorithm as follows:

Step 1: Convert frame Φ_i^2 , to the HSV color space.

Step 2: Filter out the result by determining the inclusive lower boundary and exclusive upper boundary in three channels of Hue, Saturate, and Value,

in the integer range e.g., 0–255. Let L represents the lower boundary of each channel, whilst U represents the upper boundary of each channel and use these parameters to filter the elements of Φ_i^2 , by comparing with I component's values in HSV space. Where I is a pixel basis on HSV color space at $(x, y) \in \Phi_i^2$, pixel I will be considered to be a component of $TOI, I \in TOI$, if and only if:

$$L_H \leq I_H < U_H \quad (4)$$

$$L_S \leq I_S < U_S \quad (5)$$

$$L_V \leq I_V < U_V \quad (6)$$

where H indicates the values considered in Hue space (0–255), S indicates the values considered in Saturate space (0–255), and V indicates the values considered in Value space (0–255).

Step 3: Convert TOI to RGB color space, then we get the final result of TOI denoted as T^2 , where $T^2 \subseteq \Phi_i^2$.

To increase the accuracy and performance of patch transfer, there are two purposes of the use of T^2 . In one hand, T^2 is used to increase the accuracy of matching on the right position by limiting patch transfer target only in T^2 . On the other hand, T^2 is also aimed to reduce the duration time for matching method which a ton of statistical metrics are calculated. With the T^2 , the calculation is allowed to be occurred only in the considered region T^2 . Fig. 3 illustrates the path of a match searching algorithm. As a result, this idea leads to our template matching algorithm in Section V.



(a) Original frame. (b) Color filtering result. (c) Connected component result.

Figure 2: The segmentation results comparison, the results derived from the different segmentation approaches.

B. Segmentation by Connected Component.

This segmentation approach is proposed to assert the improvement in segmentation for some dynamic textures which failed in color filtering segmentation approach. For instance in Fig. 2(b), some parts of railway sidewalk were detected with smoke detection purpose.

In order to improve the dynamic texture segmentation for creating TOI , a concept as described in the Section III - B for patch searching algorithm is applied. The target dynamic texture segmentation algorithm for TOI is described here in the following.

Step 1: Determine the seed point I_s on the dynamic texture in a video frame.

Step 2: Arrange the connected component size by choosing an appropriate connectivity in Fig. 1 for two passes algorithm, and then determine L and U values until the red mark area is covered all connected part of dynamic texture.

Step 3: Save the satisfied result as T^2 for the upcoming process, and $T^2 \subseteq \Phi_i^2$.

V. Patch transfer

A. Template Matching Approach.

The classical template matching method is generally used to find a match region on an image that match or similar to a template or patch. A traditional template matching algorithm can be described as:

Step 1: Store a source image in the system.

Step 2: Store a template in the system.

Step 3: Find out a best matched result by passing the patch one pixel at a time, line by line, from left to right and top to bottom, see Fig. 3(a). At each point, a convolution of patch over the source image is calculated and a matched metric result on each point is stored. The best matched result is analyzed, based on a statistics method.

Step 4: Indicate the matched region on the source image.



Figure 3: Describes the match routing, (a) illustrates the classical match routing, and (b) is our match searching model.

B. Target Template Matching Algorithm.

In this section, we focus attention on target dynamic texture detection. A new template matching method is applied for patch transfer algorithm to increase the accuracy and performance of matched result. Although an idea of a template should not include the background information to avoid the influence of the background changes was studied by Sha *et al.* [4].

Indeed, for dynamic texture patch searching, not only excluding the template background, but also should eliminate the background of dynamic texture in source image to get rid of a match occurred in the background location. The routing path of our template matching algorithm is illustrated in Fig. 3(b).

The target template matching algorithm is described as follow:

- Step 1:** Store the first video Φ_1^3 in the system.
- Step 2:** Store the second video Φ_2^3 in the system.
- Step 3:** Determine the Ω^2 on frame Φ_{2j}^2 in Φ_2^3 by using patch searching algorithm in subsection III-B, where $\Phi_{2j}^2 \in \Phi_2^3$ and $j \in \mathbb{N}_0$.
- Step 4:** Determine the T^2 on the frame Φ_{1i}^2 in Φ_1^3 by using a segmentation algorithm in Section IV, where $\Phi_{1i}^2 \in \Phi_1^3$ and $i \in \mathbb{N}_0$.
- Step 5:** To find the matched region between Φ_{2j}^2 and Φ_{1i}^2 , pass the Ω^2 over Φ_{1i}^2 which the location is verified by T^2 . The metric is stored in the result matrix R . Each $R(x, y)$ contains the match matrix which calculated by a statistics method. The Normalized Sum Squared Differences (NSSD), Cross Correlation (XCOR) and Normalized Cross Correlation (NCC) are used for match metric in our experiment. R matrix calculated by NSSD match metric is shown in (7).

$$R(x, y) = \frac{\sum_{xy} (P(x', y') - I(x + x', y + y'))^2}{\sqrt{\sum_{xy} P(x', y')^2 \cdot \sum_{xy} I(x + x', y + y')^2}} \quad (7)$$

R matrix calculated by XCOR and NCC match metrics is shown in (8) and (9) respectively.

$$R(x, y) = \sum_{x'y'} P(x'y') \cdot I(x + x', y + y')^2 \quad (8)$$

$$R(x, y) = \frac{\sum_{x'y'} (P(x'y') \cdot I(x + x', y + y'))}{\sqrt{\sum_{x'y'} P(x'y')^2 \cdot \sum_{xy} I(x + x', y + y')^2}} \quad (9)$$

where R represents a match metric value at (x, y) which is calculated by \sum operation on Ω^2 and Φ_{1i}^2 , when Ω^2 is passed on Φ_{1i}^2 , every considered pixels on Ω^2 and Φ_{1i}^2 at the location T^2 are used in the calculation. Let P denotes as a pixel on Ω^2 at (x', y') and I represents a pixel on Φ_{1i}^2 .

- Step 6:** Indicate the matched region on Φ_{1i}^2 , where $i, j \in \mathbb{N}_0$ and i, j are independent, and represent a frame number in the video clips.

C. Patch Transfer Algorithm.

Our new algorithm to accomplish patch transfer is discussed in this section. This algorithm combines corresponding algorithms explained in the previous sections. There are two primary components for patch transfer between two video clips as follows.

A source video, where Φ_1^3 represents a source video and Φ_1^2 represents frame in the source video, $\Phi_1^2 \in \Phi_1^3$. A target video, represented by Φ_2^3 , and Φ_2^2 represents frame in the target video, where $\Phi_2^2 \in \Phi_2^3$. The dynamic textures in this experiment are assumed to be continuously appeared in every scene in the video clips. The algorithm for patch transfer between two video clips is concluded as below:

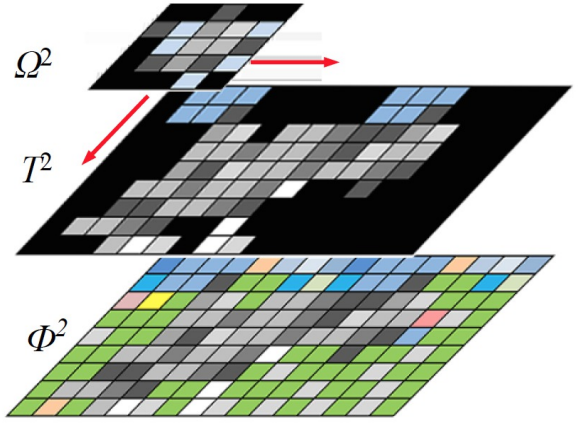


Figure 4: Demonstrates the target template matching components.

- Step 1:** Store first video Φ_1^3 in the system, where Φ_1^3 contains f_1 frames.
- Step 2:** Store the second video Φ_2^3 in the system, where the Φ_2^3 contains f_2 frames, f_1 and f_2 are independent.
- Step 3:** Determine the Ω^2 on frame Φ_{1i}^2 in Φ_1^3 by using patch searching algorithm in subsection III-B, where $\Phi_{1i}^2 \in \Phi_1^3$ and $i \in \mathbb{N}_0$. In this step the system allows user to define more than one patch to cover all area of TOI.
- Step 4:** Determine the number of output frame for new videos. Let f represents the total number of frames for the output video, and k represents the frame number in the new video, where $k = 0, 1, \dots, f$ and $k \in \mathbb{N}_0$.
- Step 5:** Determine the TOI or T^2 on frame Φ_{1i}^2 in the Φ_1^3 by using a segmentation algorithm in section IV.
- Step 6:** Pass the Ω^2 over Φ_{1i}^2 verified by T^2 , the metric is stored in the result matrix R . Each $R(x, y)$ contains the match matrix which calculated by a statistics method in (7), (8) or (9).
- Step 7:** Indicate the matched region on Φ_{1i}^2 verified by T^2 . Let M_{1i}^2 stores the best matched region matrix on Φ_{1i}^2 .
- Step 8:** Pass M_{1i}^2 over the Φ_{2j}^2 , for each considered location, the metric is stored in the result matrix R . Each $R(x, y)$ contains the match matrix which calculated by a statistics method in Step 6, where $\Phi_{2j}^2 \in \Phi_2^3$, $j \in \mathbb{N}_0$, i, j and k are independent.
- Step 9:** Indicate the matched region on Φ_{2j}^2 and suppose M_{2j}^2 stores the best matched region matrix on Φ_{2j}^2 .
- Step 10:** Transform the data between M_{1i}^2 and M_{1j}^2 .
- Step 11:** Here, if $i < f_1$ then increase i by 1 else set $i = 0$ and if $j < f_2$ then increase j by 1 else set $j = 0$, where f_1, f_2 and $f \in \mathbb{N}$, and $f_1, f_2, f > 0$. Stop if $k = f$, else increase k by 1, and go to Step

5 if the T^2 is purposed to be dynamic shape, or go to Step 6 if the T^2 is purposed to be fixed shape.

VI. Experimental Result

We developed a tool for this study to manipulate the dynamic texture transformation. The real-time output can be monitored by the tool, as shown in Fig. 5.

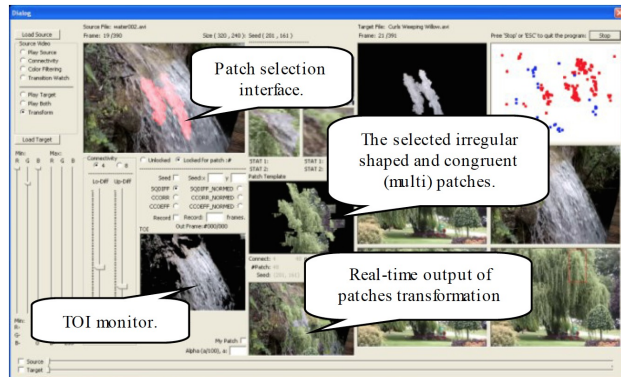


Figure 5: Our tool used to monitor real-time results for patches transformation.

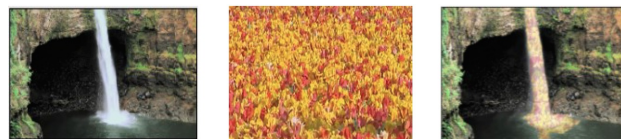
The experiment worked on the video resolution 320×240 . More than fifty video clips were tested to study the dynamic texture characteristics and to change their new look. The successful results are shown in Fig. 6.



(a) New waterfall texture derived from fire texture.



(b) New water flow texture derived from curly leaves tree texture.



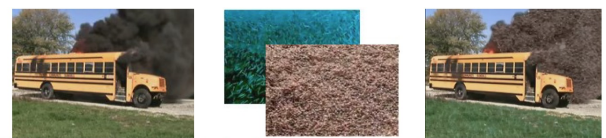
(c) New waterfall texture derived from tulip field.



(d) New smoke texture derived from ants nest texture.



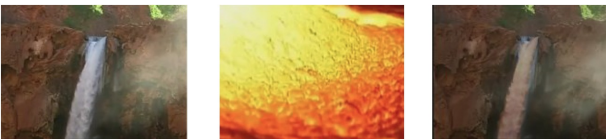
(e) New grass field texture derived from baby fish cloud.



(f) New smoke texture derived from ants nest, and new grass field texture derived from baby fish cloud.



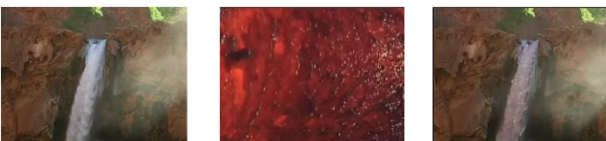
(g) New rocket's smoke texture derived from tulip field.



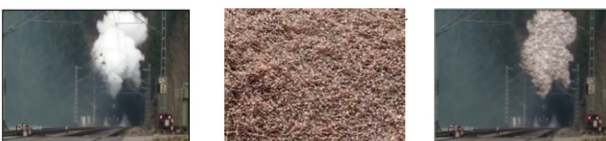
(h) New waterfall texture derived from lava texture.



(i) New lava texture derived from baby fish cloud.



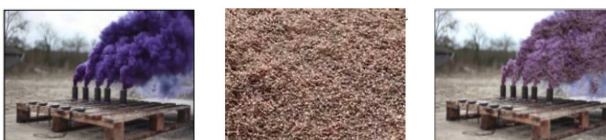
(j) New waterfall texture derived from red lava texture.



(k) New train smoke texture derived from ants nest texture.



(l) New hair color texture derived from red lava texture.



(m) New color smoke texture derived from ants nest texture.

Figure 6: Demonstration of dynamic texture transformation results; the first two columns show the original textures; the last column shows the new look.

VII. Conclusions and future work

In this paper, we presented an idea for transferring a patch of dynamic texture aiming to create a new look of dynamic texture. Several new algorithms were implemented in this study. We propose a simple algorithm for patch searching. Two real-time segmentation techniques were implemented to serve patch transfer on video clips. The target template matching

algorithm was implemented to effectively enhance the patch transfer. In the future works, we plan to investigate more elaborate techniques for varying in dynamic texture transformation. Both spatial and temporal information in dynamic texture will be integrated to our study to enhance the transformation. Blending techniques will be investigated to increase higher quality results.

Bibliography

- [1] *Tsai J.C., Shih T.K., Wattanachote K., Li K.C.* Video editing using motion inpainting // Information Networking and Applications (AINA), 2012. – pp.649–654.
- [2] *Lin C.W. and Cheng N.C.* Video background inpainting using dynamic texture Synthesis // IEEE International Symposium Circuits and Systems (ISCAS), 2010. – pp.1559–1562.
- [3] *Fazekas S., Amiaz T., Chetverikov D. and Kiryati N.* Dynamic texture detection based on motion analysis // International Journal of Computer Vision, 2009. – Vol.82 No.1, pp.48–63.
- [4] *Sha S., Jianer C. and Sanding L.* A fast matching algorithm based on K-degree template // 4th International Conference on Computer Science & Education (ICCSE), 2009. – pp.1967–1971.
- [5] *Doretto G. and Soatto S.* Editable dynamic textures // Computer Vision and Pattern Recognition (CVPR) IEEE Computer Society Conference, 2003. –Vol.2, pp.II — 137–42.
- [6] *Doretto G., Cremers D., Favaro P. and Soatto S.* Dynamic texture segmentation // 9th IEEE International Conference on Computer Vision, 2003. – Vol.2, pp.1236–1242.
- [7] *Sural S., Qian G. and Pramanik S.* Segmentation and histogram generation using the HSV color space for image retrieval // International Conference on Image Processing, 2002.Vol.2, pp.II – 589 – II–592.
- [8] *Phillips III W., Shah M. and Lobo N. de V.* Flame recognition in video // 5th IEEE Workshop on Applications of Computer Vision, 2000. – pp.224–229.
- [9] *Wang L.M., Yan X.L. and Bu L.P.* An edge growth segmentation arithmetic based on HSI color space modeling // identification & Control (ICMIC), 2012. – pp.126–130.
- [10] *O'Malley R., Jones E. and Glavin* Rear-lamp vehicle detection and tracking in low-exposure color video for night conditions // Intelligent Transportation Systems,2010. Vol.11(2), pp.453–62.
- [11] *Mazzeo P.L., Giove L., Moramarco G.M., Spagnolo P. and Leo M.* HSV and RGB color histograms comparing for objects tracking among non overlapping FOVs, using CBT F // Advanced Video and Signal-Based Surveillance (AVSS), 2011. – pp.498–503.
- [12] *Ouyang W.L., Zhang R.Q. and Cham W.K.* Fast pattern matching using orthogonal Haar transform // Computer Vision and Pattern Recognition (CVPR)2010. – pp.3050–3057.
- [13] *Delalandre M., Iwata M. and Kise K.* Fast and optimal binary template matching application to manga copyright protection document analysis systems (DAS) // IAPR, 2014. – pp.298–303.
- [14] *Shapiro L.G. and Stockman G.C.* Computer Vision // Prentice Hall, 2002. – pp.69–73.

Технология формирования локальных архивов 2D-чертежей*

Д.Ю.Васин¹, С.И.Ротков²

¹НИИ Прикладной математики и кибернетики

Нижегородского Государственного Университета им.Н.И.Лобачевского;

²Нижегородский Государственный Архитектурно-Строительный Университет

Рассмотрена автоматизированная технология формирования локальных архивов машиностроительных 2D-чертежей.

Ключевые слова: *электронный архив, инженерная документация.*

Technology of formation of local archives 2D-drawings*

D. Yu. Vasin¹, S. I. Rotkov²

¹Research Institute of Applied Mathematics and Cybernetics,

Lobachevsky State University of Nizhni Novgorod

²Nizhny Novgorod State University of Architecture and Civil Engineering

We consider the automated technology of local archives engineering 2D-drawings.

Keywords: *electronic archive, engineering documentation.*

Не вызывает сомнений, что «за годы работы стоимость архива организации может превысить стоимость всех остальных ее активов» [1]. Бесспорно, в историческом плане от деятельности любой организации, занимающейся разработкой, производством и эксплуатацией подобных изделий и объектов, в первую очередь, остаются архивы проектной и конструкторской документации. Кроме того, широкое использование систем автоматизированного проектирования (САПР) существенно ускорило и упростило процесс создания/редактирования чертежей, но остается актуальной проблема их хранения и эффективного поиска нужной информации. Поэтому создание современных электронных архивов инженерной документации и обеспечения доступа к ним крайне актуально.

Электронный архив – это система структурированного хранения электронных документов, обеспечивающая надежность хранения, конфиденциальность и разграничение прав доступа, отслеживание истории использования документа, быстрый и удобный поиск. Электронный архив относится к классу систем управления корпоративным контентом (Enterprise Content Management). Он представляет собой информационную систему, специальное программное обеспечение, призванное автоматизировать процедуры управления архивным фондом документов организации в соответствии с требованиями государственного нормативного регулирования и спецификой внутренних процессов компании. Электронные архивы предприятий являются основой системы производственного документооборота и содержат, как правило, сканированные и оцифрованные образы бумажных докумен-

тов (конструкторско-технологических, финансово-отчетных, нормативных, патентных и т.п.) и электронные документы. Все чаще архивируется и корпоративная электронная корреспонденция.

Основная причина разработки и создания архива – наличие большого количества бумажной документации, постепенно приходящей в негодность, низкой эффективности ручной работы с документацией с одной стороны и наличием уже внедренных на предприятиях САПР, существовании необходимых программно-аппаратных средств для создания электронного архива, с другой стороны. Создание законченной системы документооборота конструкторской документации ведет прежде всего к повышению эффективности работы предприятия, и, как следствие к росту прибылей.

Основная задача архива – обеспечить сохранность данных. Создается картотека дел, которая хранит информацию о том, где находится документ, отслеживает, кто его запросил, на какой срок, и когда должен вернуть. В архиве должна быть предусмотрена возможность делать выборки файлов по указанным параметрам, например, по сроку хранения, чтобы вовремя избавляться от документов, не требующих хранения.

Основные возможности электронного архива проектной документации: создание электронной картотеки проектов, регистрация и учет входящих документов; ввод документов комплектами по проектам; контроль изменения документов, поддержка нескольких версий; создание копий документа и контроль их выдачи; создание пакетов документов для предоставления отчета по проектам; возможность создания электронных чертежей, редактирования; ведение статистики использования каждого документа или пакета документов.

Работа выполнена и опубликована при финансовой поддержке РФФИ, гранты 15-07-05110, 15-07-20347

Электронный архив конструкторской документации может включать:

- Графические документы: чертежи: детали, сборочные, чертежи общего вида, габаритные, упаковочные и т.д.; схемы.
- Текстовые документы: перечни элементов; пояснительные записки; таблицы; расчеты; инструкции; технические условия; эксплуатационные документы; ремонтные документы; спецификации и т.д.

При этом, наполнение соответствующего архива может осуществляться двумя путями:

- Формированием новых электронных чертежей «с нуля» при помощи САПР-систем. Современные системы САПР широко представлены средствами, обеспечивающими моделирование объектов различной сложности в различных отраслях деятельности. Эти программные комплексы упрощают работу проектировщиков и систематизируют процесс создания цифровой модели изделия.
- Автоматизированного перевода существующего бумажного чертежа в электронный вид с последующим занесением его в электронный архив.

В процессе автоматизированного проектирования оперируют геометрическими объектами (ГО), которые являются как промежуточными, так и окончательными результатами проектирования. ГО характеризуются параметрами, определяющими их форму, а также многими другими сведениями: материалом, чистотой поверхности, термообработкой, допускаемыми отклонениями на размеры и т. д. Большинство этих сведений обычно бывает задано в алфавитно-цифровой форме и поэтому не требует сложной переработки для представления в ЭВМ. Что касается формы ГО, то ее представление значительно сложнее. В дальнейшем нас будет интересовать только эта часть информации о ГО. Поэтому под моделью ГО мы понимаем совокупность сведений однозначно определяющих его форму, то есть принципиально должна существовать возможность установить на основании сведений о ГО для каждой точки пространства, принадлежит она этому объекту или нет. Форма ГО однозначно определяется его чертежом. В модели ГО все сведения о нем должны быть представлены только в алфавитно-цифровой форме (в виде уравнений, таблиц данных, текстовых описаний, построенных по определенным правилам).

В целом, рассматриваемая технология может быть внедрена как подсистема PDM (PLM)-технологии, являющаяся одной из ключевых CALS-технологий, позволяющая управлять данными об изделии, решая две проблемы, возникающие при разработке и поддержке жизненного цикла (ЖЦ) наукоемкой промышленной продукции: управление данными об

изделии и управление информационными процессами ЖЦ изделия, создающими и использующими эти данные. Данные об изделии представляют собой всю информацию, созданную в течение ЖЦ. Они включают в себя состав и структуру изделия, геометрические параметры, чертежи, планы проектирования и производства, спецификации, нормативные документы, программы для станков с ЧПУ, результаты анализа, эксплуатационные данные и пр. Автоматизация этих процессов ставит вопросы эффективного поиска и нахождения актуальных данных, и решения этих вопросов не всегда представляется тривиальным. Создаются и изменяются такие данные в результате выполнения определенных информационных процессов ЖЦ изделия, например процедуры внесения изменений. Информационные процессы могут быть достаточно сложными, охватывающими десятки сотрудников предприятия и при этом взаимосвязанными между собой. Таким образом, в проектах по разработке изделий необходимо не только планировать все входящие в них процессы, но и управлять их выполнением, распределяя задачи между исполнителями, определяя данные, которые им при этом необходимы, и обеспечивая их совместный доступ к этим данным.

При решении задачи CALS-технологий роль PDM-технологии состоит в том, чтобы сделать информационные процессы максимально прозрачными и управляемыми. Эта задача решается путем повышения доступности данных для всех участников ЖЦ изделия, что требует их интеграции в логически единую информационную модель. Системы PDM играют роль связующего звена между этапом инженерно-конструкторской подготовки нового изделия и системами, решающими задачи автоматизации управления финансами, складским хозяйством, снабжением и сбытом, а также техническим обслуживанием.

Очевидно, что PDM-система может выступать в двух основных ролях: 1) как рабочая среда сотрудника предприятия; 2) как средство интеграции данных на протяжении всего ЖЦ изделия [6].

Рассмотрим технологию автоматизированного перевода существующего бумажного чертежа в электронный вид с последующим занесением его в электронный архив.

Первичный анализ чертежа. На данном этапе выполняется визуальная оценка качества исходного чертежа и принимается решение о возможности его автоматической или интерактивной векторизации.

Сканирование. Перевод бумажного чертежа в электронный вид осуществляется посредством планшетных сканеров с разрешением 100–300 dpi. Цветовая палитра выбирается в зависимости от качества и способа получения исходного чертежа.

Для черно-белых чертежей хорошего качества используем бинарную палитру, для «древних» чертежей (синьки) используем сканирование в полноцветном режиме, с последующей цветовой фильтрацией и бинаризацией. Нам представляется предпочтительным использование растрового формата tif в качестве базового для хранения исходного раstra. Данный формат имеет теговую структуру и допускает создание собственных информационных тегов, что исключает необходимость создания дополнительных информационно-описательных файлов в процессе автоматизированной обработки. Кроме того, данный формат имеет хорошую программную поддержку, существует достаточное количество свободно-распространяемых библиотек с открытым кодом для поддержки данного формата. Открытость кода позволяет вносить в него необходимые изменения для поддержки новых тегов.

Геометрическая коррекция раstra. На данном этапе устраняются геометрические искажения (бумагу вставили в сканер с небольшим угловым отклонением), неправильная ориентация («бокком» или «вверх ногами»), нестандартный формат (отсканированный с перекосом чертеж будет иметь нестандартные размеры) и т.п. Допускается автоматическая коррекция по 4 узловым точкам или по сетке узлов.

Предварительная обработка, фильтрация и бинаризация раstra. Невысокое качество исходных бумажных чертежей обуславливает наличие различных искажений входного растрового изображения. Наиболее часто встречающиеся виды помех: низкая контрастность исходного раstra, следы посторонней «грязи» на исходном изображении, линиигиба, «снег» на изображениях, ложные пустоты внутри растровых объектов, ошибочные слипания/разрывы растровых объектов, неравномерность толщины растровой линии. Указанные виды помех требуют выполнения операций контрастирования и фильтрации над исходным растром.

Фильтрация. Предполагалось, что изображение искажено белым гауссовым шумом и выполнялась адаптивная фильтрация яркости посредством НЧ-адаптивного фильтра. Очевидно, что данный фильтр сглаживает не только шум, но также яркостные и цветные границы. При выполнении этого вида фильтрации сначала анализируем прямоугольную, размером $K \times N$, окрестность каждого пикселя, для которого оценивается яркость. В результате этого анализа находим среднее значение яркости \bar{L}_c внутри окна

$$\bar{L}_c = \frac{1}{KN} \sum_{-K/2}^{K/2} \sum_{-N/2}^{N/2} L_c(k, n),$$

а также средний квадрат отклонения от среднего значения яркости σ^2 в этом окне

$$\sigma^2 = \frac{1}{KN} \sum_{-K/2}^{K/2} \sum_{-N/2}^{N/2} [L_c(k, n) - \bar{L}_c]^2,$$

после чего производится непосредственно фильтрация. Учтем, что σ включает в себя как компонент, обусловленный шумом $\sigma_{ш}$, так и компонент, обусловленный изменением яркости в незашумленном изображении σ_L , в свою очередь обусловленную его текстурой и контурами. Поскольку эти компоненты взаимно не коррелированы, то $\sigma^2 = \sigma_{ш}^2 + \sigma_L^2$. Адаптивная фильтрация выполняется таким образом, что в тех местах изображения, для которых дисперсия яркости σ^2 велика, фильтр выполняет слабое сглаживание, поскольку значительная доля отклонения яркости от среднего значения в этих местах обусловлена наличием световых границ или текстурами, которые необходимо сохранить. В тех местах изображения, для которых σ^2 мала, осуществлялось сглаживание в большей степени, так как отклонение яркости от среднего в этих местах изображения обусловлены шумом. При этом, значения яркости пикселей в профильтрованном изображении находятся по формуле:

$$\bar{L}_c(k, n) = \bar{L}_c + \frac{\sigma^2 - \sigma_{ш}^2}{\sigma^2} [L_c(k, n) - \bar{L}_c]$$

Из формулы следует, что при отсутствии шумов $\sigma_{ш}^2 = 0$ и результат фильтрации совпадает с исходным значением яркости, при большой дисперсии шума оценка яркости пикселя $\bar{L}_c(k, n)$ приближается к среднему значению яркости \bar{L}_c внутри окна. В итоге, результат адаптивной фильтрации оказывается лучше, чем при неадаптивной фильтрации. Адаптивный фильтр является более избирательным, нежели сопоставимый ему линейный фильтр, но требует больших вычислительных затрат.

Повышение резкости профильтрованного изображения. Процесс осуществляется посредством усиления контраста между светлыми и темными пикселями. Изменение контраста на границах происходит между соседними пикселями или группами смежных пикселей. При усилении резкости границы преувеличенно подчеркиваются за счет повышения контраста и, таким образом, создаются очень четкие края с резкими переходами между светлым и темным, которые указывают нам, где объект начинается и заканчивается.

Линейное контрастирование. При линейном контрастировании используется линейное поэлементное преобразование вида: $y = ax + b$, параметры которого a и b определяются желаемыми значениями минимальной y_{\min} и максимальной y_{\max}

выходной яркостями при экстремальных значениях исходной яркости x_{\min} и x_{\max} . Решив систему уравнений:

$$\begin{cases} y_{\min} = a \times x_{\min} + b \\ y_{\max} = a \times x_{\max} + b \end{cases}$$

относительно параметров преобразования a и b , получаем

$$y = \frac{x - x_{\min}}{x_{\max} - x_{\min}}(y_{\max} - y_{\min}) + y_{\min}, \quad (1)$$

y – выходная яркость контрастированного пикселя. Тогда алгоритм линейного контрастирования состоит в линейном просмотре всех пикселей исходного изображения и попиксельном изменении их яркости в соответствии с преобразованием (1) для получения выходного контрастированного изображения.

Бинаризация – осуществлялась для 8, 16, 24, 32-битных исходных растров. Использовался стандартный пороговый алгоритм. Порог бинаризации оценивался гистограммным методом.

Интерактивное редактирование – осуществляется любым доступным растровым графическим редактором. Основная задача редактирования – разделение неправомерно слипшихся частей чертежа, возникающие в результате искажений в процессе сканирования. В настоящее время ведутся работы по созданию интеллектуального растрового редактора для автоматизированного редактирования растров чертежей.

Векторизация – использовались алгоритмы построения линейных моделей, разработанные в НИИ ПМК ННГУ. адаптированные к рассматриваемой предметной области [2, 3]. Результаты векторизации записываются в dxr-файл для обеспечения возможности дальнейшего интерактивного редактирования существующими САД-системами. Автоматический поиск ошибок векторизации. Осуществляется автоматический поиск ошибок векторизации подсистемой поиска геометрических ошибок [4, 5]. Коррекция осуществляется существующими САД-системами в автоматически выделенных областях.

Загрузка векторного чертежа в базу данных. В соответствии с ГОСТ 2.104-2006 ЕСКД были использованы обязательные атрибуты чертежа, разделенные на две таблицы: 1. таблица «Основные Сведения» содержит атрибуты, описывающие новую запись в базе, 2. таблица «Изменения» содержит атрибуты, предназначенные для записи изменения существующей записи. Все поля обеих таблиц являются обязательными для заполнения.

Описание атрибутов таблицы «Основные Сведения». Ключевым является автоинкрементное поле

идентификатор записи. Следующие поля являются текстовыми: *Наименование изделия, Наименование документа, Обозначение документа, Код документа, Характер работы, выполняемой лицом, подписывающим документ, Фамилия лица, подписывающего документ, Дата подписания документа, Фамилия лица, принявшего подлинник, Фамилия лица, принявшего дубликат, Номер решения утверждения документации Фамилия должностного лица.* Поле *Год утверждения документации* является целочисленным. Следующие поля имеют тип Дата/Дата Время: *Дата приемки, Дата приемки дубликата, Дата визирования.*

Описание атрибутов таблицы «Изменения». Ключевым является автоинкрементное поле *идентификатор записи.* Поле *Порядковый номер изменения* – целочисленное. Следующие поля являются текстовыми: *Указания об изменении листа, Номер документа на основании которого производятся изменения, Фамилия лица, внесшего изменения, Ссылка на чертеж.* Поле *Дата внесения изменения* имеет тип Дата/Дата время.

Каждый чертеж имеет свой уникальный индекс ID и каждому чертежу будет соответствовать набор атрибутов из таблицы «Изменения». Поэтому таблицы связаны отношением «один ко многим».

Выбор СУБД. В качестве используемой СУБД предлагается платформа MySQL, поскольку поддерживает структурированный язык запросов SQL и может применяться в качестве SQL-сервера. Технология информационного обмена между сервером и клиентом стандартная: на языке SQL клиент посылает серверу запрос, тот его обрабатывает и отдает клиенту только те данные, которые были получены в результате этого запроса. Тем самым клиенту не требуется скачивать данные и производить вычисления. MySQL – имеет открытый код. Преимущества MySQL: *быстродействие.* Благодаря внутреннему механизму многопоточности быстродействие MySQL весьма высоко; *безопасность.* Довольно высокий уровень безопасности обеспечивается благодаря базе данных mysql, создающейся при установке пакета и содержащей пять таблиц. При помощи этих таблиц можно описать, какой пользователь из какого домена с какой таблицей может работать и какие команды он может применять. Пароли, хранящиеся в базе данных, можно зашифровать при помощи встроенной в MySQL функции password(); *Открытость кода.* Благодаря этому можно добавлять в пакет нужные функции, расширяя его функциональность так, как требуется; *Надежность.* СУБД достаточно стабильна и защищена от сбоев; *Сообщество.* Как следствие открытости кода, стабильной и надежной ее работы образовалось сообщество людей, которые всячески участвуют в ее развитии и совершенствовании; *Переносимость.* В настоящее время суще-

ствуют версии программы для большинства распространенных компьютерных платформ.

В настоящее время реализована загрузка в базу данных растрового изображения чертежа, либо dxf-файла чертежа и запросы на выгрузку нужного чертежа и возможных его изменений.

Литература

- [1] *Нужненко С., Орешкин А., Богданова И.* Почему нынче документооборот // CADmaster, 03/2007, 42с.
- [2] *Васин Ю.Г., Башкиров О.А., Рудометова С.Б.* Математические модели структурированного описания графических изображений // Автоматизация обработки сложной графической информации, Горький: 1984. – С.92–117.
- [3] *Васин Д.Ю.* Исследование моделей описания, разработка алгоритмического, программного и технологического обеспечения обработки растровых изображений графических документов // Дисс. канд. техн. Наук, 05.01.01. – Нижний Новгород: ННГАСУ, 2006.
- [4] *Васин Д.Ю., Ротков С.И.* Автоматическое обнаружение геометрических ошибок на машиностроительных 2D-чертежах // 24-ая Международная конференция по компьютерной графике и зрению Графикон'2014 (30 сентября — 3 октября, 2014 г.), Ростов-на-Дону: Академия архитектуры и искусств, Институт механики, математики и компьютерных наук, Южный Федеральный Университет. С.20–22
- [5] *Васин Д.Ю., Ротков С.И.* втоматическое обнаружение геометрических ошибок на машиностроительных 2D-чертежах при формировании электронных архивов // Приволжский научный журнал, Н. Новгород: ННГАСУ, 2015.-№3(35). – С.68–81
- [6] <http://www.tadviser.ru/index.php/> – Продукт: Лоцман: PLM.

Распознавание символов на основе инвариантных моментов графических изображений*

Д.Ю.Васин¹, А.В.Аратский²

dm04@list.ru

¹НИИ Прикладной математики и кибернетики

Нижегородский Государственный Университет им.Н.И.Лобачевского;

²Факультет ВМК Нижегородского Государственного Университета им. Н.И.Лобачевского

В докладе рассмотрен алгоритм формирования эффективных признаков и решающих правил распознавания символов на основе инвариантных относительно масштабирования моментов на базе низкоуровневой структурированной модели описания бинарного растрового изображения.

Ключевые слова: распознавание с самообучением, оценка сходства, алгоритм распознавания.

The character recognition based on invariant moments graphics*

D. Yu. Vasin¹, A. V. Aratsky²

¹Research Institute of Applied Mathematics and Cybernetics,

Lobachevsky State University of Nizhni Novgorod

²Student of the Faculty of Lobachevsky State University of Nizhni Novgorod

The report considered the algorithm of formation of effective signs and decision rules based on character recognition are invariant with respect to scale points on the basis of a low-level structured model describing binary bitmap.

Keywords: recognition with self-learning, assessment of the similarity, algorithm recognizers.

Одной из основных проблем, возникающих при решении задачи распознавания образов, и существенно влияющих как на эффективность распознавания, с одной стороны, так и на его качество, с другой, является выбор системы эффективных признаков классификации. В работе рассматривается формирование решающих правил в задаче распознавания символов, находящихся в стандартной горизонтальной ориентации, но допускающих растяжение/сжатие по горизонтали/вертикали отдельных символов, на базе низкоуровневой структурированной модели описания бинарного растрового изображения (БРИ) [1]. Пример изображения текста для распознавания приведен на рисунке 1.

Пример текста для
алгоритма
распознавания

Рис. 1

Моментные характеристики изображения.

Моментные характеристики нашли широкое при-

Работа выполнена и опубликована при финансовой поддержке РФФИ, гранты 13-07-00521, 13-07-12211, 15-07-20347.

менение во многих задачах обработки цифровых изображений [1]. Рассчитываемые на их основе моментные инварианты стали важнейшим инструментом для распознавания образов независимо от их особого положения, ориентации, угла обзора и других изменений. Основным достоинством моментных инвариантов являются нечувствительность их к различным геометрическим преобразованиям (сдвиг, поворот, масштабирование) [1]. В данной работе рассматриваются нормированные моменты – инвариантные относительно масштабирования. Момент — это характеристика контура изображения, объединённая (суммированная) со всеми пикселями контура. Момент $m_{p,q}$ для непрерывного изображения $f(x, y)$ определяется как:

$$m_{p,q} = \iint x^p y^q f(x, y) dx dy \quad (1)$$

в случае дискретного растрового изображения

$$m_{p,q} = \sum \sum x^p y^q f(x, y), \quad (2)$$

где p, q — мощность, на которой соответствующий компонент взят в сумме с другими отображенными. Если функция $f(x, y)$ рассматривается как бинарное изображение, тогда момент m_{00} вычисляет площадь чёрных пикселей этого изображения. Для вычисления характеристик инвариантных относительно сдвига, необходимо центрировать относительно средних значений $\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}}$ — центр тяжести фигуры.

Центральный момент определяется как:

$$\mu_{p,q} = \sum \sum (x - \bar{x})^p (y - \bar{y})^q f(x, y).$$

Доказано, что все центральные моменты выражаются через обычные, по формуле:

$$\mu_{p,q} = \sum_{i=0}^p \sum_{j=0}^q c_p^i c_s^j (-\bar{x})^{(p-i)} (-\bar{y})^{(q-i)} m_{ij}, \quad (3)$$

где $C_n^k = \frac{n!}{k!(n-k)!}$.

Для реализации инвариантности относительно масштабирования центральные моменты нормируют. *Нормированные моменты* вычисляются следующим образом:

$$\eta_{p,q} = \frac{\mu_{pq}}{\mu_{00}^{1+\frac{p+q}{2}}}, \quad \eta_{p,q} = \frac{m_{pq}}{m_{00}^{1+\frac{p+q}{2}}}. \quad (4)$$

В первом случае нормированный момент независим относительно масштабирования и сдвига, во втором – только относительно масштабирования.

Для поворота также есть соответствующие формулы разных порядков, вычисляющийся по нормированным моментам.

Инварианты бинарного изображения, представленного в штриховом виде. Основной упор в работе был направлен на нахождение инвариантных характеристик изображений символов в штриховом форме относительно растяжения/сжатия. В работе рассматривались следующие инварианты: по количеству горизонтальных и вертикальных штрихов. Рассмотрим эталонное изображение буквы Б (слева) и его же растянутое в $n = 2$ раз по горизонтали см. рисунок 2.

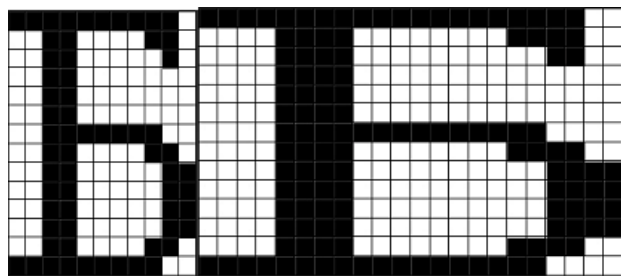


Рис. 2

Пусть w и h – соответственно ширина и высота оригинального изображения, а W и H – соответственно ширина и высота растянутого изображения. Очевидно, что между ними есть связь вида: $W = n \times w$, $H = h$. Так же обозначим dw и dh (dW и dH) – количество горизонтальных и вертикальных штрихов соответствующего изображения.

Можно заметить, что при растяжении по горизонтали сохранилась не только высота, но и количество штрихов по горизонтали. В тоже время при растяжении по ширине в n раз, у изображения образуется $(n - 1) \times w$ пустых столбцов пикселей которые надо заполнить, исходя из уже заполненных.

Очевидно, что вертикальные штрихи будут дублированы n раз. При не целых значениях n имеет смысл говорить о примерных соотношениях. Эти рассуждения аналогичны при масштабировании по вертикали. Соответственно инвариантные характеристики масштабированного штрихового изображения: $w/dh = a = const$, $h/d = b = const$. Однако на практике такая характеристика может довольно сильно колебаться. Не говоря о том, что одной такой характеристики недостаточно для полного отделения классов друг от друга.

Инвариант штриха относительно масштабирования. У штриха можно найти характеристику, не меняющуюся при масштабировании изображения. Рассмотрим горизонтальный штрих $d = \{i_d, s_d, f_d\}$ изображения размером w на h . Ясно, что при растяжении изображения по горизонтали, штрих также растягивается и отношение его длины к ширине сохраняется. Рассматривая растяжение по вертикали можно заметить, что отношение длины отрезка $[0, i_d]$ к высоте также сохраняется. Приведём эти нормированные значения к одному в виде угла Γ , см. рисунок 3.

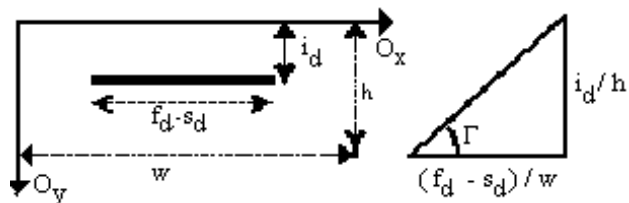


Рис. 3

Соответственно для штриха инвариантный коэффициент угла γ_d вычисляется по формуле: $\gamma_d = \arctan\{(i_d/h)(w/(f_d - s_d))\}$. Для получения инвариантной характеристики целого изображения D_n можно взять среднее арифметическое для γ по всем n его штрихам: $\bar{\gamma} = \frac{1}{n} \sum_{d \in D_n} \gamma_d$. Эмпирические наблюдения показали, что даже в большом растяжении по вертикали (в 5 раз и более) погрешность изменения не превышает сотых долей радиана. Однако эта характеристика как дискриминантный признак использоваться не может, т.к. на разных буквах она принимает очень близкие значения.

Приведённые примеры обладают слишком большой вычислительной сложностью. Они требуют полной обработки изображений, при этом не давая существенного выигрыша в выходном качестве распознавания.

Однако, используя моментные характеристики изображений, можно получить эффективные в вычислительном плане признаки, которые образуют достаточное признаковое пространство, для отделения классов друг от друга.

Использование ключевых строк и столбцов. Очевидно, что средний столбец пикселей растянутого относительно центра изображения остается без изменений. То же справедливо для центральной строки изображения при растяжении по вертикали. Это значит, что центральный столбец/строку изображения можно рассматривать как некоторую постоянную величину. Для этого необходимо найти некоторую числовую характеристику, которая однозначно ставится в соответствие соответствующей полоске пикселей или строке штрихов. Для этого воспользуемся формулой *нормированных моментов*. Рассмотрим общий подход для центральной строки, аналогичные рассуждения для центрального столбца см. рисунок 4.

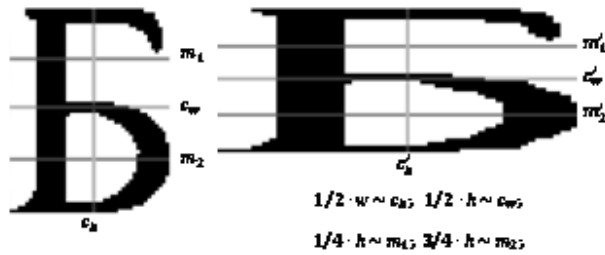


Рис. 4

Пусть высота и ширина изображения буквы соответственно h и w . Рассмотрим строку с номером $h/2$ и вычислим для неё числовую характеристику C_w^1 , которую можно считать хэш-функцией от строки. Аналогично можно вычислить значение C_w^2 , у растянутого изображения. Для столбцов аналогичные характеристики обозначим C_h^1, C_h^2 . Если вычисляемая характеристика инвариантна относительно масштабирования, то имеет смысл их сравнивать. По получившимся характеристикам получим признаковое описание объекта, в виде вектора вещественных чисел (c_w, c_h) . Тогда объект можно рассматривать как точку в признаковом пространстве \mathbf{R}^2 . Однако, не для всех букв достаточно двух значений. Например, у букв Y и X центральные полосы по горизонтали и вертикали будут похожими. В таком случае можно провести дополнительные полосы m_1 и m_2 , выбирая строки, как $1/4$ и $3/4$ высоты. Тем самым расширить признаковое пространство до \mathbf{R}^4 , и увеличить точность распознавания. Главный плюс такого подхода в том, что вычисляется момент не всего изображения, а лишь его небольшой части, а получаемые характеристики, для разных букв разные, что обеспечивает пересекающиеся множества классов. Рассмотрим подробнее, как производится расчёт моментов.

Нормированный момент штриховой строки. При работе со штриховым изображением также можно использовать центральные и нормированные мо-

менты. Заметим, что вычисление всех инвариантов опирается на вычисления простых моментов. Тогда, создав формулу для вычисления моментов штрихового изображения эквивалентную (2), формулы (3), (4) для центральных и нормированных моментов останутся справедливыми.

Разберём несколько случаев для монохромного штрихового изображения. Рассмотрим случай горизонтальных штрихов вида $d = \{i_d, s_d, f_d\}$ где i_d — номер строки по x , s_d, f_d — начало и конец штриха по y .

1. m_{00} — есть площадь чёрных пикселей. Для штрихового формата, очевидно, нужно подсчитать площадь всех штрихов D_n . Высота горизонтального штриха 1 пиксель, длина $f_d - s_d$:

$$m_{00} = \sum_{d \in D_n} (f_d - s_d).$$

2. m_{p0} — в данном случае каждый пиксель в штрихе умножается на x^p , что есть константные значения для горизонтального штриха равное i_d :

$$m_{00} = \sum_{d \in D} (f_d - s_d) i_d^p.$$

3. m_{0q} — в данном случае каждый пиксель в штрихе умножается на y^q , что есть изменяющееся от s_d к f_d значение для горизонтального штриха:

$$m_{0q} = \sum_{d \in D} \sum_{j=s_d}^{f_d} j^q.$$

4. m_{pq} — объединение рассмотренных случаев:

$$m_{pq} = \sum_{d \in D} (i_d^p \sum_{j=s_d}^{f_d} j^q) = \sum_{d \in D} \sum_{j=s_d}^{f_d} i_d^p j^q. \quad (5)$$

Поскольку штрих — одномерный кластер связанных чёрных пикселей бинарного изображения, для которых $f(x, y) = 1$, то данная формула эквивалентна дискретному моменту обычного растрового изображения. Следовательно, инвариантные моменты справедливы и для штрихового формата.

Так как основная идея состоит в использовании лишь опорных строк или столбцов изображения, то высота/ширина строки/столбца будет всегда равна 1 пикселю. И вычисляемый момент будет зависеть только от одного порядка, т.к. другая координата равна константе (при $q = 0$, это будет 1). Пусть задана строка штрихов D , тогда строковый момент вычисляем по формуле:

$$M_p(D) = \sum_{d \in D} \left(\int_0^1 \int_{s_d}^{f_d} x^p y^q dx dy \right) \Rightarrow \frac{1}{p+1} \sum_{d \in D} (f_d^{p+1} - s_d^{p+1}), \text{ при } q = 0. \quad (6)$$

Далее будет подразумевать под моментом строки (или столбца) именно выражение (6). Момент строки назывём горизонтальным, а столбца вертикальным.

Соответственно через моменты выражаем центр тяжести строки (столбца): $\bar{x} = M_p/M_0$. Справедлива и формула для нормированного момента. Однако тестирование показало, при сильном масштабировании значение нормированного момента также начинает отклоняться. Это связано с тем, что расчёт моментов ведётся в абсолютных величинах (координатах пикселей). Расчёт моментов в относительных координатах (отношение координат начала и конца штриха к общей длине полосы), даёт минимальные колебания.

$$\tilde{M}_p(D) = \frac{1}{p+1} \sum_{d \in D} \left(\left(\frac{f_d}{|D|} \right)^{p+1} - \left(\frac{s_d}{|D|} \right)^{p+1} \right)$$

В (2) $|D|$ — это длина изображения для горизонтального момента, и высота — для вертикального. Нормированный момент:

$$N_p(D) = \left(\frac{M_p}{M_0^{p+1}} \right)^{\frac{1}{p+1}}$$

На рисунке 5. представлены все нормированные вертикальные (слева) и горизонтальные (справа) моменты изображения эталонной буквы Б.

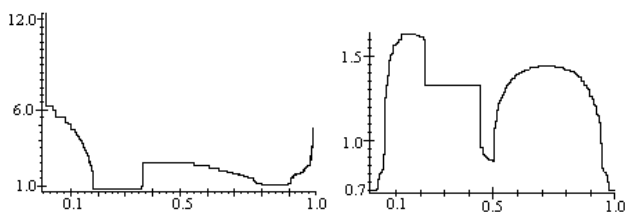


Рис. 5

Пространство признаков. В качестве набора признаков выберем нормированные вертикальные и горизонтальные моменты. Оставшийся вопрос в их количестве и номере строк (столбца) для подсчёта может быть очень вариативным. Будем исследовать некоторые выбранные фиксированные случаи. Определим для случая вертикальных моментов (аналогично для горизонтальных) основные:

Центральный момент изображения c_w — нормированный момент строки, взятой по номеру $h/2$.

Двойной момент изображения $\{a_w^1, a_w^2\}$ — нормированные моменты строк, взятых по номерам $h/3$ и $2/3$.

Тройной момент изображения $\{b_w^1, b_w^2 = c_w, b_w^3\}$ — нормированные моменты строк взятых по номерам $h/4, h/2$ и $3h/4$.

Исходя из сказанного, пара множеств и будет определять признаковое описание объектов и эталонов классов. Всего будет от 2 ($\{c_w, c_h\}$) до 6 ($\{b_w^1, c_w, b_w^3, b_h^1, c_h, b_h^3\}$) моментов. Поскольку признаковое пространство есть не что иное, как множество $R_n = \{x \in \mathbf{R}^n : x_i \geq 0, i = 1, 2, \dots, n\}$ каждый класс представляется точкой в пространстве. Алгоритм классификации объекта, в таком случае сводится к поиску ближайшего эталона по выбранной метрике.

Чем больше размерность признакового описания, там больше вероятность разнесения классов друг от друга. Тестирования с разными векторами моментов показало, что при 6-мерном векторе классы в достаточной степени отдаляются друг от друга. Однако, такое отдаление очень сильно зависит от используемого шрифта, размера и качества исходного документа. В случае с двумерным вектором моментов довольно часто ситуация когда несколько классов оказываются в большой близости друг от друга. На рисунке 6 представлено множество эталонов в двумерном признаковом пространстве.

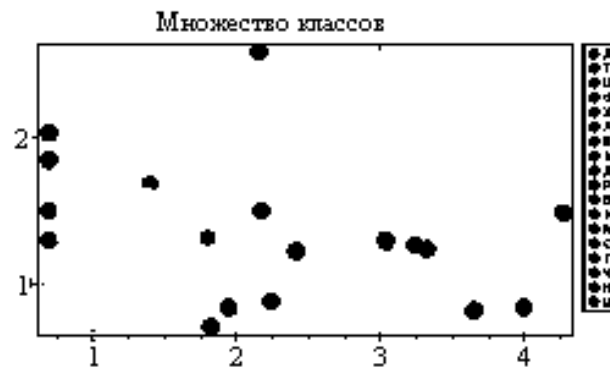


Рис. 6

В показанном распределении классы букв О и Г находятся в близости друг к другу. Такое распределение классов может

Расширения пространства признаков. К сожалению, нормированный строковый момент может давать похожие значения для разных полос. Это связано с тем, что по факту вычисляется площадь интеграла степенной функции на некотором промежутке, ограниченным рассматриваемым штрихом. Нетрудно подобрать разные по длине и положению штрихи, но одинаковые по значению нормированного момента. Что бы избежать такой ситуации можно помимо моментов хранить ещё дополнительную информацию по полосе. В частности можно хранить центр тяжести полосы, получаемые по формуле $X = N_p/N_0$, при этом размерность признакового описания возрастает в два раза.

Опишем поэтапно работу алгоритма распознавания.

Этап подготовки изображения. На данном этапе изображения разбиваемся на связные области чёрных пикселей, каждая из которых является буквой. Под выделением связных компонент понимаем присвоение уникальной метки каждому объекту изображения – букве. Каждую компоненту будем описывать меткой и описанной вокруг буквы прямоугольной областью. Отметим, что данная операция является наиболее времязатратной. Помимо выделения связных компонент, разобьем их по общей структуре текста на строки и слова.

Алгоритм организации структуры теста. На входе имеем общий список из связных компонент (в виде информации о расположении прямоугольной области на изображении и метки).

1. Отсортируем компоненты по возрастанию координаты у их центров прямоугольной области.
2. Выбираем первую компоненту из общего списка и смотрим пересечение горизонтальной линии, проходящей через центр компоненты с другими компонентами. Будем считать, что все попавшие компоненты расположены на одной строке. Для них:
 - (а) Отсортируем центра компонент по возрастанию координаты X .
 - (б) Пройдя по отсортированному списку, найдём расстояние между их крайними границами прямоугольников. Если оно больше некоторого заданного среднего значения, будем считать это пробелом.
 - (в) Запишем соответствующую структуру строки в список меток, хранящий информацию о структуре текста, где метка = 1 означает пробел, а метка = 2 — переход на новую строку.
3. Удалим попавшие в одну строку компоненты из общего списка и повторим шаг 2, пока не рассмотрим все компоненты.

Этап обучения. Обучение производится в интерактивном режиме, при этом определяется, какие моменты (центральный, двойной, тройной) будут вычислены по вертикали и горизонтали выбираются эталоны. Как и было обговорено выше, будем проводить вычисление признаков для выбранного эталона по горизонтали (вертикали) при $p = 2$:

Центральный $w = N_p(D)$ и его дополнение $\bar{X}_c = N_p/N_0$ в строке (столбце) D под номером $h/2(w/2)$.

Двойной, $a_w^i = N_p(D_i)$, $i = 1, 2$ и их дополнение $\bar{X}_a^i = N_p/N_0$ в строках (столбцах) с номерами $h/3(w/3)$ и $2h/3(2w/3)$.

Тройной, $b_w^i = N_p(D_i)$, $i = 1, 2, 3$ и их дополнение $\bar{X}_a^i = N_p/N_0$ в строках (столбцах) с номерами $h/4(w/4)$, $2h/4(2w/4)$ и $3h/4(3w/4)$.

Различные варианты пар нормированных моментов по горизонтали и вертикали будут образовывать признаковое пространство.

Этап распознавания. Основной и наиболее быстрый этап в работе алгоритма. Каждый образец буквы есть точка в пространстве $R_n\{x \in \mathbf{R}^n: x_i \geq 0, i = 1, 2, \dots, n\}$. При этом для каждого образца имеется дополнительный вектор из центров тяжести, представимый точкой в пространстве $C_n\{x \in \mathbf{R}^n: 0 \leq x_i \leq 1, i = 1, 2, \dots, n\}$. Введём δ -окрестность вокруг точки отвечающей эталону.

Возможны два варианта действия:

- при попадании образца в δ -окрестность эталона считать, что образец однозначно классифицирован и использование дополнительного пространства C_n не требуется («притягивающий» вариант);
- при попадании образца в δ -окрестность эталона считать, что образец претендует на класс эталона и требовать дополнительной проверки, используя для этого пространство C_n («отталкивающий» вариант).

Использование дополнительного пространства C_n необходимо контролировать с помощью специального коэффициента нагрузки $\alpha > 0$. Это связано с тем, что компоненты в C_n принадлежат отрезку $[0, 1]$, а в R_n отрезку $[0, \infty]$.

Алгоритм распознавания символа. На входе алгоритма растровая связная компонента и набор эталонов.

1. Для каждого эталона найдём Δ -характеризующую близость к эталону.
 - (а) Для вектора момента найдём близость по метрике $\rho = \sqrt{\sum_{i=1}^n (m_i - m_i^E)^2}$, где $m_i = c, a^1, a^2, b^1, b^2, b^3$ в зависимости от выбранных моментов.
 - (б) Делаем проверку $\rho < \delta$ для «отталкивающего» варианта ($\rho > \delta$ для «притягивающего» варианта):
 - i. Если так, рассчитываем близость векторов пространства по метрике $\tau = \alpha \sqrt{\sum_{i=1}^n (C_i - C_i^E)^2}$.
 - ii. Иначе берём $\tau = 0$.
2. Находим $\Delta = \rho + \tau$ для каждого эталона и выбираем класс эталона с минимальным Δ и определяем символ, который принадлежит данному классу.

Тестирование показало, что притягивающий и отталкивающий подходы благодаря регулирования значений δ и α во многом эквиваленты. Следует заметить, что при большой размерности R_n дополнительная нагрузка и вовсе не нужна, так как признаковое пространство в достаточной степени разделяет классы друг от друга.

Алгоритм распознавания символа запускается на каждой связной компоненте после чего на основе полученных результатов формируется текст той структуры, которую определили на этапе обработки.

На базе разработанных алгоритмов создано оригинальное ПО. Практические тесты проводились на растровом бинарном изображении текста (шрифт Times New Roman, 16 кегль). Процент правильно распознанных знаков от 77 до 97. При этом, очевидно, что чем больше размерность пространства, тем более эффективен алгоритм, а центральные моменты дают лучший результат, чем двойные моменты. Так же тестирование показало, что при небольшом повороте около $\pm 10^\circ$ эффективность распознавания (при размерности \mathbf{R}^6) сохраняется на уровне 85-95%.

Считая, что допустимое количество ошибок не должно превышать 10–15% от общего числа символов текста, признаковое пространство \mathbf{R}^6 и более, с дополнительной нагрузкой в виде центров тяжести штрихов будет оптимальным решением. Одна-

ко получаемая эффективность относится к хорошо отфильтрованным изображениям. При тестировании на старых документах с плохой фильтрацией при попадании полосы на белые пятна нормированные моменты будут подсчитаны неверно, и эффективность распознавания по сравнению с качественным изображением упадет в 2 раза (с 97-100% до 50-55%).

Литература

- [1] Васин Д.Ю., Васин Ю.Г., Громов В.П. Структурное описание растровых данных // VI Всероссийская конференция «Методы и средства обработки сложной графической информации». Тезисы. Н.Новгород. 2001. С.21–23.
- [2] Wong R. Scene matching with invariant moments // AComputer Graphics and image processing, 1978.– Vol.8. – pp.16–24.
- [3] Васин Д.Ю. Image Analysis by Moments // The Department of Electrical and Computer Engineering The University of Manitoba, 1993.

Опыт создания мультимедийных систем реального времени для обучения и презентаций*

Б.С. Долговесов

bsd@iae.nsk.su

Новосибирск, Россия, Институт Автоматики и Электростроения СО РАН

В статье описаны подходы к применению современных компьютерных технологий в образовательных и презентационных приложениях. В частности, рассматриваются конкретные решения, базирующиеся на мультимедийных системах реального времени, таких как системы виртуальной реальности, системы приёма, обработки и трансляции видео и аудио потоковых данных, генерации графики, интерактивного управления.

Ключевые слова: виртуальный презентер, виртуальная студия, видеомикшер, образовательный процесс

Real-Time Multimedia Systems for Education and Presentation*

B.S. Dolgovesov

Novosibirsk, Russia, Institute of Automation and Electrometry SB RAS

The article describes the approaches to the use of modern computer technology in education and presentation applications. In particular, it addresses specific solutions based on real-time multimedia systems, such as virtual reality systems, the system of reception, processing and transmission of video and audio streaming data, generating graphics, interactive control.

Keywords: virtual presenter, virtual studio, video mixer, educational process

1. Введение

Современное развитие информационно-коммуникационных технологий (ИКТ) обусловило возможность создания на базе технологий мультимедиа, телекоммуникаций и виртуальной реальности новых технологических решений представления информации, значительно повышающих эффективность и доступность мультимедиа контента для конкретных приложений (очное и дистанционное обучение, видеопроизводство, выставки, шоу, интерактивные презентации и т.д.). В информационном обществе современное образование все больше использует экранные технологии работы с медиа-данными. Недаром в развитых странах, практически во всех учебных заведениях среднего и высшего звена существуют свои медиацентры и учебные студии. Они предназначены как для производства учебного контента преподавателями и учителями, так и для обучения методике и технике создания медиаконтента самими учащимися. Вся эта деятельность активно финансируется государством и частными фондами, как в виде грантов на создание медиапродукции, так и в виде финансирования создания и деятельности медиацентров и студий.

2. Мультимедийные системы для обучения и презентаций

Имеющийся опыт участия в нескольких проектах школьного и вузовского образования, в ко-

торых использовались наши разработки в качестве базовых инструментов для построения учебно-производственных студий в образовательных учреждениях, показал достоинства и недостатки ранних решений [1-6] для образовательного процесса, с учетом которых созданы новые мультимедийные системы с расширенными функциональными возможностями. Эти системы наиболее адаптированы для использования в образовательных учреждениях как для различного рода демонстраций, так и для интерактивного видеопроизводства на основе современных технологий.

Одна из таких систем – программный микшер «AllMix», обеспечивающий следующие функциональные возможности:

- до 8-и входных источников данных в любых форматах (SD/HD);
- различные компьютерные медиаисточники (изображения, видеофайлы и т.п.);
- цветокоррекцию, хромакей, вписывание/масштабирование любых видеоисточников;
- превью источников/результатов непосредственно в интерфейсе (мультивью);
- расширяемый набор эффектов-переходов микширования;
- аудиомикширование с сохранением пресетов;
- логотипы, часы, картинка в картинке;
- создание многослойных титров (обеспечивается встроенным редактором и оперативным управлением);
- воспроизведение медиафайлов по сценарию;

Работа опубликована при финансовой поддержке РФФИ, грант 15-07-20347

- трёхмерная виртуальная студия в качестве медиа-источника;
- поддержку внешних консолей управления (пульта DSC, Behringer BCF 2000);
- поддержку аппаратных модулей ввода-вывода FDExt (SoftLab), DeckLink (BlackMagic);
- вещание в сеть.

Пользовательский интерфейс микшера (рис.1) предельно прост, нагляден и обеспечивает высокое качество управляющих изображений. Микшер может быть реализован в нескольких конфигурациях для разных бюджетов и задач (от съёмки/вещания лекций-презентаций до небольшой новостийной студии, от студии прогноза погоды до ток-шоу). При этом можно как сочетать в одном лице функции режиссёра, оператора, титровальщика, звукорежиссёра, так и выполнять эти функции отдельными исполнителями. Низкая стоимость, простота установки и эксплуатации, широкий набор функций микшера являются предпосылками для организации на его базе учебно-производственных студий в условиях образовательных учреждений. Студии, создаваемые на базе программного микшера «AllMix» можно использовать как для обучения интерактивному производству медиаматериалов, так и для профессионального их создания.



Рис. 1: Пользовательский интерфейс микшера

Другой класс мультимедийных систем реального времени, к которым относится «Виртуальный Презентер», предназначен в первую очередь для индивидуального использования создателями контента (лекций, презентаций, докладов и др.). Докладчику предоставляется возможность самостоятельно готовить и интерактивно управлять ходом лекции/презентации, при этом изображение докладчика помещается в генерируемую в реальном времени виртуальную среду презентации.

«Виртуальный Презентер» – многофункциональный программно-аппаратный комплекс на базе персонального компьютера (или Notebook-компьютере) с интерактивными средствами для

взаимодействия с виртуальной средой в реальном времени. Комплекс позволяет совмещать видео-изображение реального персонажа (лектора, преподавателя) и мультимедийные данные различных форматов (виртуальные сцены, видеоданные с различных носителей, Power Point – презентации, текст, и др.), предоставляя видеоматериал с разнообразным информационным наполнением. Интерактивное «присутствие» преподавателя (лектора) в предметной виртуальной среде является одним из существенных преимуществ комплекса, обеспечивающего повышение интереса к демонстрируемым материалам, что, наряду с наглядностью изучаемых предметов, повышает эффективность их усвоения. Совмещенное высококачественное видеоизображение лектора и виртуальных объектов, представляющих материалы учебного и познавательного характера, можно демонстрировать одновременно в нескольких учебных классах, лекционных залах с помощью разнообразных технических средств (проекторы, плазменные панели и др.), передавать в реальном времени удалённым потребителям и записывать на различные носители. Для управления комплексом используется графический пользовательский интерфейс (рис.2), позволяющий докладчику, без привлечения сторонних специалистов, легко управлять ходом демонстрации видеоматериалов, переключать режимы микширования, медиаисточники данных и использовать вещание в сеть.



Рис. 2: Графический пользовательский интерфейс для управления комплексом

Основные достоинства программно-аппаратного комплекса «Виртуальный Презентер»:

- эффект интерактивного «присутствия» лектора в предметной виртуальной среде как важный психологический фактор концентрации внимания слушателей на демонстрируемом материале, способствуя тем самым наилучшему его усвоению;

- демонстрация моделей в произвольном ракурсе, изменяемом в реальном времени, для повышения информативности о демонстрируемых образцах;
- формирование и интерактивная демонстрация виртуальной среды микро- или макромира (например, структуру сложной молекулы или поверхности Марса);
- интерактивность – ключевой момент, обеспечивающий имитацию непосредственного взаимодействия лектора с моделируемыми объектами и явлениями, при этом внутренняя структура объектов и связи между ними могут быть представлены наиболее наглядным образом;
- простой и лёгкий в обучении пользовательский интерфейс, основанный на технологии “Touchscreen” и позволяющий лектору или ведущему самому управлять ходом презентации, выбирать режимы микширования и медиа источники;
- возможность построения системы удалённой презентации в режиме телеконференций с высококачественной передачей изображений в реальном времени;
- создание банка данных лекционных и презентационных видеоматериалов с «присутствием» реального персонажа (лектора, экскурсовода и т. д.) в предметной виртуальной среде;
- вывод изображений как на стандартные средства отображения (проектор, монитор, плазменный экран), так и стереоскопические;
- функционирование в режиме реального времени;
- отображение на экране совместно с пользователем мультимедийных данных различных форматов (PowerPoint Presentation, AVI, аудио);
- управление процессом презентации с различных внешних устройств (наладонный компьютер, мобильный телефон, различные типы джойстиков, дистанционный пульт);
- возможность работы с мультимедийными материалами произвольного разрешения как на входе, так и на выходе системы презентации (от обычного телевизионного – SDTV до телевидения высокой четкости – HDTV или высокие компьютерные разрешения – XVGA).

Для телевизионного производства видеоматериалов и обучения процессу их создания в малобюджетных студиях разработана виртуальная студия «Фокус». Это высокоинтегрированный комплекс, включающий все основные технологии современного интерактивного телевизионного производства: съёмку сюжетов в реальном времени, воспроизведение медиаданных, генерацию 2D-3D изображений, ввод видеопотоков, микширование видео и звука, интерактивное и сценарное управление, за-

пись, вещание и протоколирование. Студия «Фокус» поддерживает:

- одновременно форматы от SD до 4K аналоговых и цифровых данных;
- съёмку в 3D декорациях в реальном времени;
- виртуальные видеокамеры вместо систем отслеживания реальных видеокамер;
- интерфейс оператора-режиссёра для каждой телепередачи;
- выбор виртуальных сцен из библиотеки;
- собственный дизайн трёхмерных декораций (пример на рис.3) и сценариев передач.



Рис. 3: Пример дизайна трёхмерных декораций

Для разработки виртуальных трёхмерных декораций и сценариев рабочего процесса студия «Фокус» позволяет использовать широкий набор современного инструментария. Многие разработанные алгоритмы для студии уникальны. Это средства рирпроекции, синхронной обработки сигналов, средства построения и управления данными в реальном времени.

3. Заключение

Профессиональная деятельность в очень многих сферах подразумевает владение средствами работы с медиаданными для создания высокоинформативных аудиовизуальных презентаций, иллюстраций, докладов, учебно-технических материалов, вещания и т.д. Это журналистика, музейная деятельность, производственно-технические, научные, торгово-экономических отрасли. Современная организация работы в ситуационных центрах (армейских, МЧС и т. д.) также требует высокоэффективной подачи наглядного материала, которая обеспечивается современными навыками работы с медиаданными.

В работе изложен многолетний накопленный опыт по созданию современных компьютерных средств производства видеопродукции и использованию их в образовательном процессе, нашедших широкое

распространение, как в отечественной, так и в зарубежной практике.

К основным областям применения можно отнести: обучение в реальном времени в системе очного и дистанционного образования, создание электронных учебных пособий с использованием интерактивных виртуальных сред, создание презентационных и обучающих видеороликов различной тематической направленности, обучение в режиме телеконференций, оперативная подготовка и переквалификация специалистов для ряда новых областей техники и промышленности, создание виртуальных музеев, планетариев и лекционных залов, создание телевизионных передач в реальном времени. В заключении, в докладе приводятся видеоматериалы по использованию созданных систем в различных областях обучения.

Литература

- [1] Долговесов Б.С. и др. Система обучения и презентаций с использованием технологии «погружения» лектора в интерактивную виртуальную среду // Сборник трудов «Online EDUCA 2007, Moscow». 2007.
- [2] Ванданов В. Г. и др. Программно-аппаратный комплекс 3d презентаций на основе виртуальной студии и виртуального окружения // Труды конференции «Трёхмерная визуализация научной, технической и социальной реальности. Кластерные технологии моделирования». 2009. Том 1. С. 73-77.
- [3] Долговесов Б.С. и др. Объектно-ориентированная база данных в интерактивных системах 3D визуализации // Вестник НГУ. Серия: Физика (том 6, выпуск 3) . 2011. С.59-63.
- [4] Городилов М.А. и др. Управление конфигурациями конвейера обработки изображений и спецэффектов систем визуализации реального времени // Сборник трудов «Технологии Microsoft в теории и практике программирования». 2011. – С.85-87.
- [5] Ковальков М.А. и др., Разработка и реализация алгоритмов рирпроекции на базе современного графического акселератора // Труды конференции Графикон-2006. 2006 . С. 360-362
- [6] Мазурок Б.С., Долговесов Б.С., Коростелёв Е.И., Артиков Т.Н., Артиков А.Н. Программно-аппаратный комплекс интерактивных мультимедийных презентаций // Труды конференции Графикон-2013. 2013. С.152-156

About the autor

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